Improving and Interpreting Machine Learning Algorithms with Applications

vorgelegt von
M.Sc.
Marina Marie-Claire Vidovic
geb. in Aachen

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzende	Prof. Dr. Anja Feldmann
Gutachter	Prof. Dr. Klaus-Robert Müller
Gutachter	Prof. Dr. Dario Farina
Gutachter	Prof. Dr. Benjamin Blankertz

Tag der wissenschaftlichen Aussprache: 11. August 2017

Berlin 2017

Abstract

Advancing understanding and interpretation of machine learning algorithms has recently been receiving much attention. Although classification systems achieve high prediction accuracies and are used across a wide spectrum of academic fields, they act like a black-box and provide little or no reasoning information about their decisions. However, several applications in science and technology require an explanation of the learned decision function.

Firstly, in the field of bioinformatics, positional oligomer importance matrices (POIMs) have been successfully applied to visualize the significance of position-specific subsequences. Although being a major step towards the explanation of trained support vector machine (SVM) models, they suffer from the fact that their size grows exponentially with the length of the motif, which renders their manual inspection feasible only for comparably small motif sizes. Therefore, in the first part of this thesis, we extend the work on POIMs, by presenting a new machine learning methodology, entitled motifPOIM, to extract the truly relevant motifs—regardless of their length and complexity—underlying the predictions of a trained SVM model. The proposed framework thereby considers the motifs as free parameters in a probabilistic model and is phrased as a non-convex optimization problem. In a next step, we derive a convex formulation of the previous presented motifPOIM approach, which provides a more robust and faster motif extraction.

Secondly, as a generalization of POIMs towards arbitrary classifiers and feature representations, the feature importance ranking measure (FIRM), has been proposed. Although FIRM provides a rich theoretical concept, it is computationally not feasible for most real-world use cases. Hence, we propose a new method named measure of feature importance (MFI), a simple and easy to use extension of FIRM that can assess — from arbitrary learning machines — the importance of even *non-linearly* coupled features, either *instance-based* (what features drove the classification decision for a given instance?) or *model-based* (which features are generally important for the classifier decision). Our formulation is based on the Hilbert-Schmidt independence criterion, which originally has been proposed to measure statistical independence for variables that exhibit non-linear couplings. We demonstrate the benefit of our proposed MFI method for SVMs and Convolutional Neural Networks on both artificially generated data and real-world applications.

Thirdly, this thesis tackles the problem of non-stationarities in the data, which is another well-known issue in the field of machine learning. Covariate shifts, i.e., fundamental changes over time in the data, are challenging for most real-world scenarios. In this thesis, we approach this problem excessively on the example of EMG signals for controlling prosthetic devices, where covariate shifts are generally caused by electrode shifts after donning and doffing, sweating, additional weight or

varying arm positions. A substantial decrease in classification accuracy due to these factors distorts the direct translation of EMG signals into accurate myoelectric control patterns outside laboratory conditions. To overcome this limitation, we propose the use of supervised adaptation methods. The approach is based on adapting a trained classifier using a small calibration set only, which incorporates the relevant aspects of the non-stationarities, but requires only less than 1 min of data recording.

Zusammenfassung

Fortschritte bezüglich Verständnis und Interpretation von Machine Learning Algorithmen haben in den vergangen Jahren stark an Aufmerksamkeit gewonnen. Obwohl Klassifikationssysteme hohe Vorhersagegenauigkeiten erzielen und über ein breites Spektrum akademischer Felder hinweg eingesetzt werden, agieren sie wie eine Black-Box und liefern wenige oder gar keine Informationen über getroffene Entscheidungen. Allerdings benötigen viele wissenschaftliche und technologische Anwendungen eine Erklärung für die von der Lernmaschine getroffene Entscheidung. Zunächst wurden auf dem Gebiet der Bioinformatik die Positional Oligomer Importance Matrizen (POIMs) erfolgreich angewendet, um die Relevanz einzelner positionsspezifischer Teilsequenzen zu visualisieren. Obwohl POIMs zum Verständnis von SVM Modellen beitragen, ist dieser Ansatz in seiner Anwendung stark beschränkt, da die POIM Größe exponentiell mit der Länge der gesuchten Motive wächst.

Im ersten Teil dieser Dissertation, erweitern wir das Konzept von POIMs. Dazu stellen wir eine neue maschinelle Lernmethode namens motifPOIM vor, die es ermöglicht die der SVM zugrunde liegenden relevanten Motive - unabhängig von ihrer Länge und Komplexität - zu extrahieren. Dabei werden die Motive als freie Parameter in einem probabilistischen Modell betrachtet, das als nicht konvexes Optimierungsproblem formuliert werden kann. Anschließend wird eine konvexe Formulierung des bisherigen "motifPOIM-Ansatzes" hergeleitet, die eine robustere und schnellere Motiv-Extraktion ermöglicht.

Als Verallgemeinerung von POIMs im Hinblick auf beliebige Klassifikatoren und Merkmalsdarstellungen wurde die Feature Importance Ranking Measure (FIRM) Methode vorgeschlagen. Obwohl FIRM ein elegantes theoretisches Konzept darstellt, ist es physikalisch für die meisten Anwendungen nicht berechenbar. Daher stellen wir im zweiten Teil der Dissertation eine neue Methode namens Measure of Feature Importance (MFI) vor. MFI ist eine einfach zu benutzende Erweiterung von FIRM, die die Wichtigkeit von nicht-linear gekoppelten Merkmale für beliebige Lernmaschinen beurteilen kann, sowohl Beispiel basiert (Welche Merkmale eines gegebenen Sample waren für die Klassifikator Entscheidung relevant?) als auch Model basiert (Welche Merkmale sind für den Klassifikator allgemein relevant?). Unsere Formulierung basiert auf dem Hilbert-Schmidt-Unabhängigkeitskriterium (HSIC), das ursprünglich als eine Möglichkeit zur Messung der statistischen Unabhängigkeit für Variablen mit nichtlinearen Kopplungen entwickelt wurde. Wir evaluieren unsere vorgeschlagene Methode für SVMs und Convolutional Neural Networks auf sowohl künstlich erzeugten Daten als auch auf realen Anwendungen.

Als drittes Thema dieser Dissertation wird die Analyse und der Umgang mit nichtstationären Daten behandelt. Die sogenannte Kovariate Verschiebung, gemeint ist die Veränderungen der Dateneigenschaften über die Zeit, stellt eine große Herausforderung

im Bereich des Maschinellen Lernens dar. Wir behandeln die Kovariate Verschiebung ausführlich am Beispiel von EMG Signalen zur Steuerung von Prothesen. Dabei werden Veränderungen in den Daten z.B. durch Verschiebung der Elektroden nach Anziehen und Abziehen der Prothese, Schwitzen, Zusatzgewicht oder variierenden Armpositionen verursacht. Diese Faktoren reduzieren die Klassifizierungsgenauigkeit und damit auch die Genauigkeit der EMG-Signal Übersetzung in myoelektrische Kontrollmuster außerhalb von Laborbedingungen. Um dieser Verschlechterung entgegenzuwirken, stellen wir eine Adaption des trainierten Klassifikators vor. Dabei basiert die Adaption auf einem kleinen Kalibrationsdatenset, das weniger als 1 Min Datenerfassung benötigt, und gleichzeitig alle relevanten Aspekte der Nichtstationaritäten beinhaltet.

Acknowledgements

Above all I would like to thank Prof. Dr. Müller who initially introduced me to the subject of machine learning in his lectures. His impressive knowledge, engagement and infectious enthusiasm for machine learning was at the same time my motivation to continue devoting to this subject. I am very grateful that he gave me the opportunity to work as a PhD student in the IDA group.
Special thanks go to Prof Dr. Dario Farina, who supported me during my time in Vienna at Otto Bock. I thank him so much for his intense and precise support during the writing process of our papers.
Furthermore, I want to thank Prof. Dr. Benjamin Blankertz for his immediately commitment for being one reviewer of this thesis.
I owe the utmost thanks to my advisor, Prof. Dr. Marius Kloft, who invariably supported and mentored me during the whole time. I thank him for all his care and valuable time. His huge amount of knowledge, his advices and his constructive comments were instrumental for my thesis. Among other things, he taught me about approaching difficult and complex problems and about the ability of viewing things from other directions. I am deeply thankful to him.
Furthermore, I am very grateful to my dearest college Nico Görnitz for our extensive and fruitful discussions on various mathematical and biological problems. I would like to thank him so much for his intensive participation, his creative and productive ideas and for sharing his immense knowledge and deep insights with me. I would like to thank him for his permanent support with helpful hints and advices for coming further with our projects.
Furthermore, I would like to thank Dr. Sören Sonnenburg for offering the quite challenging task to continue in the field of motif recognition in DNA sequences by introducing me to POIMs in the first place.
I owe my deepest thanks Dr. Johannes Höhne. He was always there, supporting me, assisting me to stay focused, helping me out of my downs and motivating me to finish my thesis as fast as I did.
Most of all, I would like to thank my parents, who gave me the opportunity to start and pursue a career in science. I am particularly indebted to my parents for their never-ending encouragement and ongoing support.

Contents

Chapter 1

Preface

Machine learning, as a branch of Artificial Intelligence, copes with the question "how can we incorporate intelligence into a machine?" Simply said, this is done by a set of algorithms, which allow to train a predictive model by learning from the information hidden inside a given data set. Afterwards, the trained model is able to assess unknown data points autonomously. Machine learning contributed remarkable achievements in almost all fields of science, such as bioinformatics, natural language processing, brain computer interface and recommender systems, to name a few. However, domain experts often are hesitant when it comes to new machine learning methods and do not trust the classifiers decision for a good reason, as the following example shows. In the work Ribeiro et al. (2016)[1], titled "Why should I trust you?", the authors train a classifier to decide whether there is a wolf or a husky in a given image. The training set consists of 20 images.

Fig. 1.1: *Demonstration of the major role of methods, that are able to interpret and explain the decision made by a classifier. In this example, the classifier was trained to decide whether it is a Wolf or a Husky on a given image. The explanation (b) of the wrong classifier decision of image (a) indicates the snow as most significant feature. Given this explanation, where the snow is significant for the class Wolf, we can assume that the classifier is trained badly and not trustworthy. This figure is taken from Ribeiro et al. (2016)*

The wolf pictures had snow in the background and the huskies did not. After training, the classifier predicts Wolf for each image if there is snow in the background, regardless of the animal color, position or pose and husky otherwise. Due to the given training

[1] Similar findings have been reported in following contributions: Baehrens et al. (2010), Bach et al. (2015), and Montavon et al. (2017).

set, snow was a significant feature for the class Wolf. However, the classifier would fail completely on images including Wolfs with different backgrounds. Unfortunately, due to their black-box characteristics, most machine learning models do not disclose the reason for the decisions they made to the user. Hence, methods that are able to explain the decision of a trained classifier and expose the discriminative features that were learned are of high value. Especially in recent years the interest of understanding the classifier more deeply has grown strongly with the variety and increased frequency of real-world applications using machine learning methods.

The other phenomenon, which the example referred to is called selection bias or selection effect, where the individuals or groups chosen are not representative for the population. Due to the fact that not all possible data configurations can be considered in advance, most real-world applications struggle with this problem. Machine learning algorithms have to deal with changing environments or changing conditions, which implant non-stationarities in the data. Due to time-consuming data recordings and even unpredictable factors, the algorithms must be equipped with the ability to adapt to changing data configurations.

Within this thesis, we tackle both:

1. **Interpretation**
   Understanding the decision made by a classifier is of high value in many applications. Especially in real-world applications, where machine learning is used for medical diagnosis or autonomous driving, determining trust by reasoning in individual predictions is of high importance. Experts need reliable indications to trust the decisions made by a black-box classifier. Therefore, methods that explain the decision made by a classifier are of great importance. This thesis addresses the weak points of previous approaches and improves them with regard to a simpler, more user-friendly and faster interpretability. The proposed methods are evaluated extensively on synthetic data as well as real-world data and compared with others approaches.

2. **Adaptation**
   Most real-world applications struggle with changing environments and unforeseen conditions. Resultant non-stationarities in the data represents a challenge to machine learning methods. To counteract a decrease in performance due to the selection bias, machine learning algorithms need to be equipped with the ability to adapt to new data conditions. In this thesis, we address the problem of changing conditions in the field of myocontrol algorithms for controlling prosthetic devices and propose a new adaptation method, which significantly increases the classifier robustness.

1.1 A Roadmap through this thesis

Chapter 2 In this chapter we introduce the basic preliminaries of this thesis. Fundamental concepts of machine learning, various classification algorithms and the well-known problem of covariate shifts are presented. The need of further reaching interpretation of the decisions made by classifiers is introduced as well as some existing explanation methods, namely POIMs, FIRM, LRP, HSIC.

Chapter 3 The focus of this chapter is mainly on bioinformatics, where we introduce the task of motif finding in DNA sequences given a trained support vector machine (SVM). Therefore, a mathematical framework is built to extract the relevant motifs from positional oligomer importance matrices (POIMs) by a non-convex optimization problem. Afterwards, we simplify the objective function, yielding a reformulation of the previously non-convex optimization problem to a convex one. We prove both, that the convexity condition is fulfilled and that the reformulation is legitimate.

Chapter 4 A new explanation method called measure of feature importance (MFI) is presented. The method allows to extract the feature importance for arbitrary learning machines and feature representations for both, model-based and instance-based explanation. We additionally present a kernelized version kernel MFI, which is able to find non-linear coupled feature importance in the data.

Chapter 5 The second main contribution of this thesis addresses the challenge of covariate shifts, which we treat on the example of EMG signals for myocontrol algorithms controlling prosthetic devices. We present an adaptation method for linear discriminant analysis (LDA) which can deal with changing environments.

Chapter 6 We conclude the thesis with summarizing the main findings and discussing open problems as well as directions for future work.

1.2 Own Contributions and Publications

For the readers awareness, we mention here that significant parts of this thesis have been previously published as journal articals and conference papers. The main contributions to this thesis are listed below.

1.2.1 Main Contributions

Journal Articles

1. MMC Vidovic, N Görnitz, KR Müller, G Rätsch, and M Kloft (2015b). "SVM2Motif — Reconstructing Overlapping DNA Sequence Motifs by Mimicking an SVM Predictor". In: *PloS one* 10.12, e0144782

2. MMC Vidovic, M Kloft, KR Müller, and N Görnitz (2017). "ML2Motif—Reliable extraction of discriminative sequence motifs from learning machines". In: *PloS one* 12.3, e0174392

3. MMC Vidovic, HJ Hwang, S Amsüss, JM Hahne, D Farina, and KR Müller (2016b). "Improving the robustness of myoelectric pattern recognition for upper limb prostheses by covariate shift adaptation". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24.9, pp. 961–970

Conference Articles

4. MMC Vidovic, N Görnitz, KR Müller, G Rätsch, and M Kloft (2015a). "Opening the Black Box: Revealing Interpretable Sequence Motifs in Kernel-Based Learning Algorithms". In: *ECML PKDD*. vol. 6913, pp. 175–190

5. MMC Vidovic, LP Paredes, HJ Hwang, S Amsüss, J Pahl, JM Hahne, B Graimann, D Farina, and KR Müller (2014). "Covariate shift adaptation in EMG pattern recognition for prosthetic device control". In: *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE, pp. 4370–4373

6. MMC Vidovic, N Görnitz, KR Müller, and M Kloft (2016a). "Feature Importance Measure for Non-linear Learning Algorithms". In: *arXiv preprint arXiv:1611.07567*
   Achieved the *best paper award* at NIPS 2016 in the workshop Interpretable ML for Complex Systems

1.3 List of Abbreviations

- CNN: Convolutional Neural Network
- diffPOIM: Differential POIM
- FIRM Feature Importance Ranking Measure
- HSIC: Hilbert-Schmidt Independence Criteria
- LDA: Linear Discriminant Analysis
- LRP: Layer-wise Relevance Propagation
- MFI: Measure of Feature Importances
- POIM: Positional Oligomer Importance Matrix
- SVM: Support Vector Machine
- USPS: United States Postal Service
- MRQ: Motif Reconstruction Quality

1.4 Basic Notation in Chapter 3

Due to the large technical depths in Chapter 3, we introduce the following notations.

Tab. 1.1: *Glossary of most important variables, functions, and symbols.*

Symbol	Description
k	Length of motif
Q_k	POIM of grade k
Ω	diffPOIM
R	motifPOIM
$x \in \mathcal{X}$	Calligraphic upper case characters are input spaces for which the corresponding lower case characters are realizations
$\mathbb{1}_{\{x=y\}}$	Indicator function (returns 1 if $x = y$ else 0)
$s(x)$	Classifier scoring function (returns scalar score given an input instance x)
$\bar{s}(x\|m_k)$	Reconstructed classifier scoring function given an input instance x and a motif m_k
L	Length of a sequence
$X[j]^k$	The subsequence within X starting at position j with length k
$\Phi(x)$	Feature representation of the WD-kernel
Σ	DNA alphabet $\{A, C, G, T\}$
PWM r	Positional weight matrix
μ	Start position of motif in the sequence
σ	Variance of the start position of the motif
PPM m_k	A probabilistic positional motif (aka *motif*) consists of a PWM together and its starting position with variance
$v(m_k)$	Weight function induced by the motif
(z, i)	A positional k-mer is k-mer $z \in \Sigma^k$ at position i
$\tilde{k}$	Length of SubPPM
$(m_k, \tilde{k})$	SubPPM
D	Number of overlapping SubPPMs
$\tilde{\mu}$	Start position of the SubPPM
$\mathcal{K}$	Set of all motif lengths
T	Vector that contains the number of PPMs for each motif lengths
λ	Weight of a PPM
η	Short cut for $(m_{k,t}, \lambda_{k,t}, \tilde{k})_{t=1,\ldots,T_k, k\in\mathcal{K}}$
ϵ	Machine precision
Π	Objective function
U	Compact set, where Π is defined on
k_{max}	Maximal POIM degree

Chapter 2

Fundamentals

In this section we introduce the basics in machine learning, starting with a theoretical section about risk minimization and kernel methods. Afterwards we present the learning models of Support Vector Machines (SVM), Linear Discriminant Analysis (LDA) and Convolutional Neural Networks (CNN), to which we will refer in the subsequent chapters. Then, we deal with the field of interpretation: why do we need interpretation of machine learning models and which explanation methods exist so far? Finally, we present the data sets, which will appear repeatedly in different sections of this thesis as well as some validation strategies we will use.

2.1 Learning Models

The core of supervised machine learning for classification is as follows: every learning task starts with a set of training data, which are some data with an additional class allocations, where the number of classes is at least two. Now, we want to extract the most relevant information and structures in the training data, so that we can predict the right class label and make class predictions even on unknown data. Several questions arise on the way from the training data to the trained learning model, such as, which is the best data representation or which learning model is the optimal choice? Unfortunately, there is no generic recipe, since the selection of an appropriate method and data representation as well as suitable preprocessing steps depends on many factors, such as the quality, the size or the nature of the data. Regarding the computation time and the required memory space, the choice of the method also largely depends on how much time and capacity is available.
Once we decided for a specific data representation and a machine learning method, one major issue is to give precise answers to the following questions: How good does the algorithm perform on unknown test data? What is the smallest possible error we can achieve and how many training samples are necessary therefore? The field facing these questions of risk minimization is called statistical learning theory, which we are going to present next.

2.1.1 Risk Minimization

Given some identically and independently distributed (iid) training data $X = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i$ is an element of a set $\mathbf{X}$ and $y_i \in \mathcal{Y} = \{+1, -1\}$ is its class label, the goal in statistical learning theory (SLT) is to find a classifier, predicting the correct label of any given observation $x \in \mathcal{X}$.

Therefore, in a training step, the learning model learns a function $f : \mathbf{X} \to \mathcal{Y}$ from X, where $f = sign(g(x))$ and $g(x) : \mathbf{X} \to \mathbb{R}$. Afterwards it should be able to classify unseen data $\boldsymbol{x} \in \mathbf{X}$ correctly, where f assigns the label $+1$ to x if $g(\boldsymbol{x}) \geq 0$ and -1 if $g(\boldsymbol{x}) < 0$. The function separates a data point (x_i, y_i) correctly if and only if the condition $y_i g(\boldsymbol{x}_i) \geq 0$ is met. Due to the fact that the whole class of possible functions $\mathcal{F} = \{f : \mathbf{X} \to \mathcal{Y}\}$, from which the learning machine chooses, contains many possible functions that separate the data, we need a strategy for choosing the best one. Therefore, we introduce the statistical learning theory based on Vapnik (1995): The best function f that one can obtain is the one minimizing the probability of misclassification, i.e., minimizing the expected error

$$R[f] = \int l(f(\boldsymbol{x}), y) dP(\boldsymbol{x}, y)\,, \tag{2.1}$$

where $l(f(\boldsymbol{x}), y)$ denotes a suitably chosen loss function, e.g., $l(f(\boldsymbol{x}), y) = \Theta(-y(f(\boldsymbol{x})))$, where $\Theta(z) = 0$ for $z < 0$ and $\Theta(z) = 1$ otherwise. The loss, mentioned above is also called $0/1$-loss. The probability distribution of the training data $P(\boldsymbol{x}, y)$ is unknown, and we thus cannot compute $R[f]$ directly. Instead we approximate the minimum of (2.1) by the minimum of the empirical risk (rate of misclassification on the training data set)

$$R_{emp}[f] = \frac{1}{n} \sum_{i=1}^{n} (f(x_i), y_i).$$

We can always obtain an empirical error of zero by increasing the function complexity, which is likely to lead to an overfitted system and a high generalization error (Steinwart and Christmann, 2008; Bishop, 2006; Shawe-Taylor and Cristianini, 2004).
By adding regularization terms to the objective function we can limit the complexity of the function class from which the learning machine selects. Such an approach has been proposed in the seminal work of Vapnik (1995). Vapnik bounds the true risk by the sum of the empirical risk and a complexity term:

$$R[f] \leq R_{emp}[f] + \sqrt{\frac{h(ln(\frac{2n}{h}) + 1) - ln(\frac{\delta}{4})}{n}} \quad \forall\ \delta \geq 0$$

with a probability of at least $1 - \delta$ for $n > h$, where h is the Vapnik-Chervonenkis (VC) dimension. In short, the VC dimension is a measurement for the separability of points from different classes. For more information see Bousquet et al. (2004) and Vapnik (1995). Next, we concentrate on the question: how should we represent the training data? Does there exists an optimal data representation for the classification algorithm?

2.1.2 Data Representation

Before introducing various concepts of learning machines, we demonstrate the necessity of an appropriate data mapping from the input space to a feature space for a more convenient classification. Therefore, we introduce the following example.

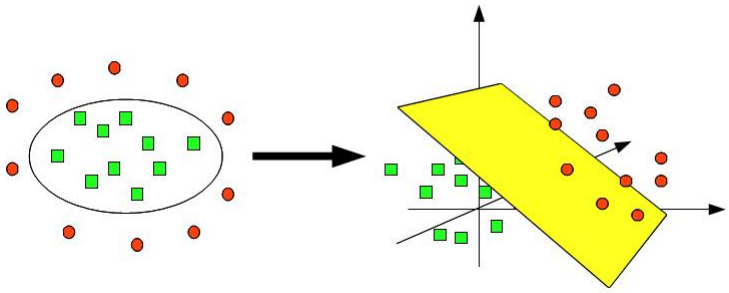


Fig. 2.1: Illustration of a feature space mapping. *The original data become separable by a hyperplane after mapping the two dimensional data into a three dimensional space by* $\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$.

Example 1 *The data, shown on the left-hand side in Figure 2.1, are not separable by a linear model in its original representation. By mapping the two dimensional data into a three dimensional space using the injective mapping*

$$\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

we obtain a data representation, which is separable by a linear classifier as shown on the right-hand side in Figure 2.1.

Thus, by mapping data into a higher dimensional space using a function $\Phi : \mathbb{R}^N \to \mathcal{F}$ we obtain a data representation $S = (\Phi(x_1), \ldots, \Phi(x_n))$ in the high-dimensional space $\mathcal{F}$ (called feature space), which is then separable by a hyperplane. In the feature space the learning machine has to find a linear separation model instead of a complex nonlinear one in the original space.

2.1.3 Support Vector Machines

Here we consider the binary classification problem, where we are given a training set $X = \{(x_i, y_i)_{i=1}^n\}$, with $x_i \in \mathbb{R}^d$, $d \in \mathbb{N}$ and $y_i \in \{-1, 1\}$. Suppose that the data is linearly separable in the feature space by a hyperplane

$$H = \langle \Phi(\boldsymbol{x}), w \rangle + b, \tag{2.2}$$

where w is the normal vector, b the offset to ground zero, and Φ a suitable mapping function. Now, scaling the hyperplane relative to the data, we achieve the so-called canonical representation of the hyperplane:

$$y_i(\langle w, \Phi(x_i) \rangle + b) \geq 1 \quad \forall \;\; i = 1, \ldots, n\,. \tag{2.3}$$

The domain of confidence, where no data point lies in between, is called the margin, which is given by the smallest distance between any two points of opposite classes and is exactly $\frac{2}{\|w\|}$ (see Appendix A.2 for derivation). It is easy to see, that the larger the margin the more reliable the classification of unknown data points. Hence, the goal is to maximize the margin. However, in practice, there often exist outliers or random

noise in the data such that a strict linear separation in feature space is not possible. Therefore, slack variables ξ_i with $\xi_i \geq 0,\ i = 1, \ldots, n$ are introduced to relax the hard margin constraints of Equation (2.3) as follows:

$$y_i(\langle w, \Phi(x_i)\rangle + b) \quad \geq \quad 1 - \xi_i \quad \forall \quad i = 1, \ldots, n \tag{2.4}$$

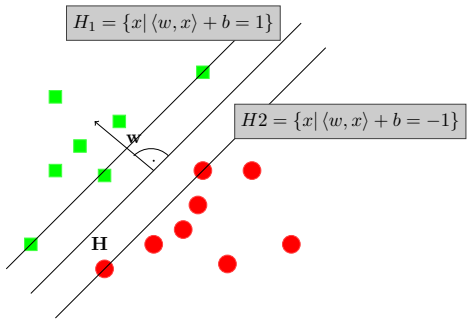


Fig. 2.2: Canonical representation of a hyperplane H. *The hyperplane separates the data in its two classes, visualized as green squares and red points, respectively.*

Now, we maximize the margin by minimizing w, which results in the following "soft margin" optimization problem with constraints, known as the primal problem

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \quad & y_i\left(\langle w, \Phi(x_i)\rangle + b\right) \geq 1 - \xi_i \qquad \forall \ \ i = 1, \ldots, n \\ & \xi_i \geq 0 \qquad\qquad\qquad \forall \ \ i = 1, \ldots, n\,. \end{aligned} \tag{2.5}$$

C is a regularization parameter, the so-called penalty factor, which has to be chosen adequately by hand (Müller et al., 2001; Schölkopf et al., 1998; Bishop, 2006). Large C-values yield high penalties for non-separable points and may lead to overfitting, whereas for C-values that were chosen too small the algorithm cannot capture the underlying trend of the data. The dual counterpart to 2.5 is as follows:

$$\begin{aligned} \max_{\alpha} \quad & L_d(\alpha) = \sum_{i=1}^{n}\alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \langle\Phi(x_i), \Phi(x_j)\rangle \\ s.t. \quad & \sum_{i=1}^{n}\alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C,\ i = 1, \ldots, n, \end{aligned} \tag{2.6}$$

where w is composed of the support vectors and their labels

$$w = \sum_{i=1}^{n}\alpha_i y_i \Phi(x_i). \tag{2.7}$$

For the explicit derivation of the dual optimization problem see Appendix A.3.

The reason for computing the problem in the feature space is the existence there of simpler classification rules (e.g., linear classifiers). Hence, the complexity of the function class is more important than the dimensionality of the problem (Vapnik, 1995). However, regarding high dimensional feature space, Müller et al. (2001) comment: *So, even if one could control the statistical complexity of this function class, one would still run into intractability problems while executing an algorithm in this space. Fortunately, for certain feature spaces $\mathcal{F}$ and corresponding mappings Φ there is a highly effective trick for computing scalar products in feature spaces using kernel functions.*

Kernel functions substitute the large computation in the feature space by a simple function in the original space (Bishop, 2006; Shawe-Taylor and Cristianini, 2004; Müller et al., 2001; Vert et al., 2004). We introduce kernel functions by computing the dot product of Example 1.

$$\begin{aligned}\langle \Phi(x), \Phi(y) \rangle &= \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)\left(y_1^2, \sqrt{2}y_1y_2, y_2^2\right) \\ &= x_1^2y_1^2 + 2x_1y_1x_2y_2 + x_2^2y_2^2 \\ &= (x_1y_1 + x_2y_2)^2 \\ &= \langle x, y \rangle^2 =: \kappa(x, y).\end{aligned}$$

The general definition of a kernel is given as follows (Vert et al., 2004):

Definition 1 (Kernel) *A function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite kernel iff it is symmetric, that is, $\kappa(x, x') = \kappa(x', x)$ for any two objects $x, x' \in \mathcal{X}$, and positive definite, that is,*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \kappa(x_i, x_j) \geq 0$$

for any $n > 0$, any choice of n objects $x_1, \ldots, x_n \in \mathcal{X}$, and any choice of real numbers $c_1, \ldots, c_n \in \mathbb{R}$.

Now, with Definition 1, the kernel trick is stated as follows:

Theorem 2 *For any kernel κ on a space $\mathcal{X}$, there exists a Hilbert space $\mathcal{F}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{F}$ such that*

$$\kappa(x, x') = \langle \Phi(x), \Phi(x') \rangle, \quad \textit{for any } x, x' \in \mathcal{X},$$

where $\langle u, v \rangle$ represents the dot product in the Hilbert space between any two points $u, v \in \mathcal{F}$.

Some popular kernels are (Müller et al., 2001; Schölkopf et al., 1998; Bishop, 2006):

$$\begin{aligned}\textit{Linear Kernel}: \quad \kappa(x, y) &= \langle x, y \rangle \\ \textit{Polynomial Kernel}: \quad \kappa(x, y) &= (\langle x, y \rangle + c)^d \, , c \in \mathbb{R} \\ \textit{Radial Basis Function(RBF) Kernel}: \quad \kappa(x, y) &= e^{-\frac{\|x-y\|^2}{2\sigma^2}}.\end{aligned}$$

Given Theorem 2, we can replace the dot product $\langle \Phi(x_i), \Phi(x_j) \rangle$ in the dual optimization problem (2.6) by the kernel function $\kappa(x_i, x_j)$. After solving this optimization problem we obtain the optimal α.
Remember that w can be expressed by the support vectors (see Equation (2.7)), which is why we can replace the dot product by a kernel function in the hyperplane H Equation (2.2). This leads to the following formulation of the kernelized SVM decision function

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \kappa(x, x_i) + b\right). \tag{2.8}$$

Note that the bias b can be computed for a non-zero α_i (the formula is given in Equation (A.4) in the appendix).
We have presented the SVM in the traditional way (Furey et al., 2000). Generating a hyperplane in the feature space and using kernel functions instead of computing the high dimensional dot products. The resulting SVM decision function is able to classify unknown data to one of the two classes +1 or -1 regarding Equation (2.8).

2.1.4 Linear Discriminant Analysis

For the following considerations we are given a training set $X = \{(x_i, y_i)_{i=1}^n\}$, with $n \in \mathbb{N}$ labeled d-dimensional samples, thus $x_i \in \mathbb{R}^d$, $d \in \mathbb{N}$, where $y_i \in \{1, \ldots, C\}$ denotes the class allocation, respectively. Furthermore, let $\pi_c = 1/C$ be the prior probability for each class $c \in \{1, \ldots, C\}$ and $f_c(x)$ the class-conditional density function of X, which we assume to be the multivariate Gaussian distribution

$$f_c(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_c|}} \exp\big(-\frac{1}{2}\hat{x}^\top \Sigma_c^{-1} \hat{x}\big), \tag{2.9}$$

where $\hat{x} := x - \mu_c$ and μ_c is the class mean. In this thesis, we focused on two Bayesian multi-class classifiers: linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) (Hastie et al., 2009). QDA determines quadratic boundaries for class separation by using class-wise covariance matrices, whereas LDA eliminates the quadratic terms by assuming equal covariance matrices for all classes and thus uses hyperplanes for classification. We consider the problem of a Bayesian multi-class classification: given an unknown point x, we want to allocate it to the class with the largest posterior probability $Pr(c|x)$ $c = 1, \ldots, C$ (Hastie et al., 2009). Circumventing the fact that the posterior probabilities cannot be obtained directly, we estimate them by using the Bayes rule on the training data:

$$Pr(c|x) := \frac{f_c(x)\pi_c}{\sum_{l=1}^{C} f_l(x)\pi_l}. \tag{2.10}$$

We obtain the QDA discriminant $\delta_c^1(x)$ for each class c by taking the natural logarithm of Equation (2.10) and rewrite it to

$$\delta_c^1(x) := -\frac{1}{2}\log|\Sigma_c| - \frac{1}{2}\hat{x}^\top \Sigma_c^{-1}\hat{x} + log\pi_c. \tag{2.11}$$

If we further assume that all classes share the same covariance matrix $\Sigma = \frac{1}{C}\sum_{c=1}^{C}\Sigma_c$, we can rearrange Equation (2.11) to the linear discriminant δ_c^2

$$\delta_c^2(x) := x^\top \Sigma^{-1}\mu_c - \frac{1}{2}\mu_c^\top \Sigma^{-1}\mu_c + log\pi_c. \tag{2.12}$$

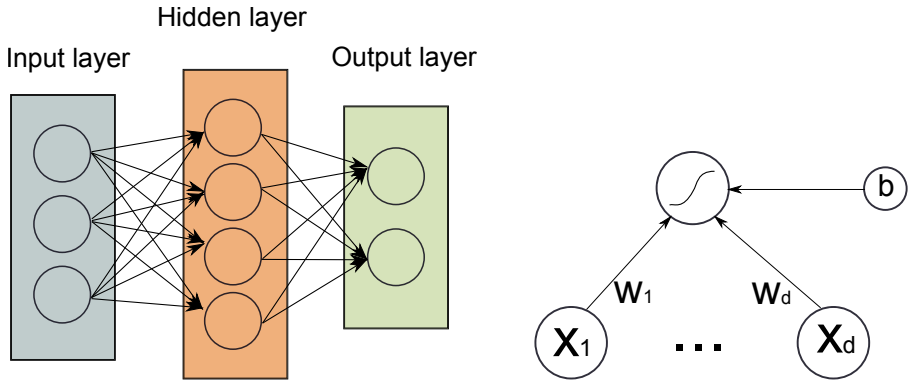
An unknown point $x^* \in \mathbb{R}^d$ is allocated to the class with the highest probability

$$y^* = \arg\max_c \delta_c^j(x^*), \tag{2.13}$$

by QDA ($j = 1$) and LDA ($j = 2$), respectively, where $y^* \in \{1, \ldots, C\}$. In a nutshell, building an LDA means, estimating the class means as well as the pooled class covariance matrix. Both methods are proven to be Bayes-optimal for strictly Gaussian distributed data.

2.1.5 Convolutional Neural Networks

Recently, deep neural networks, such as the convolutional neural network (CNN or ConvNets) have received high attention in almost all machine learning fields. They have been proven to be highly successful to image classification tasks (Krizhevsky et al., 2012). One of the characteristics of (convolutional) neural networks is the end-to-end training that eliminates the necessity of feature engineering, which is inevitable in traditional machine learning methods, such as SVM (see section 2.1.3). Furthermore, CNNs have been leveraged to classify and even generate sequential data (Kalchbrenner et al., 2016; Oord et al., 2016; Kim, 2014). Due to the vast amount of various deep network architectures and for the sake of simplicity, here, we focus on feed-forward networks only, where an example is shown Figure 2.3 (a), consisting of an input layer, a hidden layer and an output layer.



(a) Neural Network (b) Artificial Neuron

Fig. 2.3: Illustration of a feed-forward neural network (a) and an artificial neuron (b).

Just like the elemental neural networks, deep neural networks consist of layers, where each layer consists itself out of nodes, which are called neurons. A single neuron,

exemplary shown in Figure 2.3 (b), is a computational unit, which makes a particular computation based on other units it is connected to. The computation can be described in two steps: In the first step, known as pre-activation (or input activation), the product between the weight vector w and the input vector x is computed and a bias b is added,

$$g(x) = \sum_{i=1}^{d} w_i x_i + b.$$

In the second step, the output activation of the neuron is computed by passing the pre-activation through an (nonlinear) activation function σ,

$$h(x) = \sigma(g(x)). \tag{2.14}$$

The words neural and neurons point out the connection to the brain structure, which was the initial inspiration for the building of neural networks (Arbib, 2003). There is an extensive literature about how to configure and fine-tune neural network architectures (Montavon and Müller, 2012).
In the following we present the basic building blocks that are in convolutional neural network.
An illustration of a standard CNN architecture is given in Figure 2.4. Note, in this thesis, we present ConvNets on a high level since we use them in the application sections only. For further mathematical details see Simard et al. (2003), Duchi et al. (2011), and LeCun et al. (2012, 2015).

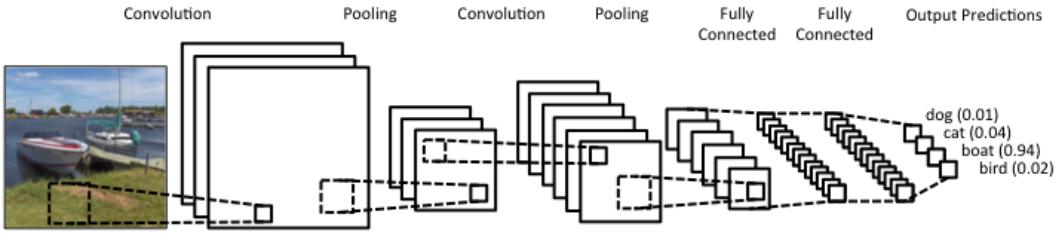


Fig. 2.4: Illustration of a convolutional neural network. *This figure was taken from* `https://www.clarifai.com/technology`

Convolution layer The purpose of the convolution step is to extract features from the input image by preserving the spatial relationship between pixels. This is done by a filter, a small matrix of fixed size, which slides over the image so that for each position the element-wise multiplication between the filter and the underlying image patch is computed. The multiplication results are added and inscribed in the output matrix respectively. Illustratively, this can be seen in the first two plots shown in Figure 2.4. A filter (e.g. size 5x5) is laid over the initial image, as shown in the first plot. Each added-up multiplication is inscribed to the output matrix of the next layer, which can be seen in the second plot of Figure 2.4. Intermediate outputs in the hidden layers are also called feature maps. In the pre-activation of a convolution layer, the

input feature map h^{l-1} of layer $l-1$ is convoluted with the linear filter ω of size $m \times n$ and a bias b is added. Corresponding to Equation (2.14), the feature map h^l of layer l is obtained by applying a non-linear activation function σ (see next paragraph):

$$h_{ij}^l = \sigma\Bigg(\sum_{a=0}^{m-1} \sum_{b=0}^{n-1} \omega_{ab} h_{(i+a)(j+b)}^{l-1} + b_{ij} \Bigg).$$

Convolutional filters can be seen as feature detectors from the initial input image, where different values of the filter matrix produce different feature maps for the same input image. A filter can detect edges, curves or higher-level structures, such as faces. When initializing a CNN, the number of filters, which define the depth of a convolution layer, as well as their filter sizes have to be specified in advance, whereas the weights of the filters itself are learned during the training process. Also the stride, which is the number of pixels the filter is moved forward over the input image has to be determined manually.

Non-linear activation function Most real-world applications involve non-linear data, which means that the convolution layer, which is intrinsically linear may not be sufficient. Therefore, non-linear activation functions, such as the rectified linear unit (ReLU) are incorporated in the network. A ReLU replaces all negative values in the feature map by zero

$$\sigma(x) = \max(0, x),$$

where x is a pixel in case of image classification. Besides ReLU, there exist other non-linear activation functions, such as tanh or sigmoid.

Pooling layer The purpose of the pooling step is to reduce the dimensionality of each feature map by retaining the most important information. Therefore, in a first step, a window of fix size is determined. Then the window slides, similar to the filters of the convolution step, over the feature map and performs a dimensionality reduction. This can be done by max pooling, where the largest element of the feature map, which is currently located within the sliding window, is taken as new value for the output matrix. Taken the average value or the sum of the feature map in the window would be named as average pooling or sum pooling. This step is visualized in Figure 2.4, where a window in plot 2 is reduced to a single pixel in plot 3. Note that the depth of the layer is maintained. Pooling layers leverage against overfitting, since they reduce the feature dimension, which in turn reduce the number of parameters and finally also make the network computationally faster. Furthermore, small distortions or transformations in the input image will not change the output of the pooling layer, which makes the network more stable regarding non-stationarities in the data (LeCun et al., 2015).

Batch normalization layer ConvNets increase their throughput by portioning the data into batches and processing one batch at one time instead of only one sample. A batch normalization layer normalizes the activations of the previous layer at each batch, i.e., it applies a transformation that shifts the inputs to zero-mean and

unit variance. This largely mitigates the internal covariate shift, which occurs when the data become extremely large or very small, when being adjusted by the weights and parameters throughout the network. For more information about batch normalization we refer to Ioffe and Szegedy (2015).

The layers presented so far form together the basis of a ConvNet and can be repeated several times (for instance in the example shown in Figure 2.4, both layers (convolution layer and pooling layer) were applied two times in a row). The overall purpose of those layers is the feature extraction. Additionally, they are able to incorporate non-linearity in the network and reduce feature dimensions while aiming to make the features invariant for scaling and translation.

Fully connected layer In a fully connected layer, which is a traditional Multi Layer Perceptron (MLP), every neuron is connected to all activations in the previous layer

$$h_j^l = \sigma(\sum_i w_{ij}^l h_{ij}^{l-1} + b_j).$$

A fully connected layer can perform the classification step in the neural network. The classification of the features h^L, generated through the last layer L is done by using the softmax activation function

$$p(y = k|h^L) = \frac{\exp(h_k^L)}{\sum_{c=1}^{C} \exp(h_c^L)}, \tag{2.15}$$

which returns probabilities for all C classes, where the output probabilities sum up to 1.
In the example shown in Figure 2.4, there are four classes, namely dog, cat, boat and bird. The highest target probability of 94% was allocated by the classifier to the class boat, which is, given the picture of the boat, the correct decision.

How to train a ConvNet? We briefly address the training of the presented ConvNet in its four essential steps. For an extensive overview and for the mathematical details we refer to LeCun et al. (1998), Simard et al. (2003), Duchi et al. (2011), and LeCun et al. (2012).

1. All filters and weights are initialized randomly.

2. Given an image of the training set, the network performs the forward propagation step along all layers and finally allocates a probability to each class. Since all parameters are randomly initialized the output probabilities of the first training iteration are random too.

3. The total error made by the network is computed by a loss function between the output and the target probabilities.

4. Now, backpropagation (LeCun et al., 1989) is used to propagate the error backward through the network, using its gradient with respect to the parameters,

respectively. Afterwards gradient descent is used for updating the parameter values, which minimize the output error of the network.

The network is trained consecutively by repeating step 2-4 for the whole training set. We have presented the architecture of a feed-forward network including four different types of layers, namely convolution layers, ReLU layers, pooling layers, and fully connected layers and its training procedure.

2.1.6 Covariate Shift

In statistics, the word covariate denotes a variable, which is observed during an experiment and which follows a probability distribution. Thus, intuitively, the covariate shift means a change in the distributions properties in the first place. Almost every real-world machine learning application has to cope with the phenomenon of covariate shift. Thereby, the fundamental assumptions made by supervised machine learning methods, which is that the training and test data follow the same probability distribution, is violated, e.g. due to the inevitable sample selection bias or a changing environment. However, an assumption about the relationship between training data and test data is essential to be able to learn from the training data (Sugiyama and Müller, 2005; Sugiyama et al., 2013).
Several works, including Shimodaira (2000) and Sugiyama and Kawanabe (2012) treat the covariate shift as assumption, where the training data $p_{tr}(x)$ and test data $p_{te}(x)$ have different distributions

$$p_{tr}(x) \neq p_{te}(x) \tag{2.16}$$

but their conditional distributions $p_{tr}(y|x) = p_{te}(y|x)$ concerning the output values given the input points remain the same. Shimodaira (2000) showed that covariate shift causes problems for misspecified models, which are models that cannot express the learning target function, that is, when there exists no model $\{P(Y, X, \Phi^*)\}$ from the parametric model family $\{P(Y, X, \Phi)\}_{\Phi \in \Theta}$ that can match the true relation between X and Y. Hence, the optimal model we select for the source domain will differ from the optimal model for the target domain due to Inequality (2.16). The influence of covariate shift could be alleviated by adjusting the training data density to the test data density by the density ratio $\frac{p_{tr}(x)}{p_{te}(x)}$ (Shimodaira, 2000; Sugiyama and Müller, 2005). Here, the key idea is that the optimal model performs better in dense regions of X.
Covariate shifts often appear when experiments have been translocated out of the laboratory into a real-word scenario. This is because not all eventualities can be foreseen or even incorporated in the laboratory conditions due to e.g. the extensive amount of data recordings. Therefore, it is essential to equip the classifier with the ability to adapt itself regarding the new data conditions. Thus, when building convenient and trustworthy classifiers, there is a strong need for adaptation methods, which exhibit a robust behavior even under changing environments.

2.2 Interpretation

Most machine learning algorithms act like black-boxes. The structures they have learned remain unknown and the reasons for the decisions they made are not disclosed to the user. However, for many applications an explanation of the respective decision is of great importance. On the one hand, human experts must know whether they can trust the decision made by the classifier. On the other hand, for various scenarios it is important to extract the underlying data structures learned by the machine. Hence, understanding the learned structures within a classifier and interpreting the decisions made by a classifier is of high value in many applications (Baehrens et al., 2010; Bach et al., 2015; Vidovic et al., 2015b, 2017). It may provide additional information to human experts and inspires more confidence in those otherwise black-box systems. Furthermore, regarding the huge dimensionality of many real-world problems, the interpretation of a system, which often is a ranking of the features, can even be used for feature selection. In this section, we discuss the interpretation techniques on which the later on proposed methods build upon. It is common to divide methodologies assessing feature importance into the following two distinct categories (Ribeiro et al., 2016)

- **Model-based** feature importance. Here, the task is to globally assess what features a given (trained) learning machine regards as most significant — independent of the examples given.
- **Instance-based** feature importance. Given a specific instance, i.e. sample, the task at hand is to assess why this instance has been assigned this specific classifier score (or class) prediction.

We start with Positional Oligomer Importance Matrices (POIMs) (Sonnenburg et al., 2008), a model-based approach, which comes from the field of bioinformatics and was specifically designed for DNA sequences. Zien et al. (2009) presented a generalization of POIMs — the feature importance measure (FIRM), which is applicable to arbitrary feature sets. Afterwards, the layer-wise relevance propagation (LRP) is presented, a method that is other than POIMs and FIRM an instance-based feature importance measure (Bach et al., 2015). Finally, we present a criterium called Hilbert Schmidt Independence Criteria (HSIC), which measures the independence of two random variables and can also be used for interpretation.

2.2.1 POIM - Positional Oligomer Importance Matrix

Suppose we have trained an SVM on a real human DNA data set for investigating a genetic illness (Mieth et al., 2016). Given a new patient DNA data to the SVM, the resulting diagnose is only a yes or a no to the potential patient disease. However, there is an enormous need of information about the genetically modified DNA motifs and their positions in the DNA sequence. To put it in a nutshell, there is a need to get to know the positional oligomers that contribute the most to the SVM decision, i.e., the discriminate motifs and their positions. This requirement is considered by positional

oligomer importance matrices (POIMs, (Sonnenburg et al., 2008)), a methodology to visualize feature importances of kernel machines over a quadbit alphabet (i.e., $\{A, C, G, T\}$, as in the case of DNA sequences), taking into account inter-correlations between the features. The approach is based on the weighted degree kernel, exemplary shown in Figure 2.5 (Rätsch et al., 2007; Sonnenburg et al., 2007; Ben-Hur et al., 2008), which compares two discrete sequences $x = (x_1, \ldots, x_L)$, $x' = (x'_1, \ldots, x'_L) \in \mathcal{A}^L$ of length L over the alphabet $\mathcal{A}$ with $|\mathcal{A}| < \infty$, by counting the number of matches of their subsequences up to a given length ℓ_{max}

$$\kappa(x, x') = \sum_{\ell=1}^{\ell_{\max}} \sum_{j=1}^{L-\ell+1} \mathbb{1}_{\{x[j]^\ell = x'[j]^\ell\}} . \tag{2.17}$$

Here, $x[j]^\ell$ specifies the length-ℓ subsequence of x starting at position j. Thus, each entry in the explicit representation of a sequence $\Phi(x)$ in the decision function of the kernel SVM

$$s(x) := \langle w, \Phi(x) \rangle$$

corresponds to a valid positional subsequence y of length $\ell \in \{1, \ldots, \ell_{\max}\}$ starting at position $j \in \{1, \ldots, L - \ell + 1\}$.

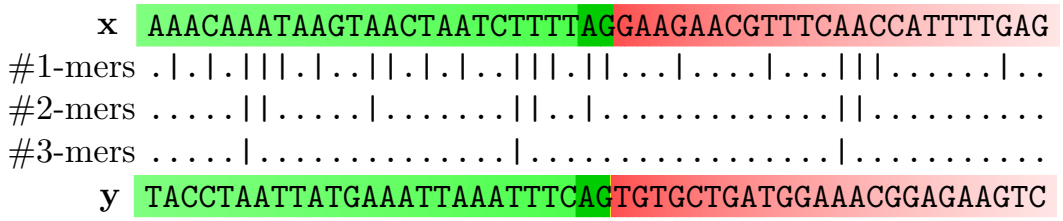


Fig. 2.5: Computation example of a weighted degree kernel for $\ell_{\max} = 3$***.*** *A match is a k-mer, which is observed in both sequences at the same position is marked by a dash. Finally all dashes are add up regarding Equation (2.17) (Rätsch et al., 2007). The figure was taken from (Sonnenburg, 2008).*

An entry in this feature representation $\Phi(x)$ of the kernel SVM (introduced in Section 2.1.3) equals one if the positional oligomer y occurs in x at position j and zero otherwise. Any zero entries of $\Phi(x)$ do not contribute to the dot product, which is why we can write $s(x)$ as a sum over the positional oligomer weights of w, which occur in x. Hence, we can rewrite the WD-kernel based scoring function as

$$s(x) = \sum_{\ell=1}^{\ell_{\max}} \sum_{j=1}^{L-\ell+1} w_{(x[j]^\ell, j)}.$$

In the case of support vector machines (SVM) (Cortes and Vapnik, 1995; Müller et al., 2001), the larger the absolute value $|s(x)|$ of a sequence x, the more reliable is the decision $\text{sign}(s(x))$. For instance in the application of gene finding, $s(x)$ would give large positive scores for sequences likely to contain genes and large negative scores for

sequences containing intergenic regions only. Following up on this intuition, POIMs are formally defined as follows.
From now on, let X be a uniformly distributed random variable over the DNA alphabet $\Sigma = \{A, C, G, T\}$ of length L.

Definition 3 (POIM) *Given an SVM scoring function s based upon a WD-kernel of, at least, degree $k \geq 1$, then for each possible k-mer y at position j we define the positional oligomer importance score as*

$$Q_{k,y,j} = \mathbb{E}[s(X)|X[j]^k = y] - \mathbb{E}[s(X)]\,,$$

which results, applied successively, in the positional oligomer importance matrix Q_k of order k.

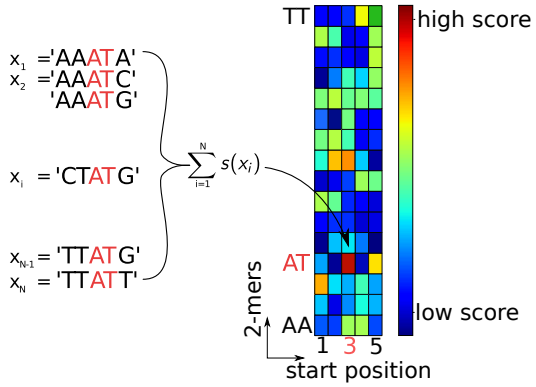


Fig. 2.6: Illustration of a POIM *of order 2 and length $l = 5$ over oligomers of length 2 ("2-mers"). Each POIM entry captures the significance of the particular 2-mer at the specific position in the sequence, which is, roughly said, the expected value of this positional 2-mer regarding the weights in the SVM WD-kernel. Boxes colored in dark red indicate the most discriminative positional 2-mers.*

There are two reasons for subtracting the expected value of the SVM scoring function $\mathbb{E}[s(X)]$ in the POIM Definition 2.18. Firstly, the expected value of the SVM scoring function can be considered as baseline value, which is necessary for the interpretation of the conditioned expected value of the scoring function with respect to a single positional oligomer. Secondly the computation speed is increased, since all non-overlapping positional oligomers do not have to be considered in the SVM POIM formula because their probability terms equal zero (cf. (Vidovic et al., 2015a; Vidovic et al., 2015b)). We can very well visualize POIMs in terms of heatmaps as illustrated in Fig. 2.6, from which we may obtain the most discriminative features by manual inspection.

2.2.2 Differential POIM

As a first step towards a standalone analysis of POIMs, Zien et al. (2007) proposed an extension of the POIM method, the so-called *differential POIM*, which aims to

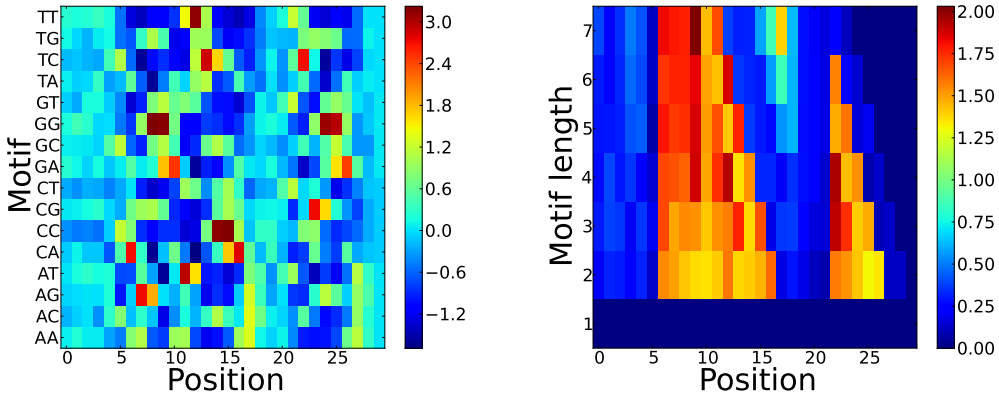
identify the most relevant motif lengths as well as the according starting positions. Formally, the differential POIM Ω is defined as a $k \times L$ matrix $\Omega := \left(\Omega_{l,j}\right)$ with entries

$$\Omega_{l,j} := \begin{cases} q_{\max}^{l,j} - \max\{q_{\max}^{l-1,j}, q_{\max}^{l-1,j+1}\} & \text{if } l \in \{2, \ldots, L\} \\ 0 \quad \text{elsewise}\,, \end{cases} \tag{2.18}$$

where

$$q_{\max}^{l,j} := \max_{y \in \Sigma^l} |Q_{l,y,j}|\,.$$

We can interpret $\Omega_{l,j}$ as an overall score for the general importance of the oligomers of length l starting at position j. An example of a diffPOIM with the corresponding POIM of order 2 is given in Figure 2.7.



(a) POIM of order 2

(b) Differential POIM

Fig. 2.7: Illustration of a POIM with its related differential POIM. *A POIM of order 2 is shown on the left, with its corresponding differential POIM, shown on the right. From the diffPOIM we can observe the most significant positions in the sequence, which are indicated by the boxes colored in dark red. Hence we can suppose that two motifs, starting at position 6 and 22, respectively are significant for the SVM decision function.*

2.2.3 FIRM - Feature Importance Ranking Measure

Due to the fact that POIMs are limited in applicability, firstly, to sequential data with binary features and secondly, to an SVM with a WD kernel, Zien et al. (2009) introduced the feature importance ranking measure (FIRM), as a generalization of POIMs to continuous features and arbitrary learning machines. FIRM measures the impact that a given feature

$$f : \mathcal{X} \to \mathbb{R}$$

from the input data has on the scoring function $s : \mathcal{X} \to \mathbb{R}$. Zien et al. (2009) give examples for features in case of a vectorial input $\mathcal{X} = \mathbb{R}^d$, such as simple

coordinate projections $f_j(x) = x_j$, coordinate pairs $f_{j,k} = x_j x_k$ or step functions $f_{j,\tau} = \mathbb{1}\{x_j \geq \tau\}$. FIRM consists of two steps. First, the conditional expected score of the scoring function that a feature f taken a certain value t is given as a function q over the values t.

Definition 4 (Conditional expected score) *The conditional expected score of a scoring function s for a feature f is the expected score $q_f : \mathbb{R} \to \mathbb{R}$ conditioned on the feature value t of the feature f:*

$$q_f(t) = \mathbb{E}[s(X)|f(X) = t] \quad . \tag{2.19}$$

The idea behind the computation of the conditional expected score as a function of the feature value t, which at the same time is the motivation of the next step, is that features with no or just small effect on the scoring function have a flat function q_f, whereas features with high importance will have a high variability in q_f. Therefore, the second step in FIRM establishes the variability of the conditional expected score as a measure for the importance of the corresponding feature.

Definition 5 (Feature importance ranking measure (FIRM)) *The feature importance $Q_f \in R$ of the feature f is the standard deviation of the function q_f:*

$$Q_f := \sqrt{Var[q_f(f(X)]} \quad .$$

Summarizing, FIRM measures feature importances for a given model. FIRM advances POIMs regarding to arbitrary learning machines and continuous features, which makes it a general theoretical approach. FIRM has a variety of interesting properties. Zien et al. (2009) tagged FIRM "objective", meaning that it is invariant with respect to translation and rescaling of features, which makes it a robust approach. Furthermore FIRM is "intelligent", since it is not limited to features that have been used in the learning machine. And above all, it is "universal", since it is applicable to any feature and any learning machine possessing a real valued output. Unfortunately, FRIM is computationally infeasible for most applications, since it explicitly computes the conditional expected value for the input space. Interestingly, Zien et al. (2009) discuss shortly the possibility of assessing all quantities empirically but let go of this idea "due to the limited amount of data". The authors therefore present exact calculations to approximate feature importances for certain settings (i.e. normally distributed data).

2.2.4 HSIC - Hilbert Schmidt Independence Criteria

Testing independence of two random variables is a challenging task due to the unknown non-linear dependence structure in the data. In 2005, the Hilbert-Schmidt independence criterion (Gretton et al., 2005) (HSIC) was proposed as a kernel-based methodology to measure the independence of two multivariate random variables X and Y. HSIC is based on the Hilbert-Schmidt Norm, which is defined as

$$||C||^2_{HS} := \sum_{ij} \langle Cv_i, u_j \rangle^2_{\mathbb{F}}, \tag{2.20}$$

where $C : \mathcal{G} \to \mathcal{F}$ is a linear operator and v_i, u_j are the orthonormal bases of $\mathcal{G}$ and $\mathcal{F}$, respectively. Now, given the cross-covariance operator

$$\begin{aligned} C_{x,y} &:= \mathbb{E}_{x,y}[(\phi(x) - \mu(x)) \otimes (\Psi(y) - \mu(y))] \\ &= \mathcal{E}_{x,y}[\phi(x) \otimes \Psi(y)] - \mu(x) \otimes \mu(y), \end{aligned}$$

where the tensor product operator $f \otimes g : \mathcal{G} \to \mathcal{F}$ is defined as

$$(f \otimes g)h := f\langle g, h\rangle_{\mathcal{G}} \quad \forall h \in \mathcal{G},$$

the Hilbert-Schmidt Independence Criterion is defined as the Hilbert-Schmidt norm of the cross-covariance operator:

$$\begin{aligned} HSIC(X, Y) := \|C_{XY}\|^2 = \mathbb{E}[k(X, X')l(Y, Y')] \\ -2\mathbb{E}[\mathbb{E}_X[k(X, X')]\mathbb{E}_Y[l(Y, Y')]] + \mathbb{E}[k(X, X')]\mathbb{E}[l(Y, Y')] \end{aligned}$$

where k and l are kernel functions.
For practical purpose, the authors introduce an empirical estimate of this measure, which is defined as follows.

Definition 6 (Empirical HSIC) *Let $Z := \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathcal{X} \times \mathcal{Y}$ be a series of m independent observations drawn from $p_{x,y}$. An estimator of HSIC is defined as*

$$HSIC_{emp}(Z, \mathcal{F}, \mathcal{G}) := (m - 1)^{-2} tr(KHLH), \tag{2.21}$$

where $H, K, L \in \mathbb{R}^{m \times m}$, $K_{i,j} := k(x_i, x_j)$,$L_{ij} := l(y_i, y_j)$ and $H_{ij} := I - \frac{1}{m}11^\top$,

where $I \in \mathbb{R}^{m \times m}$ is the identity matrix and 1 is a $m \times 1$ vector of ones.
The empirical estimate of HSIC is the sum of the squared singular values of the cross-covariance operator. This kernel-based approach for detecting dependences between random variables is simple and no user-defined regularization is needed, which may be well used in many practical applications, such as Independent Component Analysis (ICA), Maximum Variance Unfolding (MVU), feature extraction. Furthermore, a fast computation is guaranteed due to the exponential convergence. In a nutshell, HSIC measures the dependence between random variables. The empirical version of HSIC tests the independence of two random variables X and Y based on a sample of observed pairs (x_i, y_i), which is practically easy to apply.

2.2.5 LRP - Layer-wise Relevance Propagation

In the following we present the Layer-wise Relevance Propagation (LRP) framework, which is a method that enables to interpret the decision of a complex non-linear classification algorithm for a specific given input, such as an image. The method, presented in Bach et al. (2015) and Lapuschkin et al. (2016b), has been successfully applied to identify the responsible regions of a given image regarding its classification decision. In the first place, it was designed for deep neural networks, where it, as

the name suggests, propagates the prediction of a specific sample backwards through the network by decomposing it down to the relevance scores over the samples input dimensions.
Let x be an image an $f(x)$ the classifier decision. The idea of LRP is to decompose the classifier output by allocating each pixel p of x a relevance score $R_p^{(1)}$:

$$f(x) \approx \sum_p R_p^{(1)}, \tag{2.22}$$

where the number in the upper index of $R_p^{(1)}$ indicates the respective layer of the network. The relevance of each node j at the network in layer $(l+1)$ is given by $R_j^{(l+1)}$. Thus, with a maximum number of L layers the classifier output is equivalent to the relevance of the last layer: $f(x) = R_1^{(L)}$. The information flow between the layers is handled by messages $R_{i\leftarrow j}^{l,l+1}$ sent from node j from layer $(l+1)$ to node i from the underlying layer. Messages have to fulfill the condition

$$R_j^{(l+1)} = \sum_{i\in(l)} R_{i\leftarrow j}^{l,l+1}, \tag{2.23}$$

which leads to a total relevance of $R_i^{(l)} = \sum_{j\in(l+1)} R_{i\leftarrow j}^{l,l+1}$, for node i at layer (l). There are various possibilities to define the message process. The one we will use in this thesis, the so called $\epsilon - rule$, is given by

$$R_{i\leftarrow j}^{l,l+1} = \frac{z_{ij}}{z_j + \epsilon \text{sign}(z_j)} R_j^{(l+1)}, \tag{2.24}$$

where $z_{ij} = (w_{ij}x_i)^p$ and $z_j = \sum_k \in (l) z_{kj}$. The so called "stabilizer" ϵ is a small scalar with the purpose of avoiding numerical degenerations for very small values of z_j. The pixel-wise obtained relevances of the image x can be visualized in form of a heatmap. (See also Lapuschkin et al. (2016a), Montavon et al. (2017), and Kindermans et al. (2017).

2.3 Data

The proposed methods in this thesis are evaluated on various data sets in the empirical evaluation sections of the following chapters. Some methods are evaluated on same data sets, which is why we introduce them once in the fundamentals in order to prevent a multiple explanation of the same data.

2.3.1 Human Splice Sites

The human splice site data set, which is publicly available[1], consists of 15 million DNA sequences, each of length=141 nucleotides. For the positive labeled sequences, the true splice site is located at position=45 with length=20, whereby there is no splice

[1] http://www.fml.tuebingen.mpg.de/raetsch/projects/lsmkl

site in the negative labeled sequences. For the empirical evaluations in this thesis, we used one million sequences, with a ratio=0.25 of positives/(positives+negatives)). For a detailed introduction into the biological background and the process of splicing see Appendix A.1.

2.3.2 USPS Data

The United States Postal Service (USPS) data set (Hastie et al., 2009; Hull, 1994), are scannings of handwritten digits from envelopes by the U.S. Postal Service. Originally the digits were binary and of different sizes and orientations. For the purpose of handwritten digit recognition with machine learning, the images were distorted and normalized to 16 times 16 grayscale images (LeCun et al., 1990). The data set includes 9298 images of handwritten digits with the corresponding label ([0-9]), encoded through gray scale values ranging in $[-1, 1]$. For illustration, in Figure 2.8 the first five samples of the training set are shown. Each image consists of $16 \times 16 = 256$ pixels.

Fig. 2.8: Illustration of the USPS data set. *The first five samples of the USPS training set are plotted.*

The data is split into a train and test set, where 7291 samples where used for training and the remaining 2007 samples build the test set.

2.3.3 MNIST Data

The Modified National Institute of Standards and Technology (MNIST) database is a large database of handwritten digits, including a training set of 60.000 data points and a test set including 10,000 points (LeCun et al., 1998). Each of these grayscale images consists of 28×28 pixels and has a related label $0 - 9$. The data is publicly available[2].

2.4 Validation Strategies

An appropriate validation strategy is essential in order to measure the qualitative and informative impact of a method. A measure of the respective performance is also important when comparing multiple methods against each other. One widely used validation strategy is the accuracy measure (ACC), where the true positive (TP)

[2] http://yann.lecun.com/exdb/mnist/

and the true negative (TN) sample were added and divided by the total amount of samples:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}, \tag{2.25}$$

where FP and FN are the false positive and false negative samples. In experiments, where the number of positive and negative samples is highly unbalanced, the accuracy measure is biased and may not be informative. Especially in the field of bioinformatics, where the extraction of positive samples is expensive, the ratio of positive to negative samples is often less than one to 1000. Therefore, researchers would use the precision (also called positive predictive value) for validation, which only considers the ratio between the TP and the total amount of positive samples.
Regarding one key issue of this thesis, namely the interpretation of machine learning algorithms, the validation strategy is not evident. Validating the extracted motifs from biological data is often a problem that is left to the biologists due to the non existing ground truth motif. However, some real-world data exist, for which the ground truth motifs are known, such that we can apply the following validation strategy.

2.4.1 Motif reconstruction quality

As a measure of the motif reconstruction quality (MRQ), we employ the same score as presented in Sandelin et al. (2003). When comparing equally sized matrix models, this scoring reduces to the simple formula

$$\text{MRQ} = \sum_{p=1}^{k} \left[\frac{1}{k} - \frac{1}{2k} \sum_{c \in \{A,C,G,T\}} (t_{cp} - r_{cp})^2 \right], \tag{2.26}$$

where t is the underlying true motif and r the comparative one, that was predicted or found by some method.

2.4.2 Pixel Flipping

In the field of computer vision, no ground truth image is available, which would provide the most important pixels or pixel patches in advance. Therefore, for validation, we follow the Most Relevant First (MoRF) strategy and calculate its area over the MoRF perturbation curve (AOPC) as proposed in Samek et al. (2017). In a nutshell the approach works as follows: The indices of a given heatmap H are ordered according to their importances $O = (r_1, \ldots, r_L)$, where r_1 is the index of the highest value in H, i.e., the most important feature. Then, step wise, the function g removes particular position information in x_{MoRF}^{k-1} according to the index r_k and randomly replaces it with samples drawn from a uniform distribution. The recursive formula is as follows:

$$x_{MoRF}^{k} = g(x_{MoRF}^{k-1}, r_k), \quad 1 \leq k \leq L,$$

where

$$x_{MoRF}^{0} = x.$$

Computing the classifier decision score of x^k_{MoRF} in each step k leads to a visualization of the dependence between classifier performance and the impact of the perturbed entries. The quantity of interest is computed as the area over the MoRF perturbation curve for the $k-$th perturbation step

$$AOPC_k = \frac{1}{k}\sum_{i=1}^{k} s(x^0_{MoRF}) - s(x^i_{MoRF}), \tag{2.27}$$

where s is the trained classifer.

Chapter 3

SVM2Motif - Extracting Motifs by Mimicking POIMs

Identifying discriminative motifs underlying the functionality and evolution of organisms is a major challenge in computational biology. Machine learning approaches such as support vector machines (SVMs) achieve state-of-the-art performances in genomic discrimination tasks, but—due to their black-box characters—motifs underlying the decision functions are largely unknown. As a remedy, positional oligomer importance matrices (POIMs) allow us to visualize the significance of position-specific subsequences. Although being a major step towards the explanation of trained SVM models, they suffer from the fact that their size grows exponentially in the length of the motif, which renders their manual inspection feasible only for comparably small motif sizes, typically $k \leq 5$.

3.1 motifPOIM

In this section, we extend the work on positional oligomer importance matrices, by presenting a new machine-learning methodology, entitled motifPOIM, to extract the truly relevant motifs—regardless of their length and complexity—underlying the predictions of a trained SVM model. The proposed framework thereby considers the motifs as free parameters in a probabilistic model, a task which can be phrased as a non-convex optimization problem. The exponential dependence between the POIM size and the motif length poses a major numerical challenge, which is addressed by an efficient optimization framework that allows the identification of possibly *overlapping* motifs consisting of up to hundreds of nucleotides. The efficacy of the proposed approach is demonstrated on synthetic data sets as well as on a real-world *human* splice site data set. Furthermore, we evaluate the proposed methodology on the USPS data set, which is easy to interpret for humans. Parts of the methods and results were published in Vidovic et al. (2015a) and Vidovic et al. (2015b).

3.1.1 Motivation

In the field of bioinformatics, major technological advances in sequencing techniques within the past decade have facilitated a deeper understanding of the mechanisms underlying the functionality and evolution of organisms. Considering the pure size of a genome, it comes, however, at the expense of an enormous amount of data that demands for stand-alone and computationally efficient methods in, e.g., genomic

discrimination tasks[1]. One of the most accurate approaches in genomic discrimination tasks consist in the support vector machine (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995; Zien et al., 2000; Müller et al., 2001) along with the use of a weighted degree string (WD) kernel (Rätsch et al., 2007; Ben-Hur et al., 2008; Rätsch and Sonnenburg, 2004; Sonnenburg et al., 2002), which we introduced in Section 2.2.1. In a nutshell, the WD-kernel is a similarity measure between two DNA sequences, breaking them into all possible subsequences up to a length L and counting the number of matches. The WD-kernel SVM has been shown to achieve state-of-the-art prediction accuracies in many genomic discrimination tasks such as, e.g., transcription start site detection (Sonnenburg et al., 2006b)—achieving the winning entry in the international comparison by (Abeel et al., 2009) of 19 competing machine-learning models—and splice site detection (Sonnenburg et al., 2007). Efficient implementations, such as the one contained in the SHOGUN machine-learning toolbox (Sonnenburg et al., 2010), which employs effective feature hashing techniques (Sonnenburg and Franc, 2010), have been applied to problems where millions of sequences, each containing thousands of nucleotides, are processed at the same time (Sonnenburg et al., 2006a).
Unfortunately, due to its black-box character, biological factors underlying the SVM's prediction such as promoter elements and transcription start sites—the so-called *motifs* (illustrated in Fig. 3.1)—are largely unknown. A first step towards the identification of motifs underlying the functionality of organisms is achieved by POIMs (introduced in Section 2.2.1), which assign each *positional oligomer* (PO) y of length l starting at position j with an importance score $\mathrm{POIM}_{j,y} \sim \mathbb{E}[s(\mathcal{X})|\mathcal{X}[j]^l = y]$, which allows us to visualize the significance of the particular POs as illustrated in Fig. 2.6.

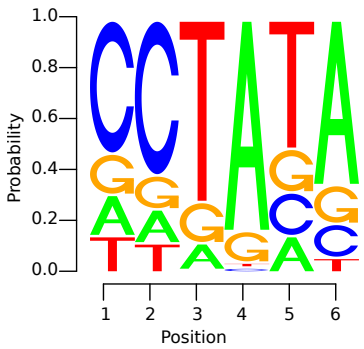


Fig. 3.1: Example of a motif*—i.e., an "interesting" subsequence of the DNA—illustrated as a* positional weight matrix *(PWM): the size of a letter indicates the probability of occurrence of the corresponding nucleotide at a certain position in the motif. The likeliest nucleotides are arranged top down.*

Although being a major step towards the explanation of trained SVM models, POIMs suffer from the fact that their size grows exponentially with the length of the motif, which

1. renders a feasible computation only for rather small motif sizes, typically $k \leq 12$ (see Fig. 3.2 for exemplary execution times)

[1] An introduction to the biological background is given in Appendix A.1

2. hampers manual inspection (in order to determine candidate motifs) already for rather small motif sizes such as $k \approx 5$ and is prohibitive for $k \geq 10$. For example, a POIM of order $k = 5$ contains, at each position, already $4^5 \approx 1,000$ oligomers that a domain expert has to manually inspect. Slightly increasing the motif length to $k = 10$ leads to an unfeasible amount of $4^{10} \approx 1,000,000$ subsequences per position in the POIM.

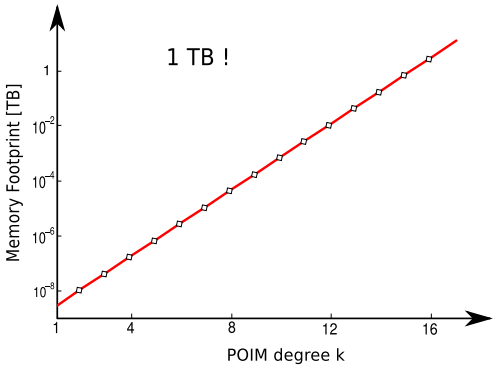


Fig. 3.2: Memory footprint for POIMs of oligomer length k***.*** *Note that the plot is in semi-logarithmic scale and thus showing an exponential growth for increasing oligomer lengths rendering a direct approach incomputable for even small* $k \geq 12$*.*

In this chapter, we tackle the problem of obtaining motifs from the output of an SVM via the use of POIMs from a different perspective. In a nutshell, our approach is the other way around: we propose a probabilistic framework to reconstruct, from a given motif, the POIM that is the most likely to be generated by the motif. By subsequently minimizing the reconstruction error with respect to the truly given POIM, we can in fact optimize over the motif in order to find the one that can reconstruct the POIM best. The latter poses a substantial numerical challenge due to the extremely high dimensionality of the feature space. Fig. 3.3 illustrates our approach.
The main contributions of this chapter can be summarized as follows:

1. Advancing the work of Sonnenburg et al. (2008) on POIMs, we propose a novel probabilistic framework to finally go the full way from the output of a state-of-the-art WD-kernel SVM via POIMs to the relevant motifs truly underlying the SVM predictions.

2. To deal with the immense size of the feature space associated with the WD-kernel, we propose a very efficient numerical framework based on numerous speed-ups such as bit-shift operations, highly efficient scalar multiplications as well as advanced sequence decomposition techniques, and provide a free open-source implementation thereof[2].

[2] https://github.com/mcvidomi/poim2motif.git

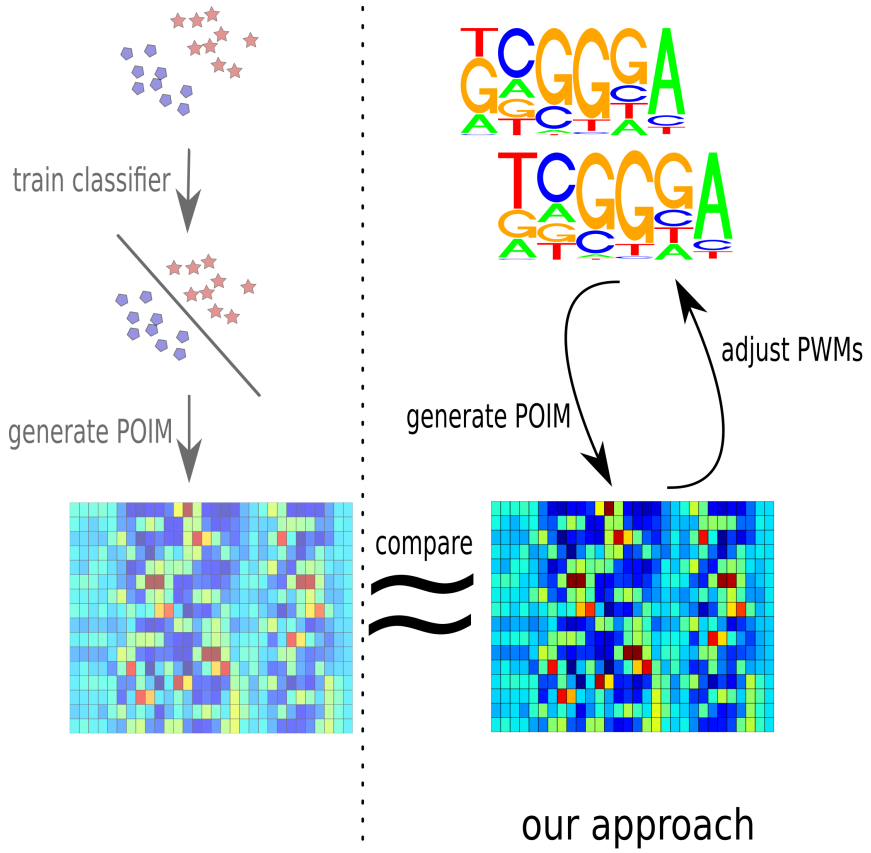


Fig. 3.3: Illustration of the proposed framework —SVM2Motif— to extract motifs from a trained SVM model: *In a first step, a POIM is computed corresponding to the trained SVM (shown on the right, from top to bottom). Then a motif approximately corresponding to the POIM is determined by associating each candidate motif (illustrated in the top right) with a motifPOIM (shown in the bottom right) via a probabilistic model and then minimizing the reconstruction error (indicated by a $\approx$ symbol) by a feedback loop (observe the curved errors on the right) with respect to the truly computed POIM (shown on the bottom left).*

3. Our approach is able to even find *overlapping* motifs consisting of up to hundreds of nucleotides, while previous approaches are limited to either comparably short or contiguous motifs.

4. We demonstrate the efficiency and efficacy of our approach on both synthetic data sets as well as a *human* splice data set, evaluated with the Motif Reconstruction Quality (MRQ) measure (see Section 2.4.1).

This chapter is structured as follows: First, we introduce the proposed probabilistic methodology—motifPOIM—for approximately determining the motif underlying the observed POIM at hand. Following this, we propose a numerical framework —SVM2Motif— for solving the corresponding non-convex optimization problem by the use of efficient sequence computation techniques such as bit shifts. We evaluate the

proposed methodology empirically both on controlled synthetic data as well as real-world *human* splice data. Furthermore, the proposed methodology is evaluated on the USPS data set, which is easy to interpret for the naive reader. Finally, we conclude and discuss starting points for future work.

3.1.2 Probabilistic Model

We segment the method in its four substantial steps, which lead to a non-convex optimization problem:

1. *motif definition:* The proposal motif is defined as probabilistic positional motif (PPM), which is a tuple $m_k := (r, \mu, \sigma)$, where $r \in \mathbb{R}^{|\Sigma| \times k}$ is a stochastic matrix (PWM, positional weight matrix) that codes for the motif and $\mu, \sigma \in \mathbb{R}$.

2. *motif weight function:* A PPM induces a probabilistic model. Given μ and σ as the starting position with its variance of the PPM, the Gaussian probability function for the starting position is

$$P^1_{(z,i)}(m_k) := \frac{1}{\sqrt{2\pi}\sigma} exp\left(- \frac{(i-\mu)^2}{2\sigma^2} \right). \tag{3.1}$$

   Furthermore, the probability of the motif sequence itself is given by the product of its PWM entries

$$P^2_{(z,i)}(m_k) := \prod_{\ell=1}^{k} r_{z_\ell,\ell}\,. \tag{3.2}$$

   Combining P^1 and P^2, the probability for each oligomer at each position

$$v_{(z,i)}(m_k) := P^1_{(z,i)}(m_k) P^2_{(z,i)}(m_k) \tag{3.3}$$

   can be assembled and gives us a weight vector similar to the weight vector of the SVM.

3. *motif scoring function:* Thus, we are able to resemble the SVM scoring function as a motif scoring function:

$$\bar{s}(x|m_k) := \sum_{i=1}^{L-k+1} v_{(x[i]^k,i)}(m_k)\,. \tag{3.4}$$

4. *motifPOIM formula:* Consequently, we define in Definition 7 a motifPOIM R in analogy to the POIM Q (see Definition refdef:poim).

Definition 7 (motifPOIM) *Given a motif scoring function $\bar{s}$ as defined in Equation (3.4), then for each possible k-mer y at position j we define a motifPOIM score as*

$$R_{y,j}(m_k) := \mathbb{E}[\bar{s}(X|m_k)|X[j]^k = y] - \mathbb{E}[\bar{s}(X|m_k)]\,, \tag{3.5}$$

which results, applied successively, in the motifPOIM R.

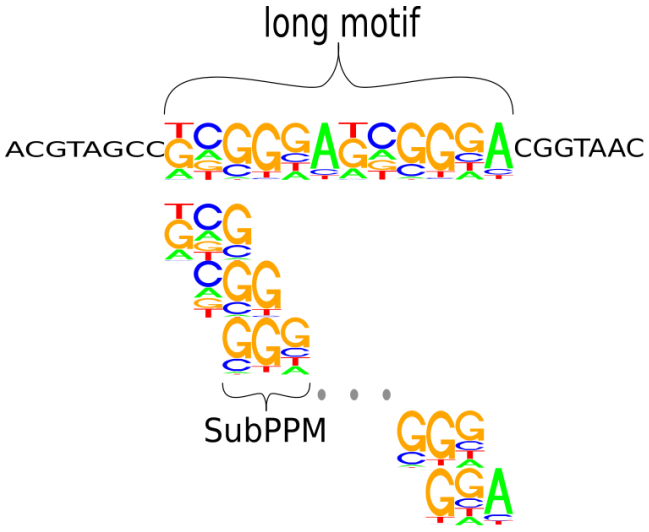
Our overall aim is, by optimizing over the motifPOIM R, to approximate the original POIM (cf. also the illustration in the introduction, given by Fig. 3.3). An interesting fact here is that, since computing motifPOIMs for longer PPMs ($m_k, k > 5$) is computationally expensive, we may use motifPOIMs of small orders $\tilde{k} \in \{2, 3\}$, although, this is no restriction of the motif length, as we model a PPM of length $k \geq \tilde{k}$ as a set of overlapping SubPPMs, which we define as follows.

Definition 8 (SubPPMs) *A PPM of length k is modeled as a set of D SubPPMs, $D := k - \tilde{k} + 1$ with length $\tilde{k} \leq k$, where SubPPMs are defined by*

$$\tilde{m}_d(m_k, \tilde{k}) := (\tilde{r}, \tilde{\mu}, \sigma), \ \forall \ d = 0, \ldots, D-1 \ .$$

Here, $\tilde{\mu} := \mu + d$ and $\tilde{r} := r[d, d + \tilde{k} - 1]$, where $r[d, d + \tilde{k} - 1]$ is the d-th until the $(d + \tilde{k} - 1)$-th column of the PPMs PWM r.

The basic idea is illustrated in Fig. 3.4, where we divide a PPM into a set of SubPPM. Instead of computing a motifPOIM for the PPM, we now compute a set of D motifPOIMs for the smaller overlapping SubPPMs.



,

Fig. 3.4: Illustration of the SubPPM approach: *instead of computing possibly intractable POIMs for long motifs directly, we decompose each of the longer motifs (here: a single motif of length 12) into smaller overlapping, conforming subsequences of length $\tilde{k}$ (in the figure: $\tilde{k} = 3$). This approach allows us to reconstruct motifs of arbitrary length using low dimensional POIMs, rendering the reconstruction of very large, possibly overlapping motifs computationally feasible.*

3.1.3 Numerical Methods

In this section, we introduce an efficient numerical framework for the extraction of motifs from POIMs by mathematical optimization. The core idea is to determine

a motif m_k with an according motifPOIM $R(m_k)$ that approximates the original POIM Q_k. To this end, let us introduce some notation. Let $\mathcal{K} \subset \mathbb{N}$ be the set of all motif lengths to be considered and $k_{\max} = \max_{k \in \mathcal{K}} k$ the maximum length. The vector $T \in \mathbb{N}_0^{k_{\max}}$ contains the number of PPMs for each motif length, where $T_k, k \in \mathcal{K}$ is the given number of PPMs of length k. For example, when $\mathcal{K} = \{2, 4, 10\}$ and $T = (0, 6, 0, 2, 0, 0, 0, 0, 0, 2)$, then the goal is to find 6 PPMs of length 2, 4 PPMs of length 4, and 2 PPMs of length 10. Our optimization method is as follows: given the set $\mathcal{K}$ and the vector T, we randomly initialize the PPMs $m_{k,t}\ t = 1, \ldots, T_k\ ,\ k \in \mathcal{K}$ and generate a set of motifPOIMs for the SubPPMs $\tilde{m}_d(m_k, \tilde{k}),\ d = 0, \ldots, D-1$. The optimization variables are all $T_k, k \in \mathcal{K}$ PPMs. For obtaining the priorities of the PPMs we weight the PPMs by $\lambda_{k,t},\ t = 1, \ldots, T_k\ ,\ k \in \mathcal{K}$ and additionally optimize over the weights. Hence, the optimization variables are:

- PPM $m_{k,t} = (r_{k,t}, \mu_{k,t}, \sigma_{k,t}), \qquad t = 1, \ldots, T_k\ ,\ k \in \mathcal{K}\ ,$
  where
  $\mu_{k,t} \in \{1, \ldots, L-k+1\}, \qquad t = 1, \ldots, T_k\ ,\ k \in \mathcal{K}$
  $\sigma_{k,t} \in [\epsilon, k], \qquad t = 1, \ldots, T_k\ ,\ k \in \mathcal{K}$
  $r_{k,t} \in [\epsilon, 1]^{4 \times k}, \quad t = 1, \ldots, T_k\ ,\ k \in \mathcal{K}$

- weight of $m_{k,t}$
  $\lambda_{k,t} \in [0, W], \qquad t = 1, \ldots, T_k\ ,\ k \in \mathcal{K}, W \in \mathbb{R}^+\ .$

A PPM generates a motifPOIM, which is given by the sum of D motifPOIMs generated by its SubPPMs. The sum of the weighted motifPOIMs, $\lambda_{k,t} R(m_{k,t}),\ t = 1, \ldots, T_k$, should estimate the POIM $Q_{\tilde{k}}$ for each $k \in \mathcal{K}$. The optimization problem is now a minimization of the distance between the sum of the motifPOIMs and the original POIM, which leads to a non-convex optimization problem with the following objective function:

$$\Pi(\eta) = \frac{1}{2} \sum_{k \in \mathcal{K}} \sum_{y \in \Sigma^{\tilde{k}}} \sum_{j=1}^{L} \left(\sum_{t=1}^{T_k} \lambda_{k,t} \sum_{d=0}^{D-1} R_{y,j}(\tilde{m}_d(m_{k,t}, \tilde{k})) - Q_{\tilde{k},y,j} \right)^2 , \tag{3.6}$$

where $\eta = (m_{k,t}, \lambda_{k,t}, \tilde{k})_{t=1,\ldots,T_k, k \in \mathcal{K}}$.
The associated constrained non-linear optimization problem is thus as follows:

$$\begin{aligned} &\min_{(m_{k,t}, \lambda_{k,t})_{t=1,\ldots,T_k, k \in \mathcal{K}}} \Pi(\eta) \\ &\text{subject to} \\ &\epsilon \leq \sigma_{k,t} \leq k, && t = 1, \ldots, T_k\ ,\ k \in \mathcal{K} \\ &1 \leq \mu_{k,t} \leq L - k + 1, && t = 1, \ldots, T_k\ ,\ k \in \mathcal{K} \\ &0 \leq \lambda_{k,t} \leq \infty, && t = 1, \ldots, T_k\ ,\ k \in \mathcal{K} \\ &\epsilon \leq r_{k,t,o,s} \leq 1, && t = 1, \ldots, T_k\ ,\ k \in \mathcal{K}, \\ & && o = 1, \ldots, |\Sigma|, s = 1, \ldots, k\ , \sum_{o=1}^{|\Sigma|} r_{k,t,o,s} = 1, \end{aligned} \tag{3.7}$$

where ϵ is the machine precision. Note that for the sake of optimization efficiency we relax the integer constraint on the motifs start positions in the sense that we optimize over positive real numbers. The objective function $\Pi(\eta)$ is defined on the compact set U, since all parameters are defined in a closed and bounded, convex space. Consequently, if U is not empty, $\Pi(\eta)$ is a continuously differentiable function, since its conforming parts, that is, the Gaussian function and the product of the PWM entries, all are continuously differentiable. Thus, the global minimum of the optimization problem (3.7) is guaranteed to exist. Due to the non-convex nature of (3.7), however, there may exist multiple local minima.

Efficient Computation

To allow an efficient numerical optimization of (3.7), we first translate the motifPOIM formula (3.5) in another, equivalent form, similar as in Sonnenburg et al. (2007). To this end, note that the expected value of $\bar{s}(\mathcal{X}|m_k)$ for the given weight vector $v(m_k)$ and a random variable $\mathcal{X} \in \Sigma^L$ is given by:

$$\mathbb{E}[\bar{s}(\mathcal{X}|m_k)] \quad = \quad \frac{1}{|\Sigma^L|} \sum_{x \in \Sigma^L} \bar{s}(x; m_k)\,.$$

It holds that

$$\begin{aligned}
\mathbb{E}[\bar{s}(\mathcal{X}|m_k)] &= \frac{1}{|\Sigma^L|} \sum_{x \in \Sigma^L} \sum_{l=1}^{k} \sum_{i=1}^{L-l+1} v_{(x[i]^l,i)}(m_k) \\
&= \sum_{l=1}^{k} \sum_{i=1}^{L-l+1} \frac{1}{|\Sigma^L|} \sum_{x \in \Sigma^L} v_{(x[i]^l,i)}(m_k) \\
&= \sum_{l=1}^{k} \sum_{i=1}^{L-l+1} \frac{1}{|\Sigma^l|} \sum_{z \in \Sigma^l} v_{(z,i)}(m_k) \\
&= \sum_{l=1}^{k} \sum_{z \in \Sigma^l} \sum_{i=1}^{L-l+1} v_{(z,i)}(m_k) \mathbb{P}(\mathcal{X}[i]^l = z)\,. \qquad (3.8)
\end{aligned}$$

Hence the conditioned expectation is almost equivalent to (3.8), except the probability term that is given by the conditioned probability conditioned that y is at position j:

$$\mathbb{P}(\mathcal{X}[i]^l = z | \mathcal{X}[j]^k = y)\,. \qquad (3.9)$$

We now consider this probability term and its effect on the summation in (3.5). To this end, we introduce the following notation as in Sonnenburg et al. (2007).

Definition 9 *Two positional oligomers (z,i) and (y,j) of length l and k are independent if and only if they do not share any position; in this case we write $(y,j) \not\prec (z,i)$ and $(y,j) \prec (z,i)$ otherwise (i.e., when they are dependent). If they are dependent and also agree on all shared positions we say they are* compatible *and we write $(y,j) \precsim (z,i)$ (and $(y,j) \not\precsim (z,i)$ if they are not compatible).*

According to the cases discussed in the above definition, the conditioned probability term can take the following values:

$$\mathbb{P}(\mathcal{X}[i]^l = z|\mathcal{X}[j]^k = y) = \begin{cases} \frac{1}{|\Sigma^l|} & \text{if } (y,j) \not\asymp (z,i) \\ 0 & \text{if } (y,j) \precnsim (z,i) \\ \frac{|\Sigma^c|}{|\Sigma^l|} & \text{if } (y,j) \precsim (z,i) \end{cases}, \tag{3.10}$$

where c is the number of shared and compatible positions of two positional oligomers:

$$c\big((y,j),(z,i)\big) = \begin{cases} l - |i-j| & \text{if } i < j \text{ and } (y,j) \precsim (z,i) \\ l & \text{if } i = j \text{ and } (y,j) \precsim (z,i) \\ k - |i-j| & \text{if } i > j \text{ and } (y,j) \precsim (z,i) \\ 0 & \text{else.} \end{cases}.$$

Taken the case $(y,j) \not\asymp (z,i)$, the probability terms in the motifPOIM formula (3.5) subtract to zero, so that the positional oligomer (z,i) is not considered in the sum $R_{y,j}(m_k)$. Hence, in order to compute $R_{y,j}(m_k)$, it is sufficient to sum over two positional oligomer sets, where one contains all (z,i) with $(y,j) \precsim (z,i)$, $\mathcal{I}^{\precsim}_{(y,j)}$, and the others contains all (z,i) with $(y,j) \precnsim (z,i)$, $\mathcal{I}^{\precnsim}_{(y,j)}$:

$$\begin{aligned} R_{y,j}(m_k) \quad = \quad & \sum_{(z,i) \in \mathcal{I}^{\precsim}_{(y,j)}} v_{(z,i)}(m_k)\big(\frac{|\Sigma^c|}{|\Sigma^k|} - \frac{1}{|\Sigma^k|}\big) \\ & + \sum_{(z,i) \in \mathcal{I}^{\precnsim}_{(y,j)}} v_{(z,i)}(m_k)\big(-\frac{1}{|\Sigma^k|}\big)\big), \end{aligned} \tag{3.11}$$

where $\mathcal{I}^{\circ}_{(y,j)} := \Big\{(z,i) \in \Sigma^{|y|} \times \big\{1,\dots,L-|y|+1\big\} | (y,j) \circ (z,i)\Big\}$ and $\circ \in \{\precsim, \precnsim\}$.

Numerical Speed-ups In addition to the speed-up achieved by the above reformulation of the problem, we can additionally save time in the motifPOIM computation by exploiting bit shift operations as follows. With the help of the dependence sets $\mathcal{I}^{\precsim}_{(y,j)}$ and $\mathcal{I}^{\precnsim}_{(y,j)}$ we know all the dependent and compatible positional oligomers of a single positional oligomer (y,j). Fig. 3.5 exemplarily illustrates the dependent and compatible oligomers z of $y = TAC$.
The core idea leading to the numerical speed-up is as follows: In each (y,j) we consider the two dependence sets. However, the fact is that an oligomer y has completely the same dependent and compatible oligomers z at each position in the sequence. Thus, a dependent set containing all dependent and compatible z of y is the same for all positions $i = 1,\dots,L$. The trick is to generate a dependency matrix $\mathcal{A}$ (see Definition 3.12) for a single y once, which can then be use at every sequence position without the need of recalculation. This matrix contains the probability terms of the motifPOIM formula since they do not change for y over the positions, saving at least $|\Sigma^k|(2(k-1)+1)$ complex computations per position. For each position j we now create a weight matrix $\mathcal{C}^j$ of same size, which contains all the weights $v_{(z,i)}(m_k)$ for

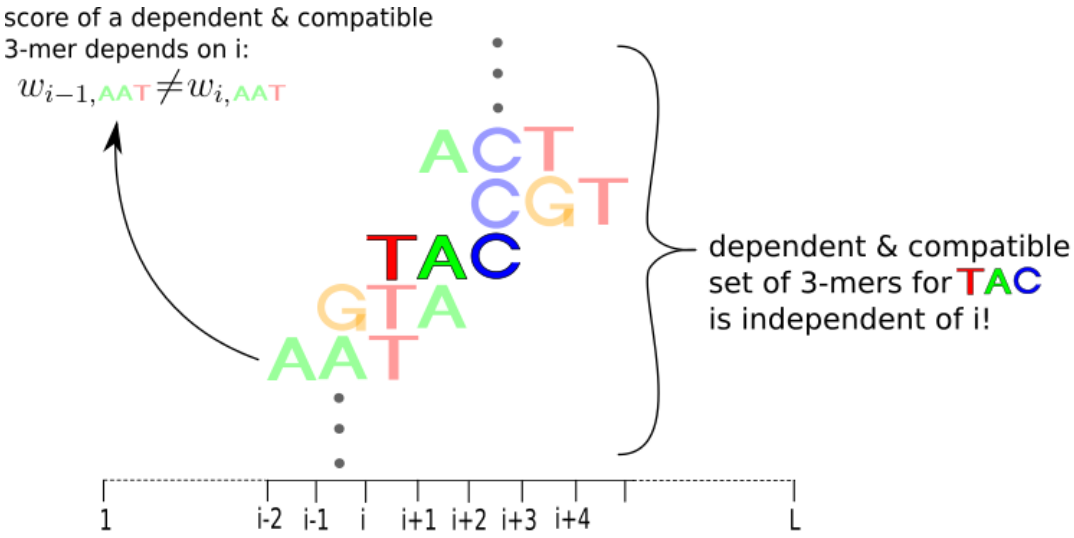


Fig. 3.5: Illustration of the definition of* dependent *and* compatible *oligomers (cf. Definition 9). *We say that two positional oligomers are dependent when they overlap each other. If they additionally agree on all shared positions, we say that they are compatible. In this figure, the positional oligomers* (TAC, i) *and* $(AAT, i-2)$ *are dependent and compatible since both of them contain the letter* T *at position* i*. Whereas the positional 3-mers* (TAC, i) *and* $(AAG, i-2)$ *are dependent but not compatible.*

the entries in $\mathcal{A}$ for a specific position j. Finally, the dot product of $\mathcal{A}$ and $\mathcal{C}^j$ replaces the long motifPOIM formula (3.11) and we achieve a faster computation speed.
Due to the fact that dependent positional oligomers overlap each other, a dependent k-mer z of the k-mer y could have a maximal distance of $k-1$ from y. Hence, we have to consider the oligomers z with a maximum distance of $k-1$ position next to both sites of y and the position of y itself. That yields to the dependence set:

$$\mathcal{I}_y(k) = \left\{ (z,i) \in \Sigma^k \times \{1, \ldots, 2(k-1)+1\} \right\}.$$

The dependent matrix $\mathcal{A}(y)$ is defined on $\mathcal{I}_y(k)$ as a matrix of size $4^k \times (2(k-1)+1)$ and contains the positional oligomer probability terms of the motifPOIM formula as entries:

$$\mathcal{A}_{z,i}(y) = \begin{cases} \frac{4^c-1}{4^k} & \text{if } (z,i) \precsim (y,k) \\ \frac{-1}{4^k} & \text{else} \end{cases} . \tag{3.12}$$

Furthermore, we create a weight matrix $\mathcal{C}^j$ of same size as A, which contains all weights $v_{(z,i)}(m_k)$ of the entries in $\mathcal{A}$ for a specific position j, so that the dot product of $\mathcal{C}^j$ and $\mathcal{A}$ replaces the sums of the motifPOIM formula (3.11), which speeds up computations considerably. This fact is stated in the following theorem.

Theorem 10 *Let y be a k-mer, m_k the PPM, $v(m_k)$ the motif weight vector, and $\mathcal{A}$ the dependent matrix of y. Introducing a matrix $\mathcal{C}^j(y|m_k)$, which is defined on $\mathcal{I}_y(k)$ as a matrix of same size $\Sigma^k \times (2(k-1)+1)$ as $\mathcal{A}(y)$ and contains all weights of the positional oligomers in $\mathcal{A}(y)$ for the motifPOIM position j as*

$$\mathcal{C}^j_{z,i}(y|m_k) = \begin{cases} v_{(z,i+j-k)}(m_k) & \text{if } 1 \leq j+i-k \leq L \\ 0 & \text{else} \end{cases} , \tag{3.13}$$

then

$$R_{y,j}(m_k) = \langle \mathcal{A}(y), \mathcal{C}^j(y|m_k) \rangle \, . \tag{3.14}$$

Proof 1

$$\begin{aligned} \langle \mathcal{A}(y), \mathcal{C}^j(y|m_k) \rangle &= \sum_{z \in \Sigma^k} \sum_{i=1}^{2(k-1)+1} C^j_{z,i}(y|m_k) A_{z,i}(y) \\ &= \sum_{z \in \Sigma^k} \sum_{i=1}^{2(k-1)+1} v_{(z,i+j-k)}(m_k) A_{z,i}(y) \\ &= \sum_{\mathcal{I}^{\prec}_{(y,k)}} \left(\frac{4^{c((y,k),(z,i))} - 1}{4^k} \right) v_{(z,i+j-k)}(m_k) + \sum_{\mathcal{I}^{\not\prec}_{(y,k)}} \left(\frac{-1}{4^k} \right) v_{(z,i+j-k)}(m_k) \\ &= \sum_{\mathcal{I}^{\prec}_{(y,j)}} \left(\frac{4^{c((y,j),(z,i))} - 1}{4^k} \right) v_{(z,i)}(m_k) + \sum_{\mathcal{I}^{\not\prec}_{(y,j)}} \left(\frac{-1}{4^k} \right) v_{(z,i)}(m_k) \, . \end{aligned}$$

Substituting the last equation into (3.14) gives us (3.11).

The case distinction in Theorem 10 is made since some dependent positional oligomers are placed outside the possible sequence positions. Suppose we compute the weight matrix $C^j_{z,i}(y|m_k)$ for $y = ACT$ at the sequence position $j = 1$. Then there are overlapping 3-mers such as, for example, $(AAA, -1)$ and $(TAC, 0)$, that do not exist in the sequence at all. Thus, they are weighted by zero.
Together with the fact that we implement the algorithm in the PYTHON programming language and use the numpy library for computations, calculations are very fast by using the algorithm shown in Table 3.1.
Another step towards an efficient computation is yield by introducing a confidence interval: The probability distribution over the PPM with starting position μ in the sequence is a Gaussian function. One characteristic of this function is that $99,7$ % of the starting positions are within the confidence interval $[\mu - 3\sigma, \mu + 3\sigma]$. Hence, it suffices to compute the motifPOIM entries for the integer values in the confidence interval and set the other motifPOIM entries to zero. Let $\mathcal{I}_{CO}$ be the set containing all positional oligomers of the confidence interval. A summary is given in Table 3.1. For each $k \in \mathcal{K}$ a motifPOIM R is constructed (see Theorem 10) and the residual between the aforementioned motifPOIM and the SVM POIM Q_k of matching order k is added to the variable iteratively computing the function value.

Data: $m_{k,t} = (r_{k,t}, \mu_{k,t}, \sigma_{k,t}), \lambda_{k,t},\ t = 1, \ldots, T_k, k \in \mathcal{K}$
Result: $\Pi(m_{k,t}, \lambda_{k,t})_{t=1,\ldots,T_k, k\in\mathcal{K}}$
begin
  $f \longleftarrow 0$
  for $k \in \mathcal{K}$ **do**
    $R \longleftarrow \mathbf{0}$
    for $y \in \Sigma^k$ **do**
      Compute $A(y)$ (see Eq. (3.12))
      for $t = 1, \ldots, T_k$ **do**
        for $j \in \mathcal{I}_{CO}$ **do**
          Compute $C^j(y|m_{k,t})$ (see Eq. 3.13)
          $R[y][j] = R[y][j] + (\langle A(y), C^j(y|m_{k,t})\rangle)$ (see Eq.(3.14))
    for $y \in \Sigma^k$ **do**
      for $j = 1, \ldots, L$ **do**
        $\Pi = \Pi + (R[y][j] - Q_k[y][j])^2$ (see Eqn. 3.6)

Tab. 3.1: *Efficient evaluation of Equation* (3.6)

3.1.4 Empirical Analysis

In this section, we analyze our proposed mathematical model (3.7) empirically. After introducing the experimental setup, we evaluate our approach on a synthetic data set, where we fully control the underlying ground truth. Then, we investigate our model on a real *human* splice data set and compare our results to motifs contained in the JASPAR database (Sandelin et al., 2004). Finally, we show that our method is applicable to images as well and shows promising results on the USPS data set, which we introduced in Section 2.3.2.

Overall Experimental Setup

For the SVM training, we use the SHOGUN machine-learning toolbox (Sonnenburg et al., 2010) (available from `http://www.shogun-toolbox.org/`), which contains a C++ implementation of a WD-kernel SVM that is specially designed for large-scale sequence learning problems and provides interfaces to MATLAB, PYTHON, R, and JAVA. The regularization constant C of the SVM and the degree d of the weighted degree kernel are set to $C = 1$ and $d = 20$, which are proven default values.
After SVM training, the POIM Q is generated through the PYTHON script COMPUTE_POIMS.PY, which is included in the SHOGUN toolbox. The PYTHON framework obtains the trained SVM and a (maximal) POIM order $k_{max} = 12$ as parameters and returns all POIMs, i.e., the differential POIM, the maximum POIM, and the regular POIMs $Q_k, k = 1, \ldots, k_{max}$. We set $k_{max} = 7$ in synthetic experiments and $k_{max} = 6$ in real experiments because of memory requirements (storing all POIMs up to an order of 10 requires about 4 gigabytes of space). Note that this is no restriction as our optimization problem (3.7) requires POIMs of order two or three only. Nevertheless,

POIMS of higher order than three can provide additional useful information since they contain prior information about the optimization variables, which we use for a proper initialization, which we refer to as greedy initialization: To efficiently optimize our highly non-convex optimization problem (3.7), an appropriate initialization of the optimization variables is mandatory. Thus, we use the differential POIM (defined in Equation (2.18)) as indicator for extracting the area of interest: we search for points of accumulation of high scoring entries, from which we manually estimate the number of motifs as well as their length and starting position. Thereby we take the whole interval of all highly scoring positions as motif length, where the start position is the first position where all k-mers show a substantial increase in their scores. Once the motif interval is estimated, we select the leading nucleotide from the highest scoring column entry within the interval from the corresponding POIM and initialize the respective PWM entry with a value of 0.7 and 0.1 for non-matches. Indeed, we found this approach to be more stable and reliable than using random initialization. These parameters serve as initialization for our non-convex optimization problem (3.7). To compute a motif from the computed POIMs, we employ the L-BFGS-B Algorithm (Liu and Nocedal, 1989), where the parameters λ and σ both are initialized as 1. An illustration of the so-obtained experimental pipeline is shown in Fig. 3.6.

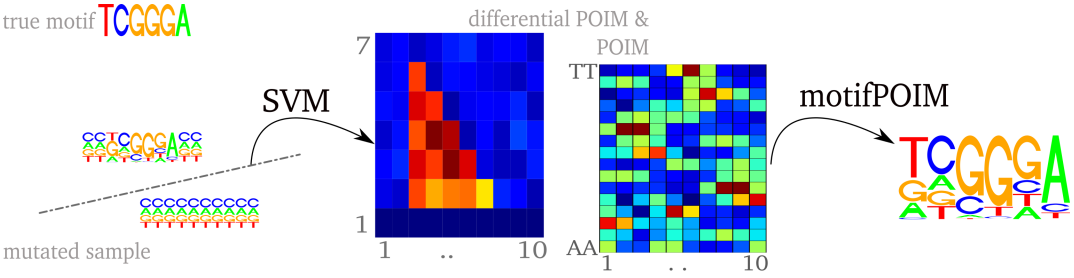


Fig. 3.6: Experimental pipeline of the motif extraction process (from left to right): *given a trained SVM, we construct the corresponding POIM before applying the proposed motifPOIM approach to reconstruct underlying motifs (PWMs). Differential POIMs give reasonably initial values for the length and number of motifs.*

As performance measure, we use the motif reconstruction quality (MRQ), which we introduced in Section 2.4, Equation (2.26), and which is the same performance measure used in the JASPAR SPLICE database (Sandelin et al., 2003).

Synthetic Data Experiments

We first evaluate the proposed methodology on synthetically generated data, where we have full access to the underlying ground truth. This experiment aims successive at demonstrating the ability of our method in finding

1. a single motif
2. a single mutated motif
3. overlapping motifs

4. long motifs.

Data Sets To this end, we generate four sample sets S_1, S_2, S_3, S_4 as follows:

1. The sample S_1 consists of 10,000 DNA sequences of length 30 over the alphabet $\{A, C, G, T\}^{30}$, randomly drawn from a uniform distribution $\mathcal{U}(\Sigma^L)$ over Σ^L. We subsequently modify 25% of the sequences by replacing the positions 11 to 16 by the synthetic target sequence CCTATA. These modified sequences form the positively labeled examples, while the remaining 75% of sequences are assigned to a negative label.

2. The sample S_2 is obtained from S_1 by mutating any of the six conforming nucleotides of the inserted motif with probability p. This models a scenario where a motif is not quite clearly expressed in the data. We realized the sample $S_2 \equiv S_2^p$ for various levels of mutation $p \in [0, 1]$.

3. Similar to S_1, the sample S_3 consists of 10,000 uniformly drawn DNA sequences of length 30, where, in 12.5% of the sequences, we replace the positions 5 to 15 by the positional oligomer (AATCTGGCGGT,5). Similarly, we insert the PO (CAATAGCCTGATGGC,10) into another 12.5% of sequences, resulting in a total of 25% of altered sequences, which are assigned to a positive label (and all other sequences are labeled negatively).

4. The sample S_4 consists of 10,000 uniformly drawn DNA sequences of length 400, where, in 25% of the sequences, we replace the positions 21 to 220 by a positional oligomer of the form $TCGGA\ TCGGA\ TCGGA$ ... with length 200.

Results

Results on the unmutated data set S_1. The results of the realization of this synthetic experiment using training subsets of size n from the base sample S_1 are shown in Fig. 3.7, for various values of n. We can observe from the figure that the reconstruction error decreases as a function of the sample size n already for $n = 100$. The corresponding motif/PWM computed by our approach correctly identifies the true underlying motif sequence as the most likely path in the PWM.

Results on the mutated data set S_2. Furthermore, we realize the very same experiment using the sample $S_2 \equiv S_2^p$ for various levels of mutations. The results are shown in Fig. 3.8. We can observe that, up to a mutation level of 60%, we correctly identify the true underlying motif as being the sequence with the highest probability in the PWM. For more than 70% of mutations in the training data, the performance drops severely. This effect however, is due to a drop of classification performance of the corresponding SVM as can be seen in Table 3.2. Table 3.2 highlights results for an exemplary sample for each level of mutation, to relate SVM classification error to mutation level, and also random PWM initialization strategy (30 runs) to greedy initialization.

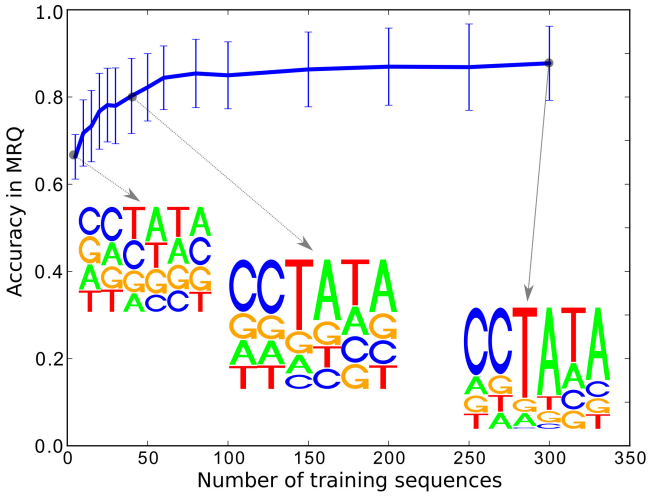


Fig. 3.7: The results of the synthetic experiment for varying SVM training sample size n ***using non-mutated sequences of length*** 30**.** *As expected, the motif is better reconstructed the more training sequences are used for the SVM training. However, as can be seen in the figure, the true motif is picked up early, a tendency that we claim to the robustness of our approach.*

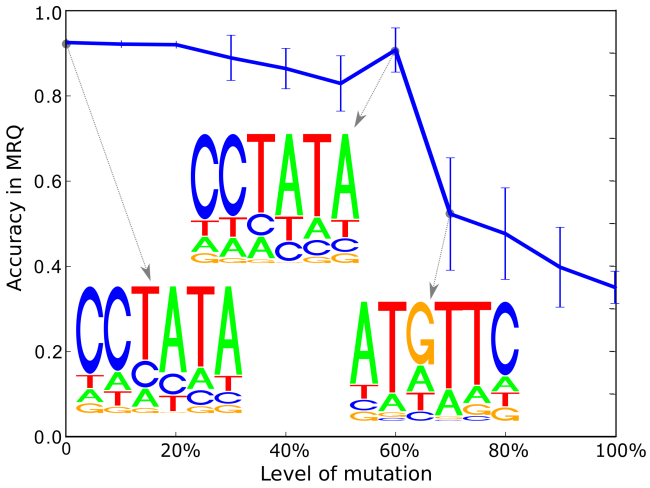


Fig. 3.8: We illustrate the robustness of our approach by plotting the reconstruction errors vs. the mutation level for a fixed amount of training samples. *We observe that even for high mutation levels (e.g. 50%) the motif reconstruction quality (MRQ) is sufficiently good to reconstruct the true underlying motif correctly.*

Results with overlapping motifs, i.e., data set S_3**.** To validate our method for overlapping motifs, we also experiment on the sample S_3. The differential POIM and the POIM of order two resulting from our experimental pipeline are shown in Fig. 3.9 (a) and (b). Interestingly, the two accumulations of entries with high scores indicate that the POIM includes two overlapping motifs. The investigation of these

		Greedy PWM init		Random PWM init	
p	SVM acc	iter	MRQ	iter	MRQ
0.0	0.9987	14	0.93	39±14	0.8±0.1
0.1	0.998	13	0.92	43±12	0.76±0.12
0.2	0.998	13	0.92	40±19	0.77±0.1
0.3	0.9991	14	0.92	45±21	0.74±0.11
0.4	0.996	13	0.92	41±17	0.8±0.06
0.5	0.9989	14	0.92	36±21	0.79±0.07
0.6	0.9944	13	0.92	41±15	0.78±0.05
0.7	0.616	13	0.46	16±6	0.53±0.08
0.8	0.5	13	0.44	15±2	0.56±0.1
0.9	0.5	14	0.35	15±2	0.55±0.07
1.0	0.5	20	0.33	16±3	0.47±0.08

Tab. 3.2: Experimental results for a fixed sample S_1 with no mutation ($p = 0$) and S_2 with various levels of mutation ($p = 0.1, \ldots, 1$). *The proposed greedy initialization of the PWMs is more reliable and stable than randomly initialized PWMs (mean and standard deviations are shown for 30 re-starts), indicated by higher MRQs and less iterations. Furthermore, the SVM classification error is related to the level of mutation and clearly correlated with the motif reconstruction quality (MRQ) of our method, independent of the initialization strategy.*

accumulations is slightly more involved than in the experiment above: we observe, for each motif length $l > 1$, $11 - l + 1$ subsequent cell entries having an extraordinary high score as indicated by light blue, green, orange, or red colors (e.g., length l=7, we observe a block of 5 subsequent entries). Thus, the first discriminative motif starts at position 5 and consists of 11 nucleotides. We can observe a drop at position 10 (notice the dark blue color) indicating the starting position of the second motif. Altogether, the figure indicates that the optimal model parameters are: $\mathcal{K} = \{11, 15\}$, $T_{11} = 1$, $T_{15} = 1$, where $\mu_{11,1} = 5$ and $\mu_{15,1} = 10$. Furthermore, Fig. 3.9 (c) and (d) show the PWM results obtained from our optimization approach. We can observe that, although the two motifs are overlapping, both motifs are identified correctly. As for the previous experiment, we also report on the optimal parameters, shown in Table 3.3.

μ	σ	λ_{opt}	Π_{opt}	iter
5	0.77	0.84	0.159	46
10	0.81	0.68		

Tab. 3.3: Optimal parameters for the synthetic data set S_3 with overlapping motifs. *Motifs have length 11 and 15 and start at position 5 and 10 respectively. The optimal function values as well as the number of function evaluations are the same, as our method optimizes holistically everything at once.*

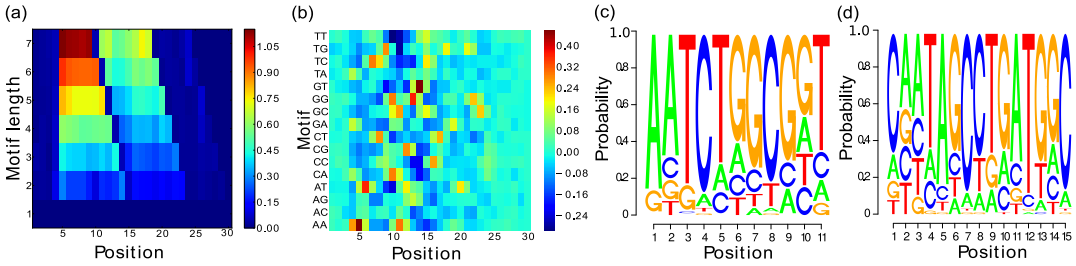


Fig. 3.9: Results for the synthetic experiment with overlapping motifs (AATCTGGCGGT, $\mu = 5$) and (CAATAGCCTGATGGC, $\mu = 10$). *The differential POIM is shown in Figure a), where we can extract the starting position of the two motifs as 5 and 10. Figure b) shows the POIM for the 2-mers, where the area between the starting and ending positions of both motifs is characterized by high scores. Figure c) and d) represent the correctly reconstructed motifs found by our proposed methodology.*

Results for a very long motif, i.e., data set S_4. At last, we investigate whether our approach is able to find a very long motif, as contained in the sample S_4. Due to the huge number of variables and the immense size of the POIM, we divide the POIM into 10 smaller conforming parts, in each searching for a motif of length 20. Fig. 3.10 shows the results. We can observe that the combination of the 10 computed PWMs reconstruct the real motif adequately.

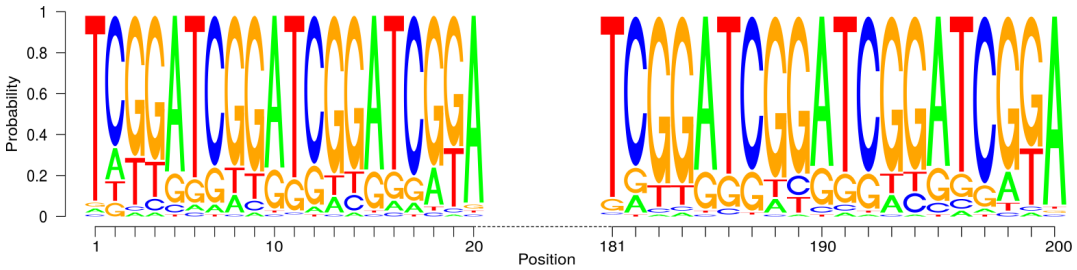


Fig. 3.10: Results of the synthetic experiment on the data set S_4. *The motif of length 200 is reconstructed correctly by overcoming the computationally infeasible POIM dimensionality of 4^{200} by splitting the long motif into smaller overlapping motifs. The resulting motif is shown here for the first 20 and the last 181 to 200 positions.*

The computation time improvement of SVM2Motif compared to the simple motif extraction of a POIM of the same order as the motif length is shown in Table 3.4. The experiment was repeated 5 times and the mean computation times and their standard deviations are shown. Due to the fact that POIMs are computable up to an order of 12 only, motifs with a length > 12 can not be extracted. For the SVM2Motif approach we observe a linear increase of computation time. Furthermore, SVM2Motif is able to find long motifs (motif length $\gg 12$) as it was shown in Figure 3.10. Note that the computation time of SVM2Motif depends on the choice of the parameters

motif length	POIM	SVM2Motif
2	0.61 ± 0.03	0.88 ±0.1
4	0.55 ± 0.01	1.8 ±0.18
6	0.79 ± 0.07	32.99 ± 0.29
8	5.18 ± 0.06	36.41 ±8.4
10	74.84 ± 1.26	52.95 ± 1.76
12	1195.58 ± 7.37	64.58 ±0.21
14	-	75.21±0.19
16	-	85.7 ±0.21
18	-	96.16 ± 0.07
20	-	106.9 ±0.32

Tab. 3.4: Computation time improvement. *Computation times (in seconds) of POIM and SVM2Motif for various motif lengths.*

used for the L-BFGS-B solver. We stopped the optimization when the relative error between the actual and last function value was smaller than $ftol = 10^{-3}$.
We can summarize that the experiments on synthetic data demonstrate the ability of our approach to robustly extract the true underlying—possibly overlapping—motifs from noisy data sets even for large motif sizes in a reasonable time.

Application to Human Splice Data

In this section, we evaluate our methodology on the *human* splice data set, which is introduced in Section 2.3.1. To verify our results, we use the JASPAR database (Sandelin et al., 2004) (available at `http://jaspar.genereg.net`), which provides us with a collection of important DNA motifs and also contains a splice site database. As a measure of the motif reconstruction quality (MRQ), we use the JASPAR SPLICE score (Sandelin et al., 2003), which is introduced in Section 2.4.
Note that real DNA sequences may contain non-polymorphic loci, which is why such a motif is not discriminative and we may thus not expect the SVM to identify this locus. We thus catch this special case and place this positional oligomer in the solution sequence. We apply the full experimental pipeline described in the previous section to the splice data, i.e., we first train an SVM, then generate the POIM and the differential POIM, from which we reconstruct a motif by our motifPOIM optimization approach. We compare our approach against the publicly available motif finder MEME (Multiple EM for Motif Elicitation, (Bailey, Elkan, et al., 1994)), a well-known motif discovering tool for DNA sequences, included in the MEME suite, which is a collection of tools for motif discovering and sequence analyzing. The user can specify the number of motifs as well as the length by either the exact length or a range specification. MEME expects the input sequences in FASTA file format. For comparison, we conducted three experiments with varying numbers of positive samples. For support vector machine training, we double the number of samples by filling in negative ones. We chose 400 positive samples (computation time ∼1min), which is the maximum amount of sequences when using the MEME online tool, 700 positive samples (∼10min), which is

the maximum recommended amount when using the MEME locally, and 2000 positive samples (∼12h). We compare the found motifs against the true splice site motif, taken from the JASPAR database by measure their accordance with the MRQ value.
Fig. 3.11 shows the preliminary results for 400 samples in terms of the differential POIM and corresponding POIM of order 2, shown for the entire sequence (see Figures 3.11 (a) and (c), respectively) as well as zoomed in for the "interesting" positions 36–76 of the sequence (see Figures 3.11 (b) and (d)). According to Fig. 3.11 (b), the largest entries correspond to a 3- and 2-mer that can be found at position 56 and 57, respectively. A significant increase of the score is recognizable for all k-mers at position 45, which is enhanced at position 46. The last largest entry for a 6-mer is found at position 58, which corresponds to the last largest entries of 4-mers at position 60 and 2-mers at position 62, from which we conclude that the discriminative motif starts at position 45 and ends at position 63. Thus, the motif we are searching for is expected to have a length of 19 nucleotides, which we use as an initialization for our motifPOIM approach. We also account for non-polymorphic loci and find that the nucleotides A and G appear in all DNA sequences of the data set, always at the positions 60 and 61, respectively. We thus place them in the final PWM with a probability of 100 percent.

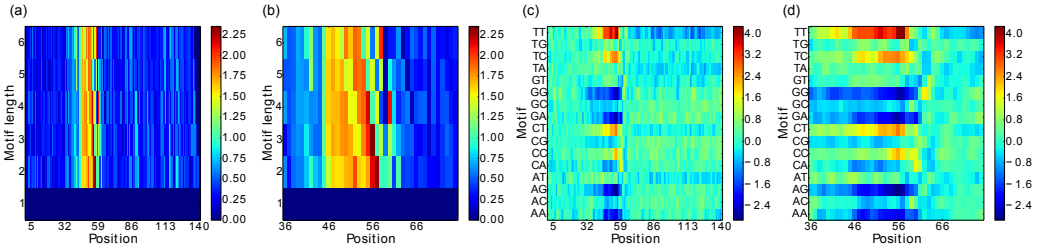


Fig. 3.11: Results of the real-world human splice experiment: *Figures (a) and (c) show the differential POIM and the POIM of order 2, respectively, for the entire sequence length of 200, while Figures (b) and (d) zoom into the "interesting" positions 36–76 only.*

The final results for 400 positive samples, are shown in Fig. 3.12, where the true underlying motif taken from the JASPAR splice database is shown in Fig. 3.12 (a), while the motif computed by our approach is shown in Fig. 3.12 (b) and the motif found by MEME is shown in Fig. 3.12 (c). The optimal parameters found by the L-BFGS-B solver are shown in Table 3.5. For all experiments, the start position is around the initialization value of 45, with a small variance of up to $\sigma = 0.44$. The great difference in the optimal function value is caused by the experiment dependent POIM scorings, for example in the POIM of order 2 of the first experiment we observe a maximal score value of 4 (see Fig. 3.11), where the maximal value in the third experiment was 5. From the resulting motif, shown in Fig 3.12 (b), we observe a striking accordance with the true motif as evidenced by a high MRQ value of 98.6. However, the motif found by MEME, shown in Fig. 3.12 (c), which has a length of 21 nucleotides, has a lower MRQ value of 94,5 although there exists a high similarity to the true motif. The reason is that the motif found by MEME starts 2 positions and

ends 1 position before the true motif. The results for 700 and 2000 positive training samples, are shown in Fig. 3.13 and Fig. 3.14, respectively. Here, the results for our approach show similar high MRQ values. MEME, found in both experiments a 21 nucleotides long motif starting 4 positions before the true motif. To get more insights, we fixed the motif length for both methods to 20 nucleotides, which corresponds to the underlying ground truth taken from the JASPAR database. The results are shown in Table 3.6. Again, we observe high MRQ values for the motif computed with our method. Interestingly, the MEME motif finder suffers a severe loss of performance for the first two experiments, achieving MRQ values around 90 for the last experiment, while the performance of our approach remains comparable. From Table 3.7, we can observe the computation times of SVM2Motif on the right column and of MEME on the left column for various numbers of positive samples. The computation times of SVM2Motif are significant lower than for MEME. Also the increase of the computation time with the number of the sequences is three orders of magnitude higher for MEME. The results show, that our approach is in principle able to infer motifs of high quality and is more robust and faster than MEME. Moreover, our approach easily handles sample-sizes beyond MEME.

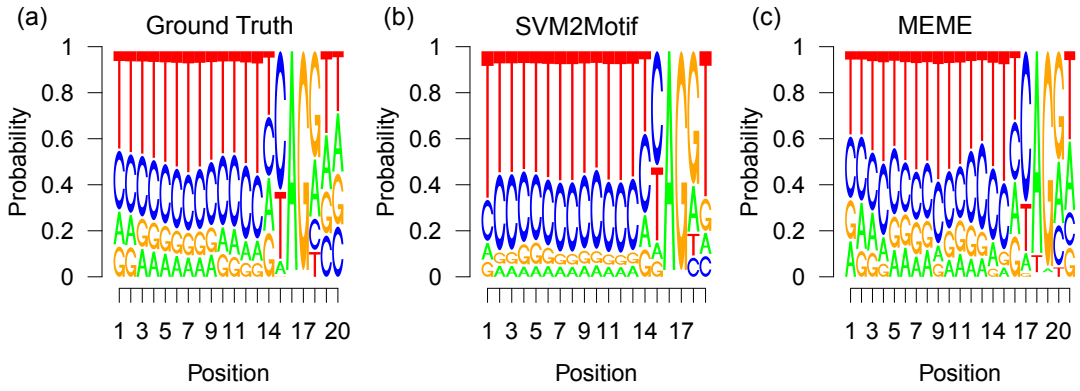


Fig. 3.12: Results for 400 human splice-site examples: *Figure (a) shows the (normalized) ground truth motif given by the JASPAR database (20 nucleotides). Figure (b) and (c) depict the corresponding PWMs, reconstructed by our approach SVM2Motif (19 nucleotides long, a MRQ value of* 98.92*) and by MEME (21 nucleotides, an MRQ value of* 94.77*) respectively.*

# pos samples	μ_{opt}	σ_{opt}	Π_{opt}	iter
400	45.0	0.24	175.34	24
700	44.5	0.44	176.31	98
2000	44.5	0.4	287.8	74

Tab. 3.5: Optimal parameters for the* human *splice data set.

# pos samples	MEME		SVM2Motif	
	length=21	length=20	length=19	length=20
400	94.77	90.2	98.92	98.6
700	90.06	88.78	98.51	98.31
2000	89.95	90.4	98.67	97.66

Tab. 3.6: MRQ values for the* human *splice data set.

# pos samples	MEME time (s)	SVM2Motif time (s)
400	206.82	137.06 ± 1.83
700	556.46	168.25± 13.01
2000	2709.26	342.99± 1.55

Tab. 3.7: Computation time comparison.

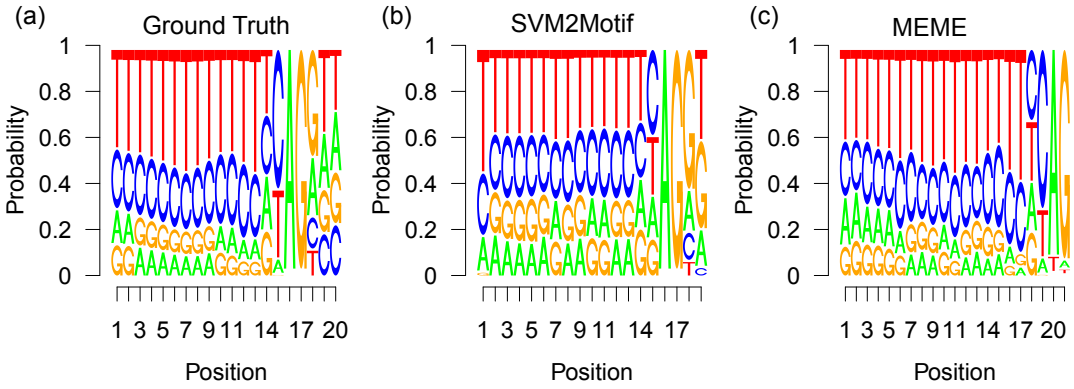


Fig. 3.13: Results for 700 human splice-site examples: *Figure (a) shows the (normalized) ground truth motif given by the JASPAR database (20 nucleotides). Figure (b) and (c) depict the corresponding (normalized) PWMs, reconstructed by our approach SVM2Motif (19 nucleotides long, a JASPAR score of* 98.51*) and by MEME (21 nucleotides, a JASPAR score of* 90.06*) respectively.*

Experimental Results for USPS Dataset

In the following, we apply motifPOIM on the USPS data, which we introduced in Section 2.3.2, and where the results are easy to interpret. Due to the fact that the USPS data set consists of grayscale images, but POIMs are only applicable to sequential data with categorical features, we have to perform two preprocessing steps: Firstly, the data are converted to a binary format by setting a threshold at -0.2 for the gray scale values. Values smaller or equal the given threshold were set to zero and to one otherwise. Secondly, we map each image to a sequence. To preserve locality in the vectorial image representation, we further preprocessed the data by scanning the image using a Hilbert curve of order 4, which is a method that is commonly applied to map images to sequences (Chung et al., 2007; Dafner et al., 2000). Fig. 3.15 (a) shows the path of the Hilbert-curve scan for the mean handwritten image of the digit three.

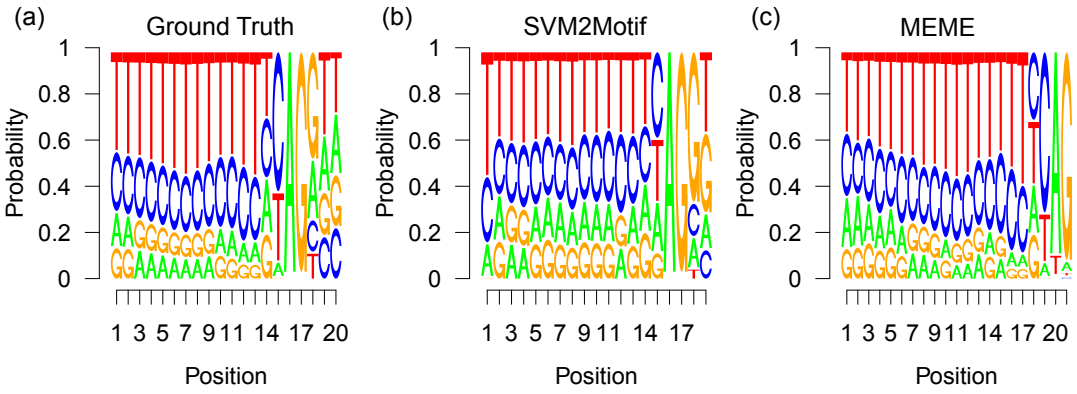
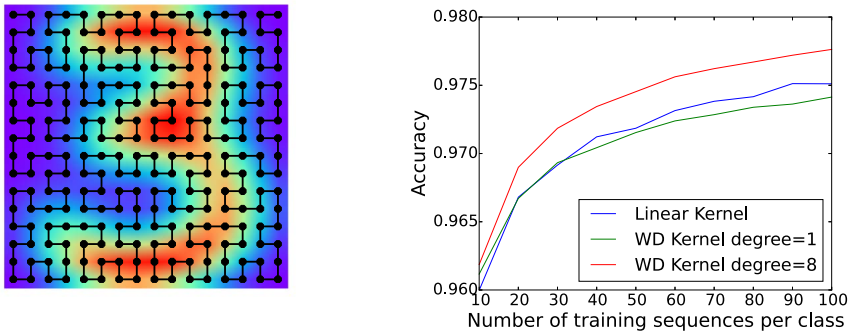


Fig. 3.14: Results for 2000 human splice-site examples: *Figure (a) shows the (normalized) ground truth motif given by the JASPAR database (20 nucleotides). Figure (b) and (c) depict the corresponding (normalized) PWMs, reconstructed by our approach SVM2Motif (19 nucleotides long, a JASPAR score of* 98.67*) and by MEME (21 nucleotides, a JASPAR score of* 89.95 *) respectively.*



(a) Hilbert Curve

(b) SVM performance

Fig. 3.15: (a) Foreground: illustration of Hilbert-curve scanning (of order 4) of an image depicting of the handwritten digit three. *The image is converted into a sequence through a curve that traverses the image in a way that mimics a fractal structure. It has been shown in Chung et al. (2007) and Dafner et al. (2000) that this strategy is able to well capture the image's locality structure. The heatmap in the background shows the average feature values for the images of the digit three.*
(b) SVM performance for various kernel functions. *The one-vs.-all SVM prediction accuracy is shown as a function of the number of training sequences per class for various WD sequence kernels over the Hilbert-scanned sequences and for a linear kernel on the gray-scale pixel values. The WD-kernel of degree 8 performers best, even for only a small number of training sequences.*

To justify the use of a high-dimensional WD-kernel, we compare it with a linear kernel

on the gray scale values as well as with the weighted degree kernel of degree one only regarding their respective SVM performances. The results in terms of multi-class classification accuracy are shown in Fig 3.15 (b), where the SVM was trained in a one-vs.-all scheme. We observe that a WD-kernel of degree 8 (dimensionality: $2^8 * 256 = 65536$) performs best in our experiments.

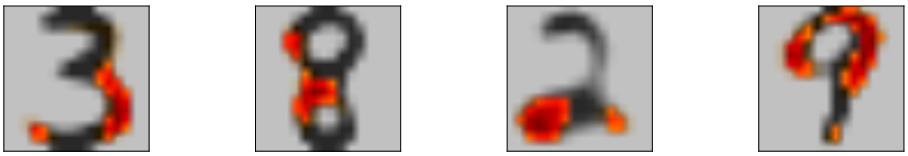


Fig. 3.16: Results on the USPS data set. *Illustration of the results found by our proposed framework, when training a WD-kernel SVM of degree 8 for the handwritten digits three vs. eight and two vs. nine, respectively. The highest scoring positions in the motif are highlighted in red. Note that these are very characteristic positions for the dissimilarities between both digits. The background consists of the average image of the respective digit, which is shown in black.*

For the remaining experiments, we focus on the binary classification task of the handwritten digits three vs. eight and two vs. nine, respectively. These respective digit pairs are considered to be especially difficult to discriminate. For both digit pairs, we train a WD-kernel SVM of degree 8 on the Hilbert-scanned sequences. Afterwards, we compute the POIM as described in Section 2.2.1 and use our presented methodology to find a motif that incorporates the discriminative positions of the SVM decision for both classes. In this experiment, we simply fix the length of the motif to 256, which thus coincides with the sequence length. The step of initializing the POIM parameter through analyzing the differential POIM is thus omitted in this experiment. The results, illustrated in Figure 3.16, show the precise coherence between the discriminative motifs found and the obvious individually characteristic differences of the two digits, respectively. For instance in the discriminative task three vs. eight, we can observe that the most distinctive positions in the motif of the digit eight (highlighted in red in Figure 3.16 (b)) are exactly the parts that are missing in the digit-three image.

3.1.5 Summary and Discussion

In this section, we introduced the proposed motifPOIM methodology, a model-based approach for extracting motifs from POIMs. In a nutshell, it is based on associating each candidate motif by a probability of occurrence at a certain location—which we call *probabilistic positional motif* (PPM)—and then (re-)construct from each PPM the POIM that is the most likely to be generated from the candidate PPM, which we call motifPOIM. The final motifs are obtained by optimizing over the candidate motifs such that the reconstruction error of the motifPOIM with respect to the truly given POIM is minimized. See Fig. 3.3 for an illustration.

We have developed a new methodology to extract long, overlapping and mutated motifs from trained support vector machines. Extending the work of Sonnenburg et al. (2008) on positional oligomer importance matrices (POIMs), the proposed novel probabilistic framework extracts the relevant motifs from the output of a WD-kernel SVM. To deal with the exponentially large size of the feature space associated with the SVM weight vector and the corresponding POIM ("... we realize that the list of POs can be prohibitively large for manual inspection." (Sonnenburg et al., 2008), page 8), we proposed a very efficient numerical framework.
We apply motiPOIM on biological data as well as on images that were converted into binary sequences. The results clearly illustrate the power of our approach in discovering discriminative motifs. In all synthetic data tasks, the hidden motifs could be found and almost perfectly reconstructed. For the human splice site experiments, we recovered known motifs up to a very high precision of 98.39% as compared to the JASPAR Splice data base. A thorough investigation of the association between the found motif and its biological function can be subject to further research. Furthermore, the results on handwritten digits clearly illustrate the power of our approach in discovering discriminative motifs even beyond bioinformatics.
For practical purposes and to enable the reader to replicate our results we published a Python framework[3]. We have implemented the core algorithms as an add-on to the Python interface of the Shogun Machine Learning Toolbox. It is not only an established machine-learning framework within the bioinformatics community, moreover, it already incorporates the possibility to extract positional oligomer importance matrices of trained support vector machines with a WD-kernel.
For an efficient optimization of the highly non-convex optimization problem (3.7), an appropriate initialization of the optimization variables is mandatory. Thus, we use the differential POIM (defined in Equation (2.18)) as indicator for extracting the area of interest: we search for points of accumulation of high scoring entries, from which we manually estimate the number of motifs as well as their lengths and starting positions.

Limitations and future work motifPOIM is a tool to extract driving motifs by mimicking a classifier. Even though the approach is able to find complex, overlapping, and long motifs, motifPOIMs face several restrictions. The main shortcoming of our presented methodology is the need of a proper initialization due the highly non-convex optimization problem. Changing the initial values of the optimization parameters may significantly change the optimization results. Fortunately, with the use of the differential POIM, we were able to manually estimate the number of motifs as well as their length and starting position, which then provides us with promising results. Therefore, future work will extend our approach to an autonomous extraction of the initialization variables, that is, the number of motifs, their length and starting positions.
Moreover, the presented method motifPOIM is limited to the interpretation of an SVM in combination with a WD kernel, which makes it applicable to sequential data with categorical features only. The current state of motifPOIM assumes that motifs are localized (they neither change shape nor position) and consist of a finite alphabet

[3] https://github.com/mcvidomi/poim2motif.git

('ACGT' in our examples). Also, examples must have same dimensionality (=same length) and going beyond those restrictions would require further research efforts. Therefore, a core issue might be the extension to other relevant kernels, such as, e.g., top kernel (Tsuda et al., 2002), which can handle sequences of variable lengths, spectrum kernels (Leslie et al., 2002), multiple kernels (Kloft et al., 2011; Kloft et al., 2009; Kloft and Blanchard, 2011; Nakajima et al., 2009; Cortes et al., 2013), other learning methods (Görnitz et al., 2009; Görnitz et al., 2009), or learning settings (Kloft and Laskov, 2010; Zeller et al., 2013; Görnitz et al., 2015).

3.2 Convex motifPOIM

In the previous section, we derived a method called motifPOIM to extract the discriminative features learned by an SVM with a WD kernel. Thereby, motifPOIM takes advantage of the POIM approach, and is able to reveal arbitrary long and even overlapping motifs from the given classifier solving a non-convex optimization problem. Although motifPOIM showed promising results on synthetically generated data as well as on real-world human splice data, their optimization generally leads to a sub-optimal local minimum and therefore may be less stable and reliable. In this section, we improve on motifPOIMs to achieve a simpler, faster, and — above all — *convex* approach. We demonstrate the advantage of convex motifPOIMs compared to its non-convex predecessor by evaluating both methods on the human splice site data set in the empirical evaluations. The presented method and results were published in Vidovic et al. (2017).

3.2.1 Motivation

Tackling the shortcomings of POIMs, we presented a new approach called motifPOIM in Section 3.1, which is an approach to extract the discriminative motifs directly from POIMs, regardless of their lengths. Even the challenging task of finding overlapping motifs is fulfilled by motifPOIM.

Unfortunately, also motifPOIMs come along with a handicap. The fact that the objective function (3.6) is highly non-convex makes the motifPOIM results highly unreliable. Nevertheless, we obtained strong results in the empirical analysis of Chapter 3.1 since we used a suitable initialization for the motifs, i.e., for the motifs PWMs, their starting positions and the variances of their starting positions. Therefore, we first extract the starting position with the help of the differential POIM (see Definition 2.18) and place it as a fix parameter in the optimization problem as described in Section 3.1.4. Furthermore, we used the differential POIM to extract the most likely motifs, which we used as initialization of the PWM entries. Using those convenient initializations, we could achieve reasonable results for all experiments (see Section 3.1.4). However, despite a suitable initialization, we observed a high variation across the resulting motifs, which indicates the existence of a great number of local optima.

Our idea of the convex improvement on motifPOIMs is based on both, the need of the explained initialization and on the following observation: from the results, shown in Table 3.3 and Table 3.5, we can observe that the variance of the starting position is always smaller than 1, which led us conclude that there is no need to include an additional variation of the starting position in the objective function.

Therefore, in the following, we investigate the optimization problem when at the same time considering the starting position as fixed.

3.2.2 Convex Formulation

In the following we consider probabilistic positional motifs (PPMs) (which we introduced in Section 3.1.2), where the starting position μ is given and fix. Thus, for a PPM $m_k := (r, \mu, \sigma)$, where μ is given, σ is chosen so small ($\sigma \leq \epsilon$), that the induced weight function

$$v_{(z,i)}(m_k) := \begin{cases} \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(i-\mu)^2}{2\sigma^2} \right) \prod_{\ell=1}^{k} r_{z_\ell,\ell} & \text{if } \sigma > \epsilon \\ \prod_{\ell=1}^{k} r_{z_\ell,\ell} & \text{else} \end{cases}, \tag{3.15}$$

does no longer depend on the probability function over the starting position of the PPM. Hence, the probability of a specific k-mer z is calculated solely as a product of its PWM entries. We can omit Eq. (3.1) since $\sigma^* << 1$ (which we figured out is given in most applications, see exemplary Table 3.3 and 3.5) provides a probability of $P^1 \sim 1$ for $i = \mu$ and 0 otherwise. In Section 3.1.3 we showed that it is sufficient to compute the motifPOIM for a small area around the starting position. The so-called confidence interval includes $99,7$ % of the starting positions of the dependent positional oligomers. In our case, where the variation of the starting position converges to zero, all sequence positions that contribute to the motifPOIM are within the following motif environment $\mathcal{U}(m_k) := \mathcal{U}(\mu) := [\mu, \ldots, \mu + k - 1]$. Thus, the SubPPMs are given as $\tilde{m}_{i-\mu}(m_k, \tilde{k})$ (Def. 8) for $i \in \mathcal{U}(\mu)$ and the convex motifPOIM formula is defined as follows:

Definition 11 (convex motifPOIM) *Given a motifPOIM as stated in Definition 7. Furthermore, given the weight function of Equation (3.15) with $\sigma \leq \epsilon$ and the motif environment $\mathcal{U}(m_k) := \mathcal{U}(\mu) := [\mu, \ldots, \mu + k - 1]$. Then we define a convex motifPOIM score as*

$$R_{(z,i)}(m_k) := \mathbb{1}_{\{i \in \mathcal{U}(m_k)\}} v_z(\tilde{m}_{i-\mu}(m_k, \tilde{k}))\,. \tag{3.16}$$

which results, applied successively, in the convex motifPOIM R.

Finally, this leads to the following convex objective function:

$$\Pi((m_{k,t})_{t=1,\ldots,T_k, k\in\mathcal{K}}) \;=\; \frac{1}{2} \sum_{k\in\mathcal{K}} \sum_{y\in\Sigma^{\tilde{k}}} \sum_{j=1}^{L-\tilde{k}+1} \left(\sum_{t=1}^{T_k} R_{y,j}(m_{k,t}) - Q_{\tilde{k},y,j} \right)^2 \tag{3.17}$$

Note, from now, in order to improve the readability, we restrict the extraction to one motif of fix length k, only. The theorems and proofs for the case of multiple motifs can be found in Appendix A.4. We also omit the motif relevance weight λ, which we introduced in the objective function 3.6, since we could not find any relation between the relevance of a motif and the value of λ.

Theorem 12 (Convexity) *Let D be a convex set, $m_k \in D$ a probabilistic motif, Q a POIM, $Q_{\tilde{k},y,j} \in \mathbb{R}$ for $y \in \Sigma^{\tilde{k}}$ and $j = 1, \ldots, L - \tilde{k} + 1$ and $\mu \in [1, L - k + 1]$, if the element wise minimum $Q_{\lfloor} \geq 1$, it holds that*

$$\Pi(m_k) \;=\; \frac{1}{2} \sum_{y\in\Sigma^{\tilde{k}}} \sum_{j=1}^{L-\tilde{k}+1} \left(R_{y,j}(m_k) - Q_{\tilde{k},y,j} \right)^2 \tag{3.18}$$

is convex.

Proof 2 *We have to proof the following inequality*

$$\begin{aligned} ||R(\Phi r + (1-\Phi)s;\mu) - Q||_2^2 \quad &\leq \quad \Phi||R(r;\mu) - Q||_2^2 \\ &\quad +(1-\Phi)||R(s;\mu) - Q||_2^2 \end{aligned}$$

to show convexity of $f(m_k)$, which is, for the case $j \notin \mathbb{1}_{\{i \in \mathcal{U}(\mu)\}}$, trivially fulfilled $Q_{\tilde{k},y,j} \in \mathbb{R}$. This, due to the fact, that a sum of convex functions is convex, leaves us with showing the following inequality

$$\big(\Phi a + (1-\Phi)b - Q_{\tilde{k},y,j}\big)^2 \leq \Phi\big(a - Q_{\tilde{k},y,j}\big)^2 + (1-\Phi)\big(b - Q_{\tilde{k},y,j})\big)^2\,, \tag{3.19}$$

where we replaced the PWM products $\prod_{l=j}^{k+j} r_{y_l,l}$ and $\prod_{l=j}^{k+j} s_{y_l,l}$ by a and b for an increased readability. After resolving and transforming Eq. (3.19) shortens to

$$\Phi^2 a^2 + 2\Phi ab - 2\Phi^2 ab \leq \Phi a^2 + 2\Phi Q_{\tilde{k},y,j}^2.$$

Since $-2\Phi^2 ab \leq 0$ and $\Phi^2 a^2 \leq \Phi a^2$, the equation reduces to

$$ab \leq Q_{\tilde{k},y,j}^2.$$

The fact that the maximum of ab is 1, concludes the proof for $Q_{\lfloor} \geq 1'$.

It is obvious that the assumption $Q_{\lfloor} \geq 1'$ made in Theorem 12 is not satisfied. Following Definition 3, the elements of a POIM can be also smaller than 1. The simplest way to meet the conditions, would be to add the absolute value of the smallest entry in the POIM increased by one, which is $q_{\lfloor} = |Q_{\lfloor}| + 1$, to all elements. Indeed, we still have to prove that adding a constant to a POIM does not affect the objective value. Therefore, we introduce the following theorem, which allows us to add a constant to a POIM without changing the optimum of the objective function 3.17.

Theorem 13 *Suppose that the objective function $\Pi(r;\mu)$ of*

$$\begin{aligned} \min_r \quad & \Pi(r;\mu) = \frac{1}{2}\sum_{y\in\Sigma^{\tilde{k}}}\sum_{j=1}^{L-\tilde{k}+1}\Big(R_{y,j}(m_{k,t}) - Q_{\tilde{k},y,j} + q\Big)^2 \\ s.t. \quad & 0 \leq r_{o,s} \leq 1 \qquad o = 1,\ldots,4, s = 1,\ldots,k\,, \\ & \sum_o r_{o,s} = 1 \qquad s = 1,\ldots,k\,. \end{aligned} \tag{3.20}$$

is convex and let r_q^* be the optimal solution, then $\forall q' \in \mathbb{R}\ r_{q'}^* = r_q^*$.

Proof 3 *Let r_q^* be the optimal solution of the objective function Π (3.20) with the inequality constraints $h_{o,s,1} = -r_{o,s}$ and $h_{o,s,2} = r_{o,s} - 1,\ o = 1,\ldots,4, s = 1,\ldots,k, i =$*

$1, 2$ *and the equality constraints* $g_s = \sum_o r_{o,s} - 1,\ s = 1, \ldots, k$, *and let* η *and* ξ *be the Lagrangian multipliers, then the Lagrangian function is as follows*

$$\mathcal{L}(r_q^*, \eta, \xi) = \Pi(r_q^*; \mu) + \sum_{o=1}^{4}\sum_{s=1}^{k} \eta_{o,s,1} h_{o,s,1} + \sum_{o=1}^{4}\sum_{s=1}^{k} \eta_{o,s,2} h_{o,s,2} + \sum_{s=1}^{k} \xi_s g_s \,.$$

The Karush-Kuhn-Tucker(KKT) conditions are satisfied for r_q^**: For the dual feasibility conditions* ($\eta \geq 0$) *and, since* r *is a stochastic matrix, the primal and the complementary slackness conditions* ($g_s = 0, s = 1, \ldots, k$, $h_{o,s,i} \leq 0, o = 1, \ldots, 4, s = 1, \ldots, k, i = 1, 2$, *and* $\eta_{o,s,i} h_{o,s,i} = 0, o = 1, \ldots, 4, s = 1, \ldots, k, i = 1, 2$) *are trivially fulfilled , which leaves us to show that the stationarity condition*

$$\nabla\Pi(r_q^*; \mu) + \sum_{i=1}^{2}\sum_{o}\sum_{s=1}^{k} \eta_{o,s,i} \nabla h_{o,s,i} + \sum_{s}^{k} \xi_s \nabla g_s = 0$$

is satisfied. Therefore, we insert the derivations and reorganize for the Lagrange multipliers ξ *, which leads to*

$$\xi_s = -\sum_{y}\sum_{j} \mathbb{1}_{\{i \in \mathcal{U}(\mu)\}} \left(\prod_{l=1}^{\tilde{k}} r^*_{y_l, j+l} \prod_{\substack{l=1 \\ l \neq t}}^{\tilde{k}} r^*_{y_l, j+l} - (Q_{\tilde{k}, y, j+\mu} - q) \prod_{\substack{l=1 \\ l \neq t}}^{\tilde{k}} r^*_{y_l, j+l} \right) + \eta_{o,s,1} + \eta_{o,s,2}.$$

With $\xi \in \mathbb{R}$ *it holds, that for any* $q' \in \mathbb{R}$ $r^*_{q'} = r^*_q$*. The fact that* Π *is convex,* h *is convex, and* g *is affine implies that the KKT conditions are sufficient for optimality and thus concludes the proof.*

3.2.3 Empirical Evaluation

In the empirical evaluation, we compare the convex motifPOIM approach to its non-convex predecessor on the human splice site set, which we introduced in Section 2.3.1. Therefore, we conduct a similar experiment as we did in Section 3.1.4. To verify our results we use the motif reconstruction quality (MRQ) (see Section 2.4), which measures the quality of our motif results with the true splice site motif obtained from the JASPAR database[4].

We apply the full experimental pipeline described in the Section 3.1.4 to the splice data, i.e., we first train an SVM, then generate the POIM and the differential POIM, from which we reconstruct a motif by our motifPOIM optimization approach. The training set consists of 2000 sequences including 500 positive samples. For SVM training, we use the SHOGUN machine-learning toolbox (Sonnenburg et al., 2010). The regularization constant C of the SVM and the degree d of the weighted-degree kernel are set to $C = 1$ and $d = 20$ for the biological experiments, which are proven default values (Sonnenburg et al., 2008). After SVM training, both the POIM Q and differential POIM are generated using the SHOGUN toolbox. We set the maximal POIM order to $k = 7$ because of memory requirements (storing all POIMs up to

[4] http://jaspar.genereg.net

an order of 10 requires about 4 gigabytes of space). Note that this is no restriction as our modified optimization problem (3.17) requires POIMs of degree two or three only. Nevertheless, POIMS of higher order than three can provide additional useful information since they contain prior information about the optimization variables.
We then search for points of accumulation of high scoring entries in the differential POIM, from which we estimate the number of motifs as well as their length and starting position. Due to the results in Table 3.2 we observed that the greedy initialization is more stable and reliable than using random initializations. Therefore, we use the greedy approach (introduced in Section 3.1.4) for estimating the initial values of PWMs given a POIM for the non-convex optimization problem. In the case of the non-convex approach, we initialize the variance of the starting position with $\sigma = 0.01$ and we keep the weight of the motif fix by $\lambda = 1$, due to the fact, that we only search for one motif.
The values of the PWM for the convex optimization problem are initialized randomly. For both, the convex and non-convex optimization problem, we employ the L-BFGS-B algorithm (Liu and Nocedal, 1989) from the *scipy* optimization toolbox (Jones et al., 2001–) and repeat the experiment 10 times. The mean results with their standard deviations for MRQ, function value, execution time are shown in Table 3.8, respectively. Note that the execution times in Table 3.8 include the optimization process only and not the additional time for SVM training and POIM computation as in Table 3.7 for example.

motifPOIM	σ_{opt}	MRQ (%)	time (s)	iter
convex	-	99.5 $\pm 7e^{-5}$	2.2 $\pm$ 0.19	36 $\pm$ 1.7
non-convex	0.57	98.6	196.1	174

Tab. 3.8: A comparison between convex and non-convex motifPOIM results *for the human splice site data.*

From the results shown in Table 3.8, we observe that the convex motifPOIM achieved a higher MRQ value than the non-convex motifPOIM (99.5 vs 98.6) within much shorter time (2 seconds vs 196 seconds) and with much less iterations (36 vs 174).
From Table 3.9, which is an extension of Table 3.4, we can observe that the use of the convex motifPOIM approach instead of the non-convex one yields a strong decrease in computation time of SVM2Motif.
A more intense evaluation of convex motifPOIMs on biological data is given in Chapter 4. The convex motifPOIM approach is applicable to image data as well if the images can be depicted as sequences of categorical features, as it was done for the USPS data set in Section 3.1.4.

3.2.4 Summary and Discussion

In this section, we improved upon motifPOIMs, which deliver unstable and unreliable results due to their non-convex optimization problem. Here, we derived a convex version of the motifPOIM approach by removing the uncertainty of the motif start positions in the optimization problem. Hence, we omitted the Gaussian distribution

motif length	POIM	SVM2Motif	SVM2Motif (convex)
2	0.61 ± 0.03	0.88 ±0.1	0.69 ±0.14
4	0.55 ± 0.01	1.8 ±0.18	0.59 ±0.05
6	0.79 ± 0.07	32.99 ± 0.29	0.57 ±0.01
8	5.18 ± 0.06	36.41 ±8.4	0.67 ±0.02
10	74.84 ± 1.26	52.95 ± 1.76	0.85 ±0.03
12	1195.58 ± 7.37	64.58 ±0.21	1.08 ±0.02
14	-	75.21±0.19	1.45 ±0.04
16	-	85.7 ±0.21	1.9 ±0.16
18	-	96.16 ± 0.07	2.47 ±0.07
20	-	106.9 ±0.32	3.48 ±0.55

Tab. 3.9: Computation time improvement. *Extension of Table 3.4 for the computation time (in seconds) of SVM2Motif using the convex motifPOIM approach for various motif lengths.*

in the weight function, which is justifiable regarding the following two facts: first, the starting positions of the motifs are necessary initialization parameters in the non-convex optimization problem and have to be chosen properly in advance from the diffPOIM to obtain reasonable results. Second, the variances of the starting positions were always smaller than 1, which let us conclude, that a well-chosen starting position does not need any further shifting. Hence, we omit the Gaussian distribution over the starting position in the weight function. Based on the above mentioned assumptions we could achieve a convex objective function. Computational experiments at the end of this chapter showed that the proposed convex motifPOIM achieved an increased MRQ value on the human splice data set compared to its non-convex predecessor. The computation of the convex optimization problem is two orders of magnitudes faster than solving the non-convex optimization problem using the same solver, which indicates a gain of efficiency when using the convex motifPOIM approach.

Chapter 4

MFI - Measure of Feature Importance

High prediction accuracies are not the only objective to consider when solving problems using machine learning. Instead, several applications in science and technology require an explanation of the learned decision function.
In Chapter 3 we developed a method which allows to extract the most relevant motifs learned by an SVM with a WD kernel from POIMs. Due to the fact that the approach is limited to sequential data with categorical features its applicability is restricted to fields like computational biology and computer security. A first generalization of POIM was achieved by FIRM, which is a rich theoretical concept but remains, for most applications computationally infeasible. In this chapter, we propose the measure of feature importance (MFI), a simple and easy extension of FIRM that can assess the discriminative features from arbitrary learning machines. The kernelized version kernel MFI can moreover reveal the importance of *non-linearly* coupled features. MFI can be applied for both *instance-based* and *model-based* explanation. Our formulation is based on the Hilbert-Schmidt independence criterion (HSIC), which originally had been proposed as a way to measure statistical independence for variables that exhibit non-linear couplings. Our empirical evaluation confirms the usefulness of the proposed approach on artificially generated data as well as on real-world data. Parts of the methods and results were published in Vidovic et al. (2016a) and Vidovic et al. (2017).

4.1 Motivation

In the previous chapters, we learned that POIMs can be used to visualize the important features, which are responsible for the WD-kernel SVM decision. Unfortunately, due to the fact that POIMs can be used for SVMs in combination with a weighted degree kernel only, their application is limited to sequential data with categorical features. Therefore, POIMs are mainly applied to biological problems, where the data stem from the quadbit alphabet {A,C,G,T}. In order to resolve those limitations of POIMs, Zien et al. (2009) come up with a method named feature importance ranking measure (FIRM). Maintaining the underlying concept of POIMs, FIRM is a general method, which is applicable to arbitrary learning machines and feature representations (see also Section 2.2). Although FIRM offers a new, theoretical rich method for interpreting machine learning algorithms, it is, for most applications, computationally intractable. Based on the theoretical foundation of FIRM, we devise a new method –MFI– for extracting the feature importance from arbitrary learning

machines (Vidovic et al., 2016a; Vidovic et al., 2017)[1]. It builds on the concepts of FIRM and POIMs, addressing their shortcomings. In contrast to FIRM, MFI can be easily computed for any learning machine, including deep neural networks, and any feature set (Nasir et al., 2014; Görnitz et al., 2014). We use a fast and simple sampling-based approach, which greatly simplifies implementation and evaluation. Furthermore, we propose kernel MFI, the kernelized version of MFI, which is even able to detect the importance of *non*-linearly coupled features. In Section 2.2 we introduced the difference between model-based and instance-based methods. POIMs and FIRM are model-based methods. However, often we are interested in the features that drove the classification decision for *a specific sample* instead of an explanation for the whole model. We call this instance-based explanation. Regarding the importance of the two explanation strategies, our method MFI enables us to assess both model-based (POIMs, FIRM) and instance-based (e.g., LRP) importances of discrete or continuous non-linearly coupled features.
After presenting our methods, namely MFI and kernel MFI, we empirically examine their properties and evaluate their effectiveness on several data sets: artificially generated DNA sequences (for which we can control the ground-truth), real-world human splice-site data, enhancer data, and gray scale USPS image data. Furthermore, we compare our methods to the LRP method (introduced in Section 2.2.5) on the MNIST data set (see Section 2.3.3).

4.2 Method

In this section, we describe our proposed method — Measure of Feature Importance (MFI). MFI extends the concepts of POIM and FIRM (which are contained as special cases) to non-linear feature interactions and instance-based feature importance attribution, and it is particularly simple to apply. To distinguish between model-based and instance-based MFI, we introduce a function called "explanation mode", which maps the sample in their respective feature space. Exemplary, for instance-based explanation, a DNA sequence would be mapped to itself, whereas the same sequence would be mapped to a POIM in case of model-based explanation.

Definition 14 (MFI and kernel MFI) *Let X be a random variable on a space $\mathcal{X}$. Furthermore, let $s : \mathcal{X} \to \mathbb{R}$ be a prediction function (output by an arbitrary learning machine), and let $f : \mathcal{X} \to \mathbb{R}$ be a real-valued feature. Let $\phi : \mathcal{X} \to F$ be a function ("explanation mode"), where F is an arbitrary space. Lastly, let $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and $l : F \times F \to \mathbb{R}$ be kernel functions. Then we define:*

$$MFI: \quad S_{\phi,f}(t) := \mathbb{E}[s(X)\phi(X)|f(X) = t] \tag{4.1}$$

$$\text{kernel } MFI: \quad S^{+}_{\phi,f}(t) = Cov[k(s(X), s(\cdot)), l(\phi(X), \phi(\cdot))|f(X) = t]. \tag{4.2}$$

An illustration of the instance-based and model-based explanation is given in Figure 4.1 exemplary for both sequence and image data.

[1] Note that the method MFI is named gPOIM in the publication (Vidovic et al., 2017), as the focus was on biological experiments only.

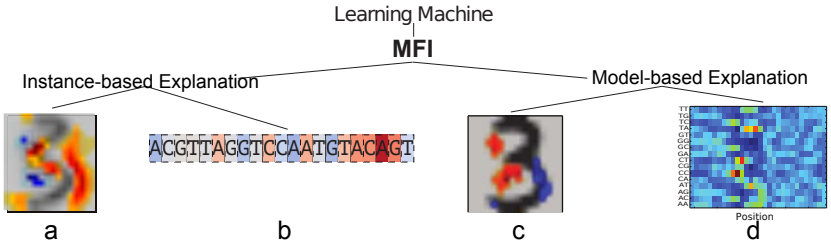


Fig. 4.1: MFI Examples. *We consider two possible flavors of feature importance: (left) instance-based importance measures (e.g.* Why is this specific example of '3' classified as '3' using my trained RBF-SVM classifier?*); (right) model-based importance measure (e.g.* Which regions are generally important for the classifier decision?*).*

Model-based MFI

Here, the task is to globally assess what features a given (trained) learning machine regards as most significant — independent of the given test samples. In the case of sequence data, where we have sequences of length L over the alphabet $\Sigma = \{A, C, G, T\}$, an importance map for all k-mers over all positions is gained by using the explanation mode $\phi : \Sigma^L \to \Sigma^{k \times L - k + 1}$, where each sequence is mapped to a sparse PWM, in which entries only indicate presence or absence of positional k-mers. In the case of two dimensional image data, $X \in \mathbb{R}^{d_1 \times d_2}$, where we already are in the decent visual explanation mode, $\phi(X) = X$ keeps the surroundings by mapping the data to itself. In both cases, we set $f(X) = t$, where $t = const$, which is why we can neglect it. The various case studies are shown in Figure 4.1 on the right.

Instance-based MFI

Given a specific sample, the task at hand is to assess why this sample has been assigned this specific classifier score or class prediction. In the case of sequence data we compute the feature importance of any positional k-mer in a given sequence $g \in \Sigma^L$ by $f(X) = X_{i:i+k}$, with $t = g_{i:i+k}$. In the case of images, where $g \in \mathbb{R}^{d_1 \times d_2}$ is the image of interest and $g_{i,j}$ expose one pixel, $f(X) = X_{i,j}$ maps the random samples $X \in \mathbb{R}^{d_1 \times d_2}$ to one pixel $t = g_{i,j}$. In both cases, we set $\phi(X) = 1$, which is why we can neglect it. For examples see Figure 4.1 on the left.

Relation to Hilbert-Schmidt Independence Criterion

In 2005, the Hilbert-Schmidt Independence Criterion (Gretton et al., 2005) (HSIC) was proposed as a kernel-based methodology to measure the independence of two distinct variables X and Y:

$$\begin{aligned} HSIC(X, Y) &= \|C_{XY}\|^2 = \mathbb{E}[k(X, X')l(Y, Y')] \\ &- 2\mathbb{E}[\mathbb{E}_X[k(X, X')]\mathbb{E}_Y[l(Y, Y')]] + \mathbb{E}[k(X, X')]\mathbb{E}[l(Y, Y')] \end{aligned}$$

where k and l are kernel functions and C_{XY} is the cross-covariance operator. We introduced HSIC more detailed in the fundamental Section 2.2.4. The following lemma

indicates the interesting relation between MFI and the empirical version of HSIC stated in Definition 6.

Lemma 1 (Relation of Kernel MFI to HSIC) *Given the kernel MFI of Definition 14 $S^+_{\phi,f}$, then $S^+_{\phi,f} = Cov[k(s(X),\cdot), l(\phi(X),\cdot)|f(X)=t]$ and the corresponding empirical Hilbert-Schmidt Independence Criterion stated in Definition 6 becomes: $HSIC_{emp}(S_\phi, \mathcal{Y}, \mathbb{R}) = \|S^+_\phi\|^2 = tr(KL)$.*

The relation to HSIC provides us with a practical tool to assess the importance of non-linear features as defined in kernel MFI in Definition 14.

Computation In order to make this approach practically suitable, we resort to sampling as an inference method. To this end, let $Z \subset \mathcal{X}$ be a subset of $\mathcal{X}$ containing $n = |Z|$ samples, Equation (4.1) can be approximated by

$$S_{\phi,f}(t) := \mathbb{E}_{X=Z}[s(X)\phi(X)|f(X)=t] = \frac{1}{|Z_{\{f(z)=t\}}|}\sum_{z\in Z} s(z)\phi(z)\mathbb{1}_{\{f(z)=t\}}\,, \tag{4.3}$$

where $Z_{\{f(z)=t\}} \subseteq Z$ contains only elements for which $f(z)=t$ holds. It holds true that if the number of samples $|Z| \to \infty$ then $S_{\phi,f} \to S^*_{\phi,f}$. A corresponding sampling scheme is also available for kernel MFI. To simplify notation and to resemble POIMs for subsequent analysis, we re-index MFI for model-based explanations on DNA sequence data as follows

$$\begin{aligned} S_{k,y,j} := |Z_{\{X[j]^k\neq y\}}| \cdot S_{\phi_{k,y,j}} &= |Z_{\{X[j]^k\neq y\}}| \cdot \mathbb{E}_{X=Z}[s(X)\phi_{k,y,i}(X)] \\ &= |Z_{\{X[j]^k\neq y\}}| \cdot \mathbb{E}_{X=Z}[s(X)\mathbb{1}_{\{X[j]^k=y\}}] \\ &= \mathbb{E}_{X=Z}[s(X)|X[j]^k=y]\,, \end{aligned} \tag{4.4}$$

which gives us the unnormalized POIM formulation of Definition 2.18.
A practical instruction on how to use MFI and kernel MFI is given in Algorithm 1. For the instance-based case, let x be the given instance, where the task is to measure the feature importances, respectively. Thereby, let T be the number of features that are to be considered. Furthermore, let $Z_{\setminus f_a} \cup x_{f_a},\ a = 1,\ldots,T$ be the set of random samples, of which the a-th feature of each sample is replaced by the a-th feature of x. Then, for each feature both MFI and kernel MFI measure the respective importance, which can be seen in Algorithm 1 (2) and (4). Firstly, the model-based kernel MFI (see Algorithm 1 (3)) computes the "output kernel matrix" K, which is the pairwise kernel matrix over the classifier outputs of Z. Secondly it successively computes first the "feature kernel matrix" L, which is the pairwise kernel matrix over the a-th feature of the sample in Z and assigns the a-th feature with the importance obtained by the trace of the product between K and L (cf. Lemma 1). The only difference to instance-based kernel MFI (see Algorithm 1 (4)) is that the "feature kernel matrix" is computed over the classifier outputs of $\tilde{Z} = Z_{\setminus f_a} \cup x_{f_a},\ a = 1,\ldots,T$, respectively.
In the following we show that we can control the error we made by sampling instead of computing the analytical expected value by an upper bound. Therefore, we use McDiarmid's inequality, which is based on the following assumption:

Algorithm 1: MFI and kernel MFI

input: the number of samples n, a trained learning machine $s(\cdot)$
generate: n samples $Z = \{z_1, \ldots, z_n\}$, where $z_i \sim Unif_{\mathcal{X}},\ \forall i = 1, \ldots, n$
(1) For model-based MFI:
return $\hat{S}_\phi = \frac{1}{|Z|} \sum_{z \in Z} s(z)\phi(z)$
(2) For instance-based MFI:
initialization: a=0
repeat
  a:=a+1
  for $a = 1 : T$
  $\tilde{Z} = Z_{\setminus f_a} \cup x_{f_a}$
  compute $\hat{S}_{f_a} = \frac{1}{|\tilde{Z}|} \sum_{z \in \tilde{Z}} s(z)$
until $a > n$
(3) For model-based kernel MFI:
compute pairwise "output kernel matrix" K, where $K_{i,j} = k(s(z_i), s(z_j))$
repeat
  a:=a+1
  for $a = 1 : T$
  compute pairwise "feature kernel matrix" L, where $L_{i,j} = l(\phi_a(z_i), \phi_a(z_j))$
  compute $\hat{S}^+_{\phi_a} = (n-1)^{-2}\mathbf{tr}(KL)$
until $a > n$
(4) For instance-based kernel MFI:
compute pairwise "output kernel matrix" K, where $K_{i,j} = k(s(z_i), s(z_j))$
repeat
  a:=a+1
  for $a = 1 : T$
  $\tilde{Z} = Z_{\setminus f_a} \cup x_{f_a}$
  compute pairwise "feature kernel matrix" L, where $L_{i,j} = l(s(\tilde{z}_i), s(\tilde{z}_j))$
  compute $\hat{S}^+_{f_a} = (n-1)^{-2}\mathbf{tr}(KL)$
until $a > n$

Assumption 1 (Bounded Difference Assumption) *Let $Z_1, \ldots, Z_n$ be independent random variables taking values in some set A. We say that a function $f : A^n \to \mathbb{R}$ satisfies the bounded difference assumption, if there exist real numbers $c_1, \ldots, c_n > 0$ so that for all $i = 1, \ldots, n$,*

$$\sup_{z_1, \ldots, z_n, z_i' \in Z} |f(z_1, \ldots, z_n) - f(z_1, \ldots, z_{i-1}, z_i', z_i + 1, \ldots, z_n)| \leq c_i. \tag{4.5}$$

Theorem 15 (McDiarmid's inequality) *Let $Z_1, \ldots, Z_n$ be independent random variables taking values in some set A. Under the bounded difference assumption, i.e., Assumption 1, for all $T > 0$,*

$$P(f(Z_1, \ldots, Z_n) - \mathbb{E}f(Z_1, \ldots, Z_n) \geq T) \leq e^{-2T^2 / \sum_{i=1}^n c_i^2}. \tag{4.6}$$

Theorem 16 (MFI bound) *Let $Z_1, \ldots, Z_n$ be independent random variables taking values in some set A, and fulfilling the conditional expectation of MFI (see Equation (4.1)), which is $f(Z_i) = t$ for all Z_i, $i = 1, \ldots, n$, than for any $\delta \in]0, 1]$, with probability at least $1 - \delta$,*

$$\sup_{t,\phi_i,i=1,\ldots,n_\Phi} (|\hat{S}_{\phi,f}(t, Z_1, \ldots, Z_n) - S_{\phi,f}(t, Z_1, \ldots, Z_n)| \leq B\sqrt{\frac{\ln(\frac{n_\Phi}{\delta})}{2n}}, \tag{4.7}$$

where n_Φ is the total number of features in Φ and $B = \max_{z \in Z} |s(z)\phi(z)|$.

Proof 4 *Put*

$$g_j(Z_1, \ldots, Z_n) := \sup_{t,\phi_j,j=1,\ldots,n_\Phi} |\hat{S}_{\phi_j,f}(t, Z_1, \ldots, Z_n) - S_{\phi_j,f}(t)|,$$

where Φ_i is the $j-th$ feature of Φ, then g satisfies the bounded difference assumption with $c_{ji} = \frac{B}{n}$, with $B = \max_{z \in Z} |s(z)\phi(z)|$. Therefore, by McDiarmid's inequality,

$$\mathbb{P}\left(\sup_{t,\phi_j,j=1,\ldots,n_\Phi} (|\hat{S}^+_{\phi_j,f}(t, Z_1, \ldots, Z_n) - S_{\phi_j,f}(t)| \geq T) \right) \leq e^{-2T^2 / \sum_{i=1}^n (\frac{B}{n})^2}.$$

Hence, we can follow that

$$\mathbb{P}(\sup_t (|\hat{S}^+_{\phi,f}(t, Z_1, \ldots, Z_n) - S_{\phi,f}(t)| \geq T) \leq n_\Phi e^{-2T^2 / \sum_{i=1}^n (\frac{B}{n})^2}$$

With

$$\delta := n_\Phi e^{-2T^2 / \sum_{i=1}^n (\frac{B}{n})^2} \tag{4.8}$$

we can rearrange Equation (4.8) for T to

$$T = \sqrt{\frac{\ln(\frac{n_\Phi}{\delta}) \sum_{i=1}^n (\frac{B}{n})^2}{2}} = B\sqrt{\frac{\ln(\frac{n_\Phi}{\delta})}{2n}},$$

which concludes the proof.

4.3 Empirical Evaluation

In this section, we investigate our method empirically. We start off with model-based explanations, where the goal is to assess the general idea of what the classifier has learned. Further on, we evaluate instance-based explanation strategies to explain feature relevance structures for specific samples.
For validation, we follow the Most Relevant First (MoRF) strategy and calculate its area under the curve as proposed in (Samek et al., 2017) and introduced in Section 2.4.2.

4.3.1 Computer Vision Experiments

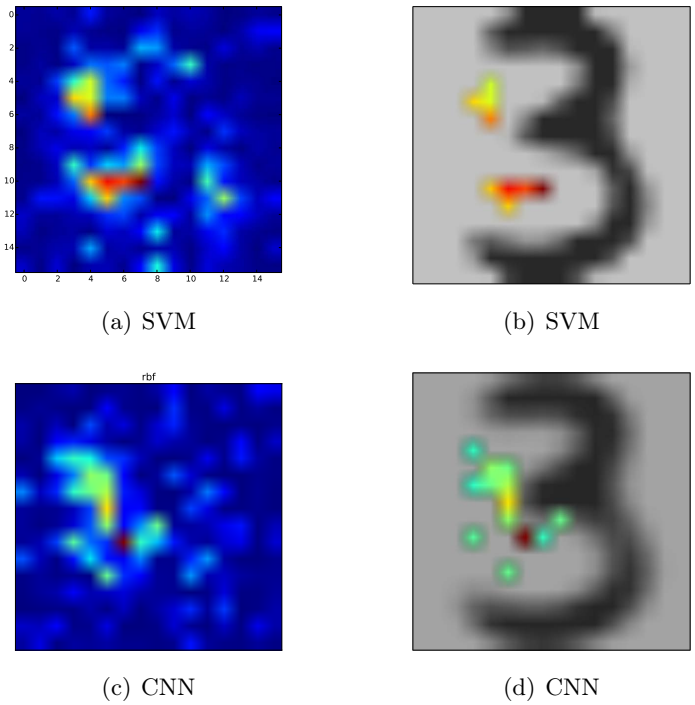
In the following experiments, we focus on the binary classification tasks of the handwritten digits three vs. eight taken from the USPS data set, which we introduced in Section 2.3.2.

Model-based Feature Importances

First we show the results for the model-based experiments, where we trained both, an SVM (see Section 2.1.3) with an RBF kernel (sigma = 3.16) and a convolutional neural network (see Section 2.1.5) with the following architecture. Two times in a row we perform a batch normalization layer, a convolutional layer with 28 filters of size 5x5 using ReLU activation functions, a batch normalization layer and a max-pool layer of size 2. Afterwards, we perform a dense layer with 256 ReLUs followed by a dense layer with 2 softmax units. After training, we applied the empirical kernel MFI with 1000 samples on both classifiers. The results are shown in Figure 4.2 as heatmap on the left and as mean digit, where the most important pixel were highlighted on the right. We can observe that for both, SVM (shown on the top row) and CNN (shown on the bottom row), the pixel-bridge that changes the digit three to the digit eight is of high importance. Although the region of interest is similar for SVM and CNN, the pattern of the important pixels differs substantially. Figure 4.3 shows the relation between classifier performance and the amount of pixel flippings in terms of most relevant first (see Section 2.4 for detailed information about the validation strategy). Compared to a random pixel flipping, we can clearly observe, for SVM and CNN, that the performance drops significantly faster when flipping the most important pixels found by our proposed kernel MFI. Note that the score on the y-axis has different ranges for SVM and CNN. The SVM decision function produces values in $\mathbb{R}$ where the sign of the value determines the predicted class. Contrary, the outputs of a CNN are probablities for each class label, which is why the score ranges in the interval $[0, 1]$. To find a suitable trade-off between runtime and accuracy, we evaluate both, runtime and convergence behavior (in terms of the Frobenius distance of two consecutive results) for increasing numbers of samples using a trained SVM. From the results, shown in Figure 4.4, we observe that the Frobenius distance (green curve) converges to zero already for small sample sizes (215 samples). Unfortunately, runtime grows very fast (almost exponentially) showing the boundaries of our method. Hence, a good trade-off between runtime and accuracy would be any sample size between 500 and 2000 in this experiment.

Instance-Based Feature Importances

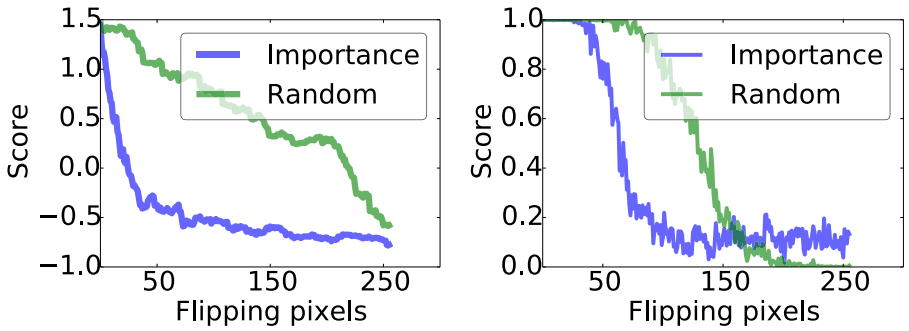
Previously, we showed the feature dependence structure underlying the classifier regarding to the whole model. In the following, we show on the basis of single samples, which feature structures have strong evidence and thus drive their classifier's decision. For the following instance-based explanation experiment, an SVM with an RBF kernel (sigma=3.16) was trained on the handwritten digits three and eight of the USPS training data set. MFI was computed for each pixel t in the image as follows:



(a) SVM (b) SVM

(c) CNN (d) CNN

Fig. 4.2: Results on the USPS data set using kernel MFI*. The results are shown as heatmap on the left and as mean digit, where the most important pixels are shown on the right. The results are shown for an SVM with an RBF kernel on the first row, and for the CNN on the second row, with 1000 random samples, respectively.*



(a) SVM (b) CNN

Fig. 4.3: Relevance pixel flipping. *Illustration of the classifier performance when flipping pixels by their computed relevance (see Figure 4.2) compared to flipping pixels randomly from a uniform distribution for a) SVM and b) CNN.*

randomly, we generated 10,000 samples in the same range and with the same size of the training gray scale images. Successively, for each pixel, we substitute the pixel information of the test image in all samples and computed their significance activity

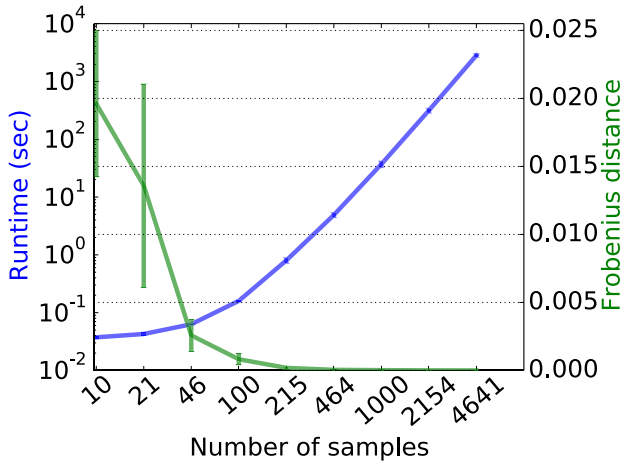


Fig. 4.4: Runtime behavior. *Runtime of MFI using a trained SVM measured in seconds for various sample sizes (plotted in blue) and the Frobenius distance of two consecutive results (green curve).*

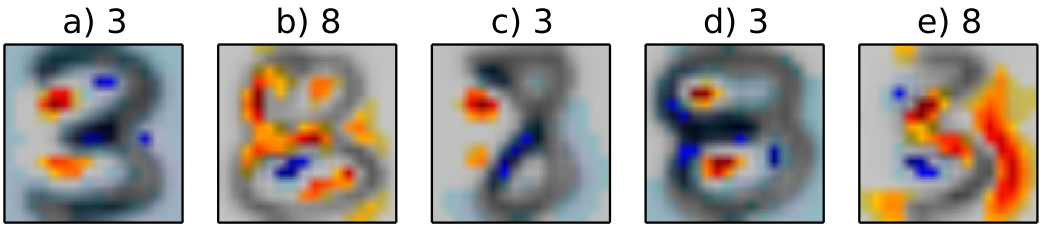


Fig. 4.5: Instance-based MFI explanation of the SVM decision for five USPS test data images. *The reconstruction sampling base comprises 10,000 samples. The highlighted pixels are informative for the individual SVM decisions (plotted at the image top) – only the first two images were correctly classified.*

score, respectively.

From Figure 4.5 we observe that the pixels building the vertical connection from a three to an eight on the left side have a strong discriminative evidence. If these positions are blank, the image is classified to a three, which, in case of the last three images leads to mis-classifications. For each example in Figure 4.5, the corresponding AOPC curves (introduced in Section 2.4) are shown in Figure 4.6.

Finally, we investigate the runtime behavior of instance-based MFI for various sample sizes. From Figure 4.7, we observe that the runtime, plotted in blue, increase linearly on a logarithmic scale with the number of samples. At the same time, the Frobenius distance between two consecutive results decreases.

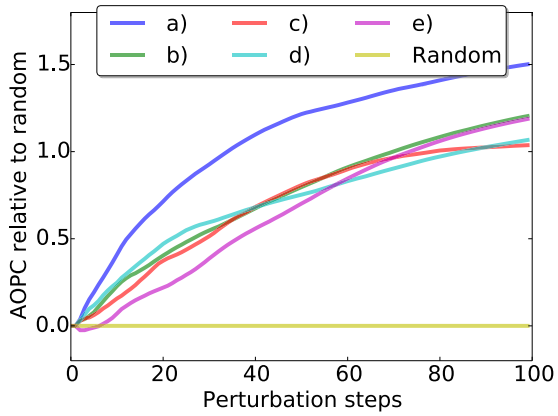


Fig. 4.6: Illustration of the AOPC curves *(see Equation (2.27)) relative to the random baseline for each Image of Figure 4.5.*

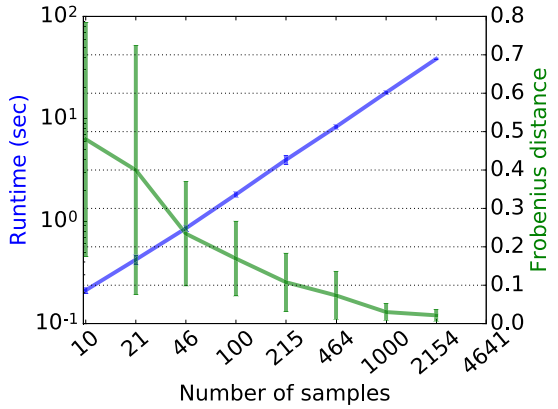


Fig. 4.7: Illustration of the runtime behavior for instance-based MFI. *The runtime, plotted in blue, is measured in seconds for various sample sizes and the Frobenius distance of two consecutive results is shown as green curve.*

Comparison to LRP

We compare MFI and kernel MFI with LRP, an approved method for interpreting deep neural networks (Bach et al., 2015; Binder et al., 2016), introduced in Section 2.2.5, on the MNIST data set (for more details on the MNIST data set see Section 2.3.3). Therefore, we trained a CNN (introduced in Section 2.1.5) on the MNIST data set with the following architecture: two convolutional layers with 8 filters of size 3x3 and ReLU, respectively, a max pooling layer of size 2x2, two convolutional layers with 16 filters of size 3x3 and ReLU, a max pooling layer of size 2x2, a dense layer with 40 ReLUs and a dense layer with 10 softmax units. The trained CNN achieved a 98.25% accuracy on the test data. For kernel MFI we used polynomial kernel functions of degree 2.

For comparison, we use the most relevance first method (see Section 2.4.2). Thereby,

the original image is degraded by flipping the most relevant pixels consecutively regarding the pixel importance assigned by the methods respectively. Simultaneously, in each flipping step, we measure the classifier output and plot it over the number of pixel flips. The steeper the decrease of the classifier output, the more descriptive is the heatmap of the respective method. A random pixel flipping serves as baseline. We picked 5 random test images of the MNIST dataset for comparison. The results are shown in Figure 4.8. We observe that kernel MFI, plotted as blue line performs similar to LRP, which is plotted in purple. Both show a fast decrease in confidence of the classifier. In contrast, MFI, shown as green curve, performs poorly and for two images (b,d) it is as bad as a random pixel flipping, which is plotted in cyan. We suspect the poor performance of MFI is due to the existence of nonlinear feature couplings learned by the CNN. Comparing the methods with respect to the computational efficiency, LRP is much faster than kernel MFI. However, LRP can be used only when both is fulfilled: first, the algorithm must be able to be formulated as a neural network and second, the weights of the network have to be known. On the contrary kernel MFI is applicable to any machine learning algorithms, keeping it as black-box system and just using the real valued output values for a given input. Furthermore, kernel MFI also allows a model-based interpretability of a classifier, whereas LRP is applicable to single samples only.

Biological Experiments

The empirical evaluation on biological data has three parts: First, we investigate and discuss the properties of our proposed method MFI for both instance-based and model based explanation. For the model-based explanation, we also investigate MFI in combination with the convex motif extraction method, i.e. convex motifPOIM (see Section 3.2) when compared to their predecessors, i.e., POIMs with non-convex motifPOIM, on artificially generated data. In the second part, we apply MFI with convex motifPOIMs to find driving motifs in real-world human splice-site data where ground truth motifs are known. Here, we compare motif reconstruction accuracies against state-of-the-art competitors under various experimental settings. Finally, we perform an analysis of the publicly available enhancer data set and try to find and verify the driving motifs in a real-world setting where no ground truth motifs are given.
Since we focus on computational biology settings and specifically on the important task of motif finding in DNA sequences, we measure the accuracy of predicted motifs regarding the ground truth sequence motif in terms of motif reconstruction quality, which is introduced in Section 2.4) (Sandelin et al., 2003). As in Vidovic et al. (2015a) and Vidovic et al. (2015b), we use differential POIMs (cf. Equation (2.18)) to estimate starting position and length of the motifs (details to this preprocessing step can be found in Section 3.1.4).

Controlled Experiments

In this section, we assess and discuss the properties of both, MFI and convex motifPOIMs, where the latter was presented in Section 3.2. We start by showing the

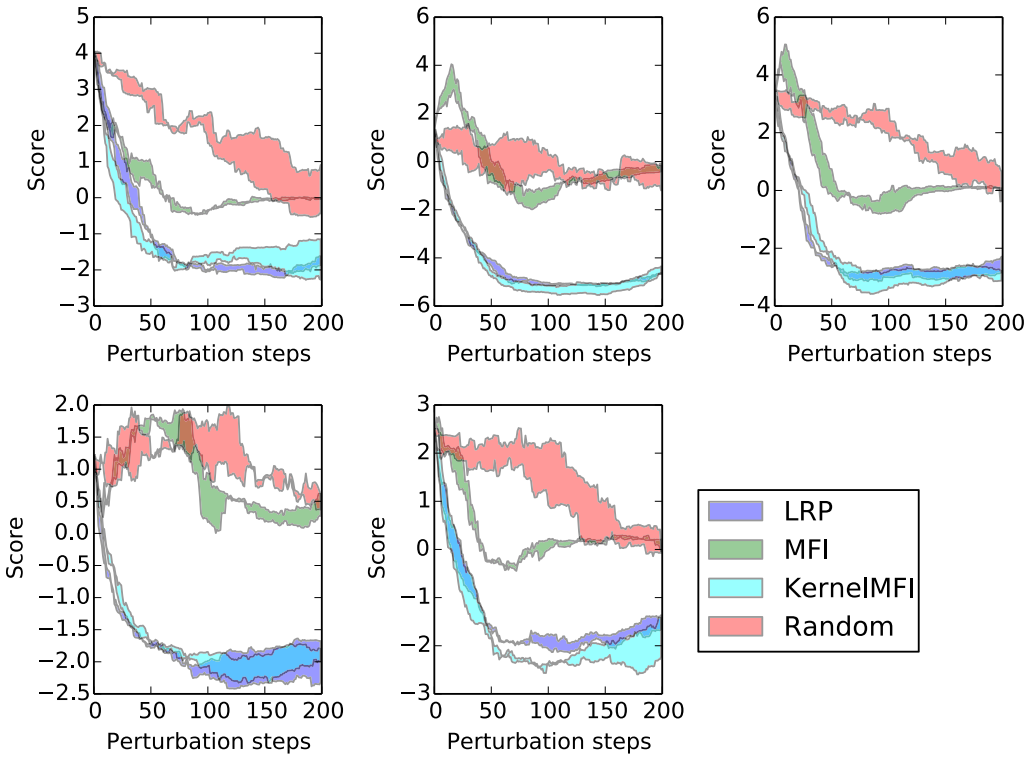


Fig. 4.8: Comparing the performance of MFI, kernel MFI and LRP on the MNIST data set for a trained CNN. *We degrade the image by flipping the most relevant pixels regarding the methods importance mapping, respectively. Simultaneously the impact on the classifier output is measured and plotted over the number of pixel flips. A random pixel flipping serves as baseline.*

benefits of instance-based explanations, a new mode of explanation, which was made possible by MFI. Further, we continue to discuss MFI in the traditional model-based explanation mode and compare solutions against its predecessor (POIM) in a variety of experiments. Finally, we show that convex motifPOIMs are able to extract complex motifs and unleash the full potential of our method by application to convolutional neural networks.

Instance-based Explanation of DNA Sequences For the instance-based experiment, we used 10,000 randomly generated sequences, with two motifs, ("GGCCGTAAA",pos=11) and ("TTTCACGTTGA", pos=24) placed in one quarter for training an SVM with a WD-kernel. The SVM achieves an accuracy of 98,63%. In the following we explain the classifier decision for individual test sequences by subsequently explaining one example from the sets of the true positive, false positive, false negative and true negative test samples. The set of random samples used for the MFI computation (Equation (4.3)) comprises 10,000 samples. From the results, shown in Figure 4.9, we observe that the nucleotides building the two patterns have a strong

discriminative evidence. If the discriminative patterns are too noisy, the sequences are classified to the negative class, which, in case of the false negative (FN) example leads to mis-classification. Elsewise, if only one of the two patterns were inserted, the classifier gives high evidence to the single pattern, which also leads to mis-classification.

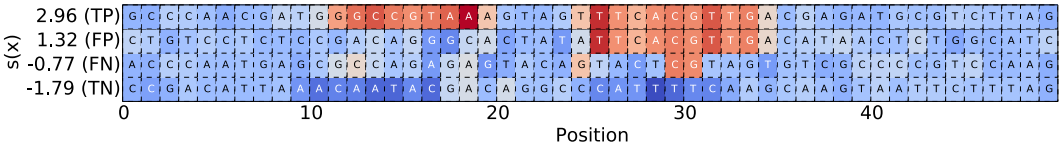


Fig. 4.9: Instance-based explanation. *Instance-based explanation of the SVM decision for samples coming from the true positive (TP), false positive (FP), false negative (FN) and true negative (TN) test set, respectively. The highlighted nucleotides are informative for the individual SVM decisions, which are plotted as scores on the y-axis.*

Model-based Explanation of DNA Sequences We generated randomly 10,000 sequences of length 30, where positive examples carry the motif CCTATA at position 11. As classifiers, we employ support vector machines with weighted degree kernel (degree=8) and convolutional neural networks with following architecture: a 2D convolution layer with 10 tanh-filters of size 8x4, a max-pool layer of size=2, a dense layer with 100 ReLUs and a dense layer with 2 softmax units.
To show that MFI converges fast towards POIMs, we measured the Frobenius distance between MFI and POIMs for an increasing number of samples used to build MFI. In average, 1000 samples are enough to cross a 10^{-3} error bound. The experiment was repeated 25 times and means as well as standard deviations are reported in Figure 4.10.
Subsequently, as shown in Figure 4.11, we investigate the stability and accuracy of MFI (using 1000 samples, green line) under noise when compared against the computed POIM (blue line) as implemented in the Shogun machine learning toolbox (Sonnenburg et al., 2010) (only available for linear SVMs with weighted-degree kernel though). Noise was induced by mutating each of the nucleotides of the underlying motif with some probability (x-axis). As can be seen, there is virtually no difference between both methods for the same classifier using convex motifPOIM. Hence, we established that MFI are a valid replacement for POIMs. To fully take advantage of the MFI approach, we are able to use more complex classifiers, e.g. CNNs (red line) which shows superior behavior. The drop after a noise level of 60% can be explained as follows. At a noise level of 66.6% all motifs have equal probability, which is why above that level, other motifs become more likely than the inserted motif. Hence, due to the considerable rarity of the motif at 66.6% the classifier's ability drops significantly.

Motif Extraction by Mimicking MFI To show whether or not we are able to find long motifs with our proposed method, we draw 10.000 uniformly distributed

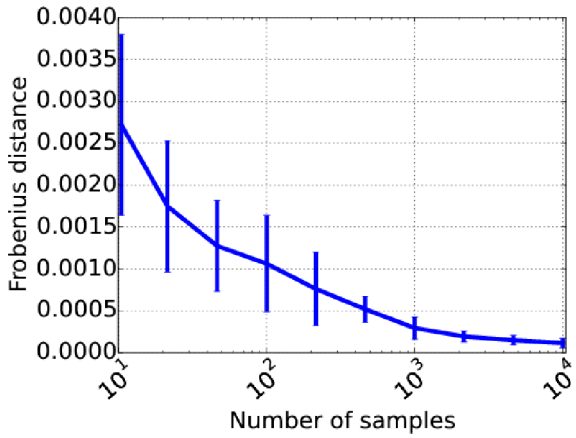


Fig. 4.10: Reconstruction accuracy of MFI. *Visualization of the reconstruction accuracy of MFI when compared to POIM for an increasing number of samples, measured by Frobenius distance.*

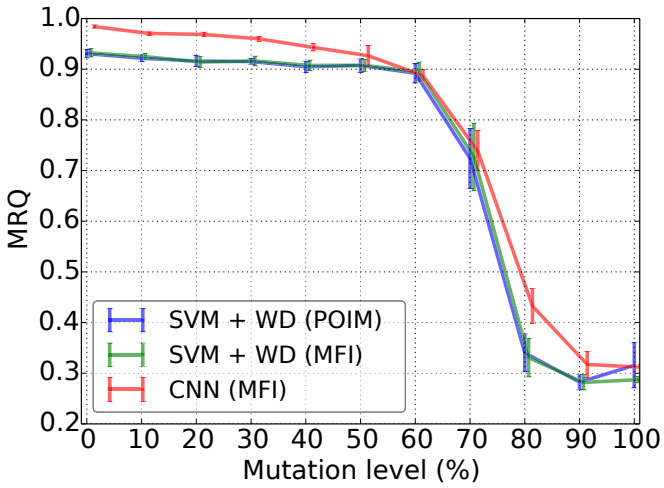


Fig. 4.11: Accuracy comparison. *The MRQ of SVM+POIM+convex motifPOIM, SVM+MFI +convex motifPOIM and CNN+MFI +convex motifPOIM for various levels of mutation in the data set.*

toy DNA sequences of length 100, where we insert a motif of length 50 at position 10 in 25% of the data. The motif pattern was of the form TGGCCGTAAA, which was inserted five times in a row. From the results, shown in Figure 4.12, we can observe that the real motif was found correctly.
In the following, we show that our method is capable of handling the difficulty of finding motifs that overlap each other, which means, motifs are sharing at least one position. For the experiment, we generate 1000 random sequences, where we placed the motifs ("TGGCCGGAAA",11) and ("TTCCCGTTGACAT",16) in 125 sequences, respectively. The results are shown in Figure 4.13, where we observe that the highest

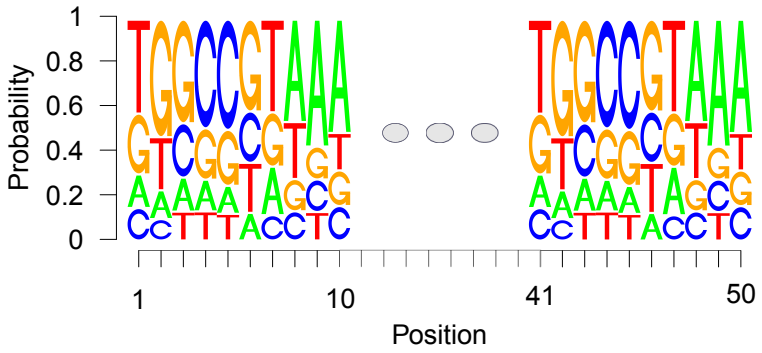


Fig. 4.12: Long motif extraction. *An induced ground truth motif of length 50 with the recurrent pattern TGGCCGTAAA is reconstructed concisely from noisy data.*

probability is given to the truly underlying motifs. The starting positions of the motifs were extracted from the differential POIM, which is shown in the center of Figure 4.13.

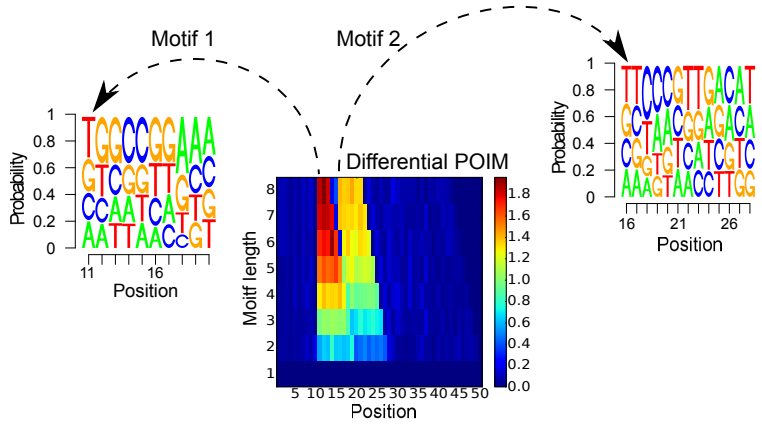


Fig. 4.13: Overlapping motifs extraction. *Based on the differential POIM (center), the estimated starting positions of the motifs are 11 and 16. Arrows point to the extracted motifs with highest scoring sequences coinciding with the induced ground truth motifs. Motifs are overlapping from positions 16 to 21.*

Furthermore, we investigate the runtime behavior of the presented method. We aim to show two key results. First, the algorithm should produce an adequate MFI, which can be measured by the Frobenius distance to the true POIM, in a reasonable time. We can observe from the left side of Figure 4.14 that the runtime increases when at the same time the Frobenius norm between MFI and the true POIM decreases. After already 25 sec. we observe an accuracy of 10^{-4}. Second, the optimization procedure should be computable in a reasonable time, also for complex motif finding problems. Therefore, we measured the runtime for increasing complexity, i.e. increasing number of motifs and motifs lengths. The results are shown on the right side of Figure 4.14. The runtime increases almost linearly with the complexity of the program. Both

experiments together show that our method is computable in reasonable times.

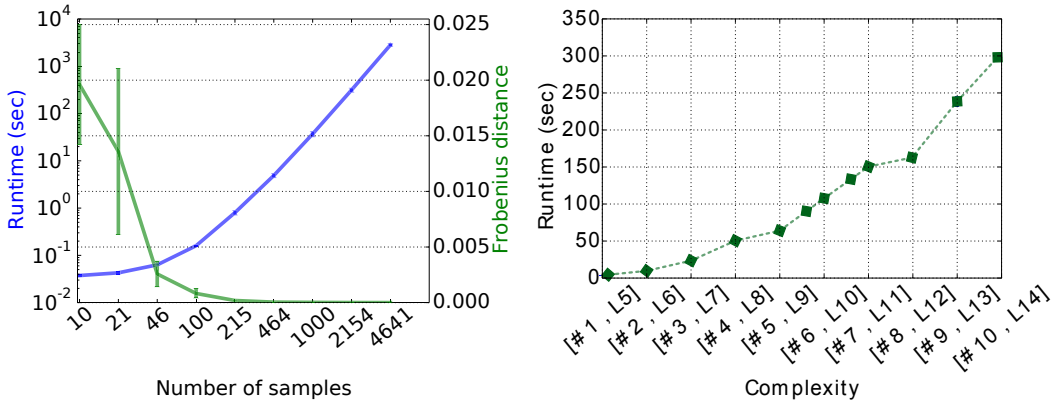


Fig. 4.14: Runtimes. *Left: Runtime (in seconds) for increasing number of samples (blue) and corresponding Frobenius distance of two consecutive results (saturation curve, green). Right: Runtime (in seconds) for increasing complexity (number # and length L of motifs).*

Motif Extraction from Human Splice-site Data

We evaluate our proposed methods (MFI and convex motifPOIM) on the human splice site data set, which is introduced in Section 2.3.1. We used POIM and motifPOIM as baseline methods and MEME (Bailey et al., 2015) as the state-of-the-art competitor. We use a random initialization of the values in the PWM for both, convex motifPOIM and non-convex motifPOIM. To verify our results we employ the splice site motifs given by the JASPAR database[2] (Mathelier et al., 2015) as ground truth. The results in Table 4.1 show the mean and standard deviation of the MRQ accuracies for various numbers of training examples and 10 repetitions of each experiment. For all experiments, besides for the MEME motif finder, we employ a weighted degree kernel for the SVM with degree=8 setting hyper-parameters according to Vidovic et al. (2015a) and Vidovic et al. (2015b). Using POIMs as implemented in the Shogun machine learning toolbox (Sonnenburg et al., 2010), we test the (non-convex) motifPOIM method against our convex motifPOIM. The resulting lower standard deviations indicate that our convex motifPOIM approach is more reliable than its non-convex predecessor. Furthermore, we gain almost 2% MRQ due to its inherently stable behavior. Next, we compare the results when using MFI (with 1000 random samples) instead of the Shogun implemented POIM. Here, we observe that the results are indistinguishable and thus, empirically justifying our sampling approach on non-trivial real-world data. Having established MFI as a valid approach for replacing POIMs, we proceed by taking advantage of its full potential and apply convolutional neural networks with following architecture: a 2D convolution layer with 10 tanh-filters of size 8x4, a max-pool layer with size=2, a dense layer with 100 ReLUs and a dense

[2] http://jaspar.genereg.net

layer with 2 softmax units. The architecture is similar to the one used in Alipanahi et al. (2015) and gives similar, almost perfect results, which are shown in Figure 4.15. As can be seen in Table 4.1, (g)POIM-based approaches outperform the MEME motif finder, which did not converge in reasonable time (>20h) for 30,000 sequences. Also, for less than 6,000 samples, MEME seems rather unstable as indicated by the high standard deviations.

Tab. 4.1: Results (human splice site experiment) *MRQ values and standard deviations for the human splice data set comparing MFI with convex motifPOIM against POIM with both motifPOIM variations as well as for the state-of-the-art competitor MEME. The SVM was trained using weighted degree kernels. Due to lack of space, MP is the abbreviation for motifPOIM and cMP for convex motifPOIM.*

		POIM		MFI	
#	MEME	SVM+MP	SVM+cMP	SVM+cMP	CNN+cMP
300	89.31±5.27	97.79±0.37	98.77±0.17	98.97±0.24	98.94±0.26
600	90.02±2.86	97.91±0.24	99.16±0.14	99.18±0.14	99.17±0.14
1,200	92.66 ±4.99	97.49±0.13	99.36±0.10	99.25±0.03	99.32±0.13
2,400	93.18 ±4.18	97.61±0.24	99.37±0.07	99.38±0.06	99.37±0.05
6,000	94.70 ± 0.17	97.91±0.31	99.42±0.14	99.45±0.06	99.44±0.06
30,000	-	97.05±0.09	99.39±0.08	99.54±0.02	99.56±0.02

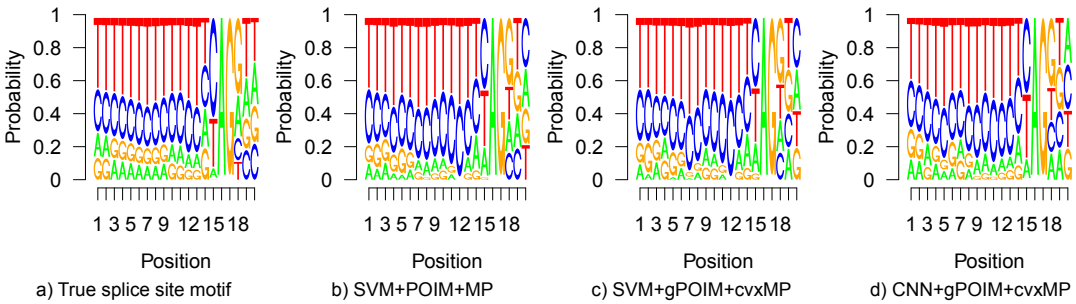


Fig. 4.15: Inferred motifs (human splice sites experiment): *a) true motif given by the JASPAR database , b) predicted motif from SVM+POIM+motifPOIM (MRQ=97.05), c) SVM+MFI +convex motifPOIM (MRQ=99.54), and d) CNN+MFI +convex motifPOIM (MRQ=99.56).*

The use of MFI enables us to not only extract motifs based on the trained model, instead, we are able to explain classifier decisions for specific sequences. Figure 4.16 shows the position-wise importances for 4 different sequences (true positive, false positive, false negative, and true negative) for the full 141 nucleotide sequence and a zoomed-in version. As can be seen, most important (dark blue and red) regions are around the true underlying sequence motif site with red for higher scores $s(x)$ and blue for lower/negative classifier scores $s(x)$.

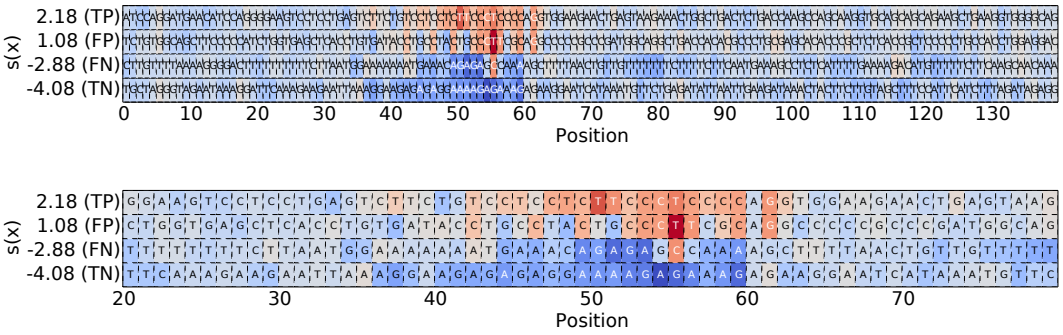


Fig. 4.16: Instance-based explanation (human splice-site experiment). *Position-wise importances for four specific sequences from the human splice site data set: (a) a true positive with high positive score $s(x)$, (b) a false positive with low positive score $s(x)$, (c) a false negative with low negative score $s(x)$, and (d) a true negative with high negative score $s(x)$. Upper figure shows whole sequences, lower figure is a zoomed-in version for better readability.*

Exploratory Analysis of Enhancers and their Strength

For most applications, there will be no ground truth motifs available in advance. To give an example on how to apply and verify ML2Motif in this real-world situation, we chose to test our method on an enhancer data set[3] supplied by Liu et al. (2016). The data set comprises 742 weak enhancers, 742 strong enhancers and 1484 non-enhancers, each with a sequence length of 200 respectively.

Following Liu et al. (2016), we build a two-layer classification framework, where the first layer decides whether or not the given sample is an enhancer. In case of positive prediction, the second layer will predict the enhancers strength. For both layers, we trained an SVM ($C = 1$) with a WD-kernel (kernel degree $k = 8$), where the first layer was trained on non-, strong, and weak enhancers and the second layer on strong (+1 class) and weak (-1 class) enhancers only. A 5-fold cross validation was applied to test prediction accuracy. Here, we report a 95% accuracy for the first layer and 90% for the second layer. Both methods exceed the given baseline method (iEnhancer-2L, 76.89% and 61.93%, respectively) by a comfortable margin, which we claim on the richer feature representation (i.e. weighted degree kernel vs. RBF kernel).

If we apply ML2Motif to the SVM solution, we can have a first glimpse at the problem by using the instance-based explanation mode for a set of randomly chosen sequences of differing classes (cf. Fig 4.17 and Figure 4.18). We observe that importances spread over the whole sequence length. This could be a hint that either multiple motifs spread over the whole sequence or that motifs are not located (=they can change position). Moreover, the importances include almost exclusively Guanine-sequences for the enhancer class. Hence, extracted motifs should contain strong Guanine components.

Using again diffPOIMs to estimate locations and length of motifs, we extract the three most prevalent motifs (positions 138, 0, 82 and length 57, 30, 8) as shown in

[3] http://bioinformatics.hitsz.edu.cn/iEnhancer-2L/data

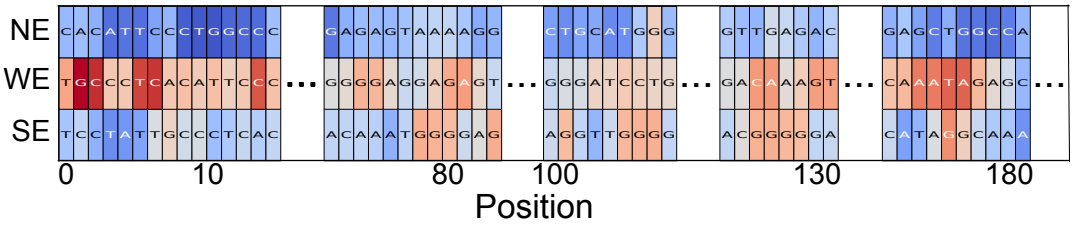


Fig. 4.17: Instance-based explanation (enhancer experiment, layer 1). *Instance-based explanation of one sample of each type, strong enhancers (SE), weak enhancers (WE), and non-enhancers (NE). Due to the length of the sequences, only relevant parts of the instanced-based explanation are shown. We can observe that there are multiple relevant motifs, which also depend on the enhancer type (WE or SE).*

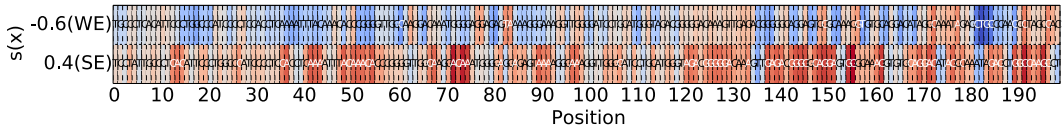


Fig. 4.18: Instance-based explanation (enhancer experiment, layer 2). *Position-wise importances are shown for a strong enhancers (SE) and a weak enhancers (WE) sequence.*

Figure 4.19. As already suspected from the instance-based explanations, the motifs contain strong Guanine components. Surprisingly, Guanine seems to dominate all three motifs with no or only little influence of other nucleotide bases. To test whether or not solutions are degenerate, we rank the test sequences according to the inferred $n \in \{1, 2, 3\}$ highest scoring motifs (green bars in Figure 4.19). Interestingly, two motifs are enough to surpass the accuracy of the baseline method (iEnhancer-2L, red dashed line). We therefore conclude that poly(G) sequences are the key for understanding enhancers.

4.4 Summary and Discussion

Our proposed method MFI enables the machine learning researcher to open the black-box of any machine learning model. Like POIMs and FIRM, it takes feature correlations into account but uses a simple sampling based strategy to assess the importance of any feature of interest. We showed, that we are able to control the error made by using the sampling approach by an upper bound. Unlike POIMs, MFI is less restricted by specific learning settings. It can be applied to continuous features as well as categorical ones, sequences as well as other structures, such as images. Generally, it is not restricted by a specific type of application and/or learning machine. Hence, it could be easily applied to other types of applications, such as explanation of most expressive electrode-combinations in hand movement recognition with EMG signals (Vidovic et al., 2016b), change point/anomaly detections in time series for

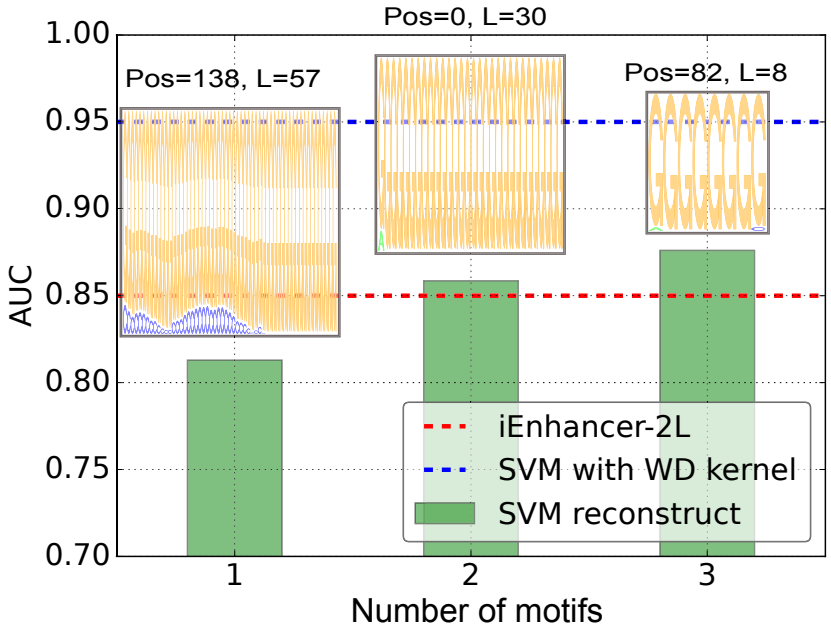


Fig. 4.19: Classification accuracy (enhancer experiment, layer 1). *An SVM with a WD-kernel (blue dashed line) was trained to discriminate between enhancers and non-enhancers and archives superior AUC of 95% when compared to the baseline method iEnhancer-2L (red dashed line, AUC of 85%). Subsequently, ML2Motif was applied to extract $n \in \{1, 2, 3\}$ most significant motifs (x-axis) from the SVM classifier. To test their respective relevance, test sequences were ranked according to the extracted motifs. Results show (green bars with corresponding motif plotted on top) that two motifs suffice to surpass the baseline method.*

fault detections in wind turbines (Bessa et al., 2016; Görnitz et al., 2015), explanation of important pixel patches in computer vision (Bach et al., 2015), quantum chemistry (Schütt et al., 2017), and extraction of latent brain states (Porbadnigk et al., 2015). Compared to POIMs, which were restricted to the use of the weighted degree kernel, MFI is applicable to arbitrary kernel functions. Thus, using kernel functions, such as top kernel (Tsuda et al., 2002), which can handle sequences of variable length, we have expanded the scope of possible applications in computational biology. For future work, it would be interesting to investigate the performance of MFI for different kernel functions. MFI can be used to compare various machine learning methods regarding their underlying discriminative features. Therefore, MFI can contribute to a higher confidence in the classifier as for example in BCI (Müller et al., 2003; Müller et al., 2008), where experts can decide whether the extracted discriminative EEG patterns learned by the classifier are informative or not. However, there is one main shortcoming that we face when applying MFI. The number of samples depends on the complexity of the problem. This means that complex problems including high dimensional input spaces may require a large number of samples, which subsequently would increase computation times. For future work, as a compensation, we suggest to

further study the proposed methodology for various sampling methods, such as min-max sampling or cluster sampling to increase efficiency and speed up computation time. Also the use of generative models, such as Generative Adversarial Networks (GANs) (introduced by Goodfellow et al. (2014)) could be investigated as they could generate suitable samples for MFI.
Summarizing, in this chapter, we have contributed to opening the black-box of learning machines. Our proposed method MFI is generally applicable to arbitrary learning machines for either instance-based or model-based explanation of even non-linear coupled features. In particular we could show that our novel MFI nicely extends and unifies existing works (cf. Section 3.1 and 3.2). Thus, MFI is a novel algorithmic tool which profoundly improves flexibility and expressiveness of the POIM family. Various experiments on computer vision data as well as experiments on artificially generated DNA sequences, real-world human splice site data and enhancer data demonstrate the properties and benefits of our approach.

Chapter 5

Covariate Shift Adaptation

Fundamental changes over time of EMG signal characteristics are challenging for myocontrol algorithms controlling prosthetic devices. These changes in the data are generally caused by electrode shifts after donning and doffing, sweating, additional weight or varying arm positions, which results in varying signal distributions – a scenario often referred to as covariate shift. A substantial decrease in classification accuracy due to these factors hinders the possibility to directly translate EMG signals into accurate myoelectric control patterns outside laboratory conditions. To overcome this limitation, we propose the use of supervised adaptation methods. The approach is based on adapting a trained classifier using a small calibration set only, which incorporates the relevant aspects of the non-stationarities, but requires only less than 1 min of data recording. The method was tested first through an offline analysis on signals acquired across 5 days from 7 able-bodied individuals and 4 amputees. Moreover, we conducted a three-day online experiment on 8 able-bodied individuals and 1 amputee, assessing user performances and user-ratings of the controllability. Across different testing days, both offline and online performances improved significantly when shrinking the training model parameters by a given estimator towards the calibration set parameters. In the offline data analysis, the classification accuracy remained above 92% over five days with the proposed approach whereas it decreased to 75% without adaptation. Similarly, in the online study, with the proposed approach the performance increased by 25% compared to a test without adaptation. These results indicate that the proposed methodology can contribute to improve robustness of myoelectric pattern recognition methods in daily life applications. Most parts of this chapter were previously published in Vidovic et al. (2014) and Vidovic et al. (2016b).

5.1 Motivation

Pattern recognition is a promising approach for controlling upper limb prosthetic devices with surface EMG electrodes (Scheme and Englehart, 2011; Hargrove et al., 2010; Englehart and Hudgins, 2003; Hudgins et al., 1993). It enables the user to intuitively control a myoelectric prosthetic hand with multiple degrees of freedom. However, despite the good laboratory performance of this approach, its clinical and commercial impact is still limited. One of the reasons for this limitation is that laboratory conditions often do not include sources of EMG signal non-stationarities, such as electrode shifts following donning and doffing, changes in arm position, variable loads when grasping objects, muscle fatigue, and varying electrode-skin impedance

(Jiang et al., 2012; Farina et al., 2004). These factors have a significant influence on the data distributions and thus on the robustness of the system, as exemplary illustrated in Figure 5.1 for the case of tests on different days.
In this example, a linear discriminant analysis (LDA) classifier (for an introduction to LDA see Section 2.1.4) can separate two classes (hand open and wrist extension) without errors on the first day, when the classifier was trained. However, because of changes in the data distribution, the LDA performance continuously decreased until complete failure on the third day of measurements. The presented scenario, where training and test data distributions differ from each other is known as covariate shift or sample selection bias, which we introduced in Section 2.1.6. Different strategies

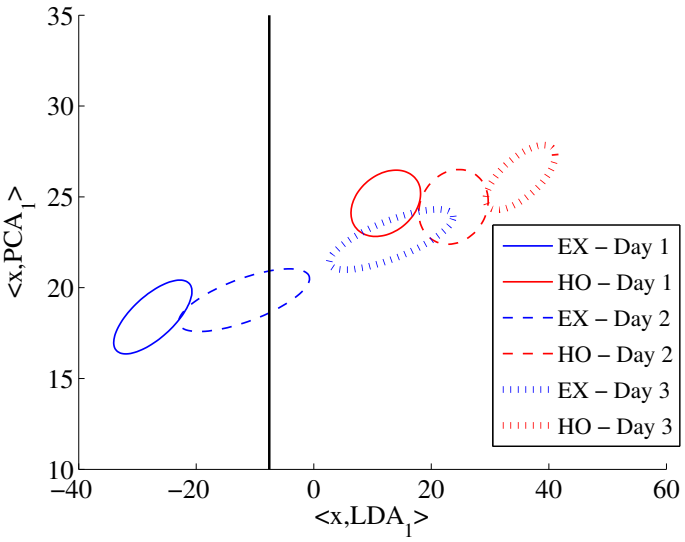


Fig. 5.1: Covariate Shift *Illustration of the effect of doffing-donning of the prosthesis in three days when classifying two movements: Hand Open (HO) and Wrist Extension (EX). The optimal separation hyperplane for the first day data (black vertical line) performs worse on the data of the second day and fails completely on the data of the third day. For a two dimensional representation, the data were projected on the first LDA and PCA component, respectively (Shenoy et al., 2006).*

have been suggested to counteract the covariate shift and improve the robustness of EMG pattern recognition (Sensinger et al., 2009). One approach has been the inclusion of examples of non-stationarities in the training set, which increases the generalization ability of the classifier but requires a large training data set. In addition, there are factors that cannot be easily included during training, such as changes in the way the users perform the attempted tasks. In Scheme and Englehart (2011), a full calibration was performed every day, which, however, is very demanding and time consuming for the user. To avoid this, adaptation strategies have been explored (Chen et al., 2013; He et al., 2012), where the model parameters are updated by samples of the testing data. In a fully unsupervised adaptive approach, the classification accuracy may however decrease, due to the inclusion of mis-classified samples. To address this problem, Sensinger et al. (2009) proposed to estimate the confidence of

the classifier decision, and then to use only the most reliable decisions for adaptation. This approach was outperformed by all supervised adaptation methods, investigated in the same study for comparison. Most of the work done on adaptation have been done offline on pre-recorded data. Although offline studies are useful to compare different algorithms while optimize their parameters, they reflect the actual problem only partly. This is because offline studies cannot consider the fact that the user can react on the algorithmic output and adapt his muscle contractions to improve the outcome in a real-time application (Jiang et al., 2014). Hahne et al. (2015) demonstrated the advantage of concurrent adaptation between the user and the algorithm on myoelectric control.

Extensive research on adaptive classification has been performed also in related fields. For example, covariate shift adaptation (Sugiyama et al., 2007; Sugiyama et al., 2008) has been applied for binary classification problems in brain-computer interface applications (Vidaurre et al., 2011). In this field, the adaptation method suggested by Sugiyama et al. (2007) computed first a probability density on the test set and then the expected test error over test samples. This approach requires, however, a relatively large labeled calibration set, which would imply a large effort by the user.

In this thesis, we address the problem of classification robustness across sessions and days by adapting a myocontrol algorithm, using a short calibration data set that requires only less than 1 min data recording. We test this approach on a large data set that includes both able-bodied subjects and amputees, in offline as well as online conditions. The results indicate that the proposed methodology is an appropriate compromise between a limited user effort in re-training and a substantial improvement in classification accuracy.

The remainder of this Chapter is organized as follows. In Section 5.2 we present the mathematical models, the data sets, and the experimental paradigms used in this study. The results were presented in Section 5.3, and are afterwards discussed in Section 5.4 before we conclude with some final remarks in Section 5.5.

5.2 Method

In this thesis, we focused on the two Bayesian multi-class classifiers: linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) [20], which we introduced in Section 2.1.4. The difference between QDA and LDA is, that QDA determines quadratic boundaries for class separation by using class-wise covariance matrices, whereas LDA eliminates the quadratic terms by assuming equal covariance matrices for all classes and thus uses hyperplanes for classification. These models work well under ideal laboratory conditions for controlling prosthetic devices with EMG signals. However, in the presence of non-stationarities, the true data distribution $p_{te}(x)$ (i.e. the test data distribution) almost never coincides with the training distribution p_{tr}.

In the following, we propose an adaptation methodology to adjust the training model parameters (i.e., the mean and the covariance matrix, see Section 2.1.4) towards the parameters of a very small calibration data set including the current conditions.

The basis for the model adaptation is a small calibration data set $X_{cal} = \{t_i, u_i\}_{i=1}^{m}$, $m \in$

$\mathbb{N}$, which follows the test distribution $X_{cal} \sim p_{te}$, where $t_i \in \mathbb{R}^d$ and $u_i \in \{1, \ldots, C\}$. Let $\mu_{tr\,c}$ and $\Sigma_{tr\,c}$ be the class-wise mean and the class-wise covariance matrix of the training set X and $\mu_{cal\,c}$ and $\Sigma_{cal\,c}$ the ones of the calibration set X_{cal}. We propose an adaptation by shrinking the training parameters towards the ones obtained from the calibration set:

$$\tilde{\mu}_c = (1 - \tau)\mu_{tr\,c} + \tau\mu_{cal\,c}, \tag{5.1}$$

and

$$\tilde{\Sigma}_c = (1 - \lambda)\Sigma_{tr\,c} + \lambda\Sigma_{cal\,c}, \tag{5.2}$$

where τ and $\lambda \in [0, 1]$ are the regularization parameters. To get subject independent reasonable values for τ and λ they were estimated by a grid search with a step size of 0.1 across all days and subjects on the validation data set (see Section 5.2.1). If there is only one shrinkage parameter to be determined, the other is held at zero.
In the following analysis, we denote LDA as the classifier trained on the training data set that served as a baseline for comparison with its adapted versions (same for QDA):

1. LDAMA: LDAM(ean)A(daptation) adapts the mean of LDA towards the mean of the calibration data set, where the covariance matrix remains unchanged.

2. LDACMA: In the LDAC(ovariance)M(ean)A(daptation) method the mean and the covariance matrix of LDA are simultaneously adapted towards the parameters of the calibration data set as shown in Equation (5.1) and (5.2).

3. LDAnew: LDAnew is a new LDA classifier trained on the calibration set only.

4. LDADEA: LDAD(ata)E(xtension)A(daptation) adapts the mean and the covariance matrix, equal to LDACMA, but after each daily adaptation towards the calibration data set, the calibration data set was incorporated into the initially training data set and the LDA, which is used for the ongoing adaptation was retrained on the extended training data set.

5. LDAFA: Unlike LDACMA and LDADEA, which used the non-adapted LDA trained on the training or extended training data set for adaptation, LDAF(urther)A(daptation) repeatedly adjusted the already adapted LDA towards the calibration data set.

5.2.1 Evaluation procedure

Dataset

For the offline analysis, we used the five-day data set of the experiments presented in Amsüss et al. (2014). The data set was obtained from seven able-bodied subjects (five males, two females 25.4±1.4 yrs) and four transradial amputees (males, ages 25, 28, 29, and 64 yrs). All amputees and two able-bodied subjects were experienced with myocontrol, while the remaining able-bodied subjects were naive. Eight commercially available double differential electrodes (13E200=50AC Otto Bock Healthcare Products

GmbH, Vienna, Austria) were used for the data recording. They were placed equidistantly around the forearm of the subjects approximately 7 cm from the olecranon. To mimic a real-life usage, for each amputee (two left and two right side affected), the electrodes were fitted in an individual hard socket prosthesis. For able-bodied subjects the electrodes were mounted on the dominant forearm by a spring-grid, which ensures that the electrodes were slightly pressed on the skin. The electrode positions were marked by a water-resistant pen and the markers were used for mounting the electrodes on the other days.

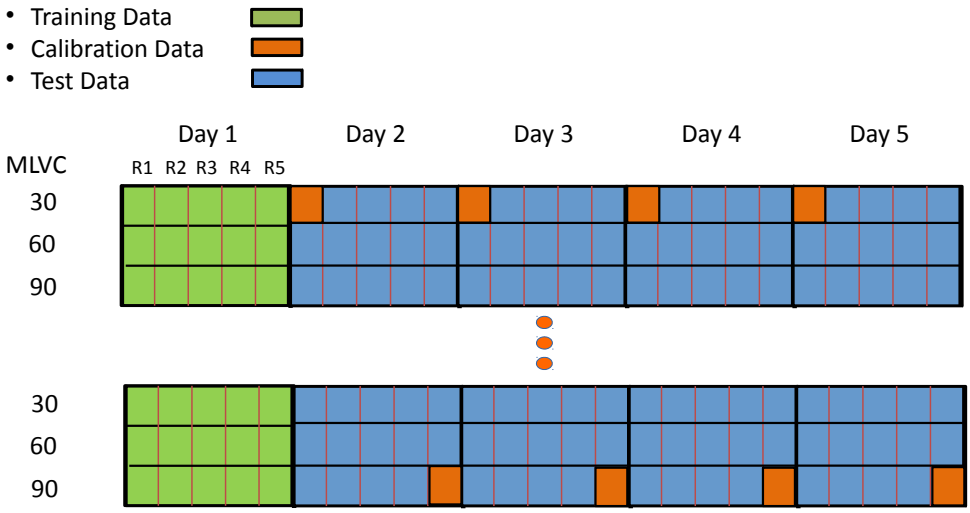


Fig. 5.2: Illustration of the data segmentation. *R_n, $n = 1, \ldots, 5$ includes the trials of all movements for one particular contraction level. The classifier was trained on one day (exemplary on Day 1 (green)), adapted by one run of the test day (orange) and tested on all other runs of the test day (blue).*

Offline experimental paradigm

The subjects were instructed to perform 8 movements: wrist pronation (WP), wrist supination (WS), wrist extension (WE), wrist flexion (WF), hand opening (HO), fine pinch (FP), key grip (KG), and no movement (NM) at three contraction forces (30, 60, and 90% maximum long-term voluntary contraction, MLVC) on five subsequent days (Mon.-Fri.). During the data recordings, they were seated on a chair, holding their forearm parallel to the ground in a 90 degree angle to the upper arm. The calibration phase consisted in performing each movement for 30 s at 100% MLVC. The reference value for normalizing the muscle activity was the average maximum root mean squared (RMS) value over the eight channels $\overline{RMS} = \sum_{p=1}^{8} \sqrt{\frac{1}{l} \sum_{i=1}^{l} (\xi_i^p)^2}$, where l is the number of samples per time-window and ξ^p the instantaneous EMG signal of channel p.

Each subject performed two sessions of data recordings on five subsequent days. Between the sessions the amputees performed a donning and doffing, as they would do during daily use. For the able-bodied subjects, the donning and doffing was mimicked by a lateral displacement of the electrodes by 0.8 cm. One session took approximately 45 min and, since each movement was recorded five times at the three contraction forces, it comprised 120 trials ($3 \times 5 \times 8 = 120$ trials). For each 5-s trial the user followed a trapezoidal force profile while doing the instructed movement, following a cursor representing the $\overline{RMS}$ value. For the first second, the user followed the trapezoidal ramp up to the given force level, where the force level was maintained for 3 s before the user followed the ramp down to the No-Movement level. In the following we will refer to one run as the set of 8 trials, in which each class is represented once. The division of the data into training, test, and validation set was done as follows. The first session of the first day was used as training set and the first session of the subsequent days were used as test sets. The second session of each day was used as the validation set for the parameter optimization of our proposed adaptation method (following standard cross-validation schemes see Lemm et al. (2011)).

Signal acquisition and processing

The acquired raw signals were amplified to the range 0-4.5 V and filtered in a bandwidth of 20-450 Hz. Moreover, a 50-Hz notch filter was included in the active Otto Bock electrodes. The filtered signals were sampled at 1 kHz, digitized by a 10 bit A/D converter, and transferred via Bluetooth to a computer by the Axonmaster (Otto Bock HealthCare Products GmbH Vienna, Austria).

Feature extraction

For this study, we calculated the logarithm of the signal variance (logVar), as proposed in Hahne et al. (2012), in intervals of 250 ms, which overlapped by 50 ms (15 features per channel and per trial).

Subject variability

To investigate the inter-subject variability concerning false class label predictions we visualized the non-diagonal elements of the confusion matrices, reshaped as a single-column vector for each subject. Moreover, to indicate the changes of the variability of false predictive class labels over days, we included the mean confusion matrix of all days but training too.

General adaptation procedure

The data (first session of each day) were split as shown in Figure 5.2, where the entire data set of one day (green) was used as initial classifier training set. The other data set was further separated into a calibration set that comprised one run (orange) and in the testing set, consisting of the other 14 runs (blue). In a first step, the LDA was trained on the entire training data set. Afterwards, the LDA was adapted by the

calibration data set. Finally, the performance of the adapted LDA was calculated on the test set. A reverse leave one out cross validation was performed by taking each run of one day as calibration data set and computing the performance on the remaining test set.

Optimal shrinkage parameter

First, we trained the LDA on the entire data set of the first day. We accomplished the computation of the optimal shrinkage parameters on the validation data set, i.e., the second session of each day, where we averaged over the daily results. To get a subject-independent adaptation of the LDA, we computed the optimal shrinkage parameter values for τ and λ by a grid search that included the data from all subjects. We adapted the LDA by all possible combination pairs of τ and λ with elements in $\{0, 0.1 \ldots, 1\}$ towards one run of the validation set. For the case of LDAMA, the covariance adaptation was held to zero with $\lambda = 0$. For each parameter pair, we adapt with 15 different runs for each day and each subject. To reveal the error we made for each subject by taking the average optimal shrinkage values over all participants instead of the single subject specific values we computed the Euclidean distance between both: LDACMA adapted with the subject average regularization values and LDACMA adapted with the subject-specific τ and λ values.

5.2.2 Online Experiment

To evaluate the adaptation performance for a real time application, we conducted an online experiment including one amputee and 8 able-bodied subjects. One able-bodied subject and the amputee were experienced with myocontrol, while all other subjects were naive. The experiment consisted of a training day (day zero) and three test days (day 1-3). The initial data recording on the training day involved 3 trials for each of the 3 contraction strengths for the 8 movements and served the LDA as training data set.
On each test day, we recorded a small calibration data set, where the user was instructed to perform each movement in a perceived 60 % force level for 4 s once, with no feedback but a time progress bar. This data set was used to train LDAnew and to adapt LDACMA, where the latter was adapted by the optimal subject independent parameters found by the grid search (5.2.1). To compare the performance of LDA, LDAnew, and LDACMA, we conducted a 1-min online test, twice for each LDA variation. The user was unaware of the control algorithm tested in each case. During the 1-min test the user was asked to copy a sequence of different target movements. Two pictures, side by side, were presented to the user on a screen. The target movement was shown on the left accompanied by a vocal cue. The subject was asked to replicate the given target movement as fast as possible and was provided with visual feedback of the movement decoded by the algorithm under test (direct user feedback). A hit was counted if the user reached the target movement and held it for 1 s. Afterwards the next target movement was shown on the left. In total, the user had 10-s to reach a hit. After each test iteration, the subjects provided a subjective indication on the comfort in controlling the specific system, on a scale between 1 and 10.

5.3 Empirical Evaluation

First, we present the systematic class confusion within the training day as well as within the subsequent days for all subjects. The significant performance drop on the other days motivate an adaptation. According to that, we present the results for the optimal shrinkage parameter choice, which gave the basis for the further classifier adaptation computations. Subsequently, we report the offline results on the five-day data set for each adaptation strategy. Then we present the influence of the contraction strength to the adaptation performance and finally we give the results of the online experiment.

5.3.1 Systematic class confusion

We investigated whether or not there exists a systematic class dependency within the confusion matrix between the training day (TD) and the other days (OD). We observed an increased number of false predicted class labels in OD compared to TD. This was also reflected in a decreased classification accuracy in OD (see accuracy on top of each column). There was a significant dependency between the confusion matrix of TD and OD (Fisher's exact test: $p < 10^{-10}$). Furthermore, we observed a high variability of false predicted class labels over subjects (Figure 5.3). The wrong predicted class labels of the best performing amputee, AP 1, yield a classification error lower than 0.5% and are thus not visible. The other amputees perform worse than the able-bodied subjects, which can be seen by the greater number of false predicted classes.

5.3.2 Optimal shrinkage parameter choice

To obtain subject-independent values for τ and λ we computed the optimal shrinkage parameter by a grid search (Figure 5.4), including all subjects as described in Section 5.2.1. The best values obtained for LDACMA were $\tau = 0.8$ and $\lambda = 0.8$. For the case of LDAMA, the covariance adaptation was held to zero with $\lambda = 0$, which gave an optimal mean shrinkage value of $\tau = 0.7$.
The error between LDACMA adapted with the subject average regularization values and LDACMA adapted with the subject specific τ and λ values is shown in Figure 5.5 for each subject. For the able-bodied subjects (subjects 1 to 7) the mean error was lower than 2%. For the amputee A1, the shrinkage parameters were almost perfect (err $< 1\%$), whereas the error for the amputee A3 was comparatively large (err $\approx 8\%$). For the other two amputees the mean errors were approximately 4%.

5.3.3 Covariate shift adaptation

In a first step, we trained a LDA classifier on the entire data set of the first day session of the able-bodied subjects. Using a 5-fold cross validation, we achieved high classification results with an accuracy $> 95\%$ when testing the same subjects on the same day. When testing the LDA classifier on the following days without retraining,

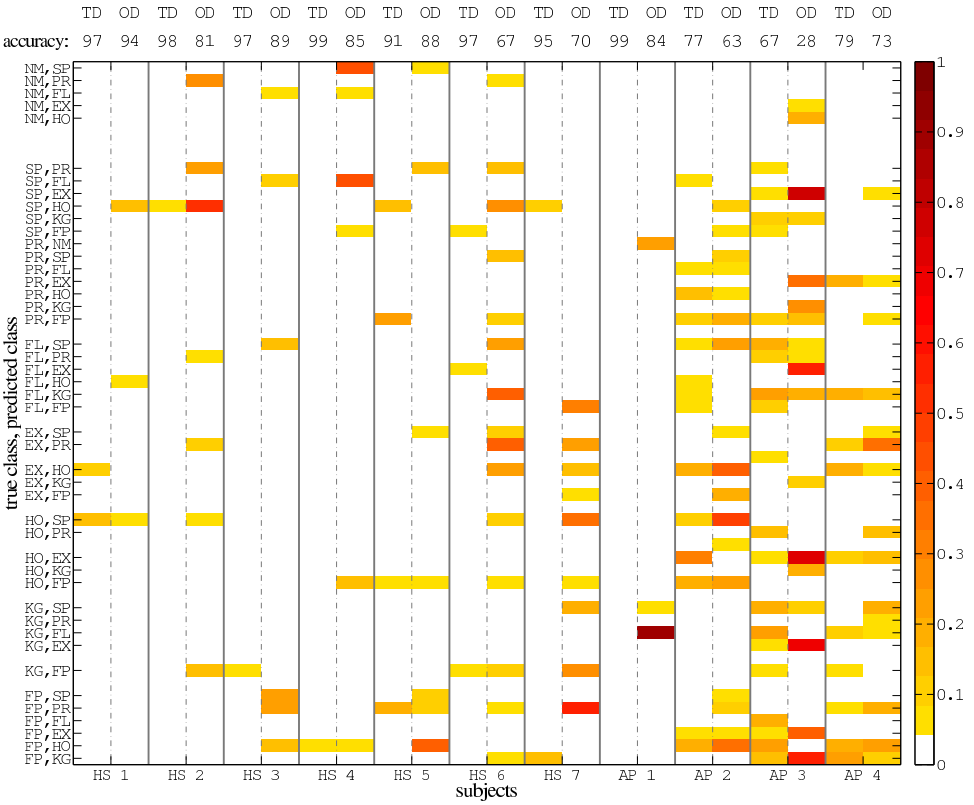


Fig. 5.3: *Vectorial representation of the non-diagonal elements of the confusion matrix for the able-bodied subjects (HS 1-7) and for the amputees (AP 1-4). The false predictive class labels are given as pair (M1, M2) on the y-axis, where M1 is the real class and M2 the predicted class and the degree of misclassification is given by the color, where dark red indicates a high value and lighter colors a lower value of false positives. Moreover, the false predictive class labels are shown for the training day (TD) in the first column and as average over all other days (OD) in the second column for each subject. The accuracy for each subject is reported on the top of each column for both TD and OD.*

however, the performance dropped substantially by almost 20%, as shown in Figure 5.6 a). Further, we investigated the performance of the proposed adaptation methods. The optimal shrinkage parameter values for τ and λ (see Equation (5.1) and (5.2)) were obtained by a grid search, as shown in Figure 5.4.

We first evaluated the mean adaptation, where we shrank the mean of the LDA towards the mean of the calibration set (LDAMA: ($\tau = 0.7$, $\lambda = 0$)). Afterwards, we extended the mean adaptation by an additional covariance matrix regularization (LDACMA: ($\tau = 0.8$, $\lambda = 0.8$)). Finally, we trained an LDA on the small calibration set only (LDAnew: ($\tau = 1$, $\lambda = 1$)). Regarding the baseline LDA performance, the LDAMA significantly improved the classification accuracy for each day. Although the LDA performance continuously decreased from day 2 to day 4, the LDAMA performance remained constant above 90%. Compared to the mean adaptation, LDACMA could

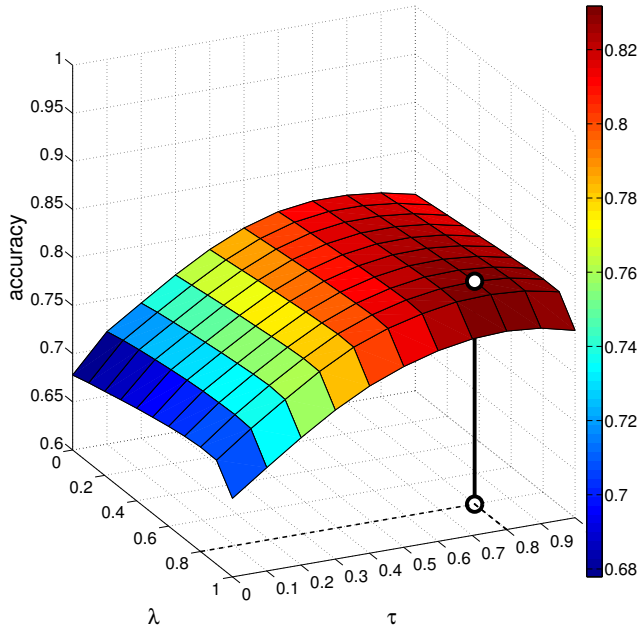


Fig. 5.4: Illustration of the grid search. *The computation includes all eleven subjects for finding the optimal adaptation parameters τ and λ for LDA. A continuous performance increase was preserved by shrinking the mean of the training set towards the mean of the calibration set, where the optimal value was $\tau = 0.7$ ($\lambda = 0$). By shrinking additionally the covariance matrix, the optima were found by $\tau = 0.8$ and $\lambda = 0.8$, a further, but relatively small performance gain was obtained.*

not gain any further performance increase. The LDAnew also significantly improved the LDA performance, but performed worse than both adaptation methods (see Figure 5.6 a)).

A similar trend was observed with the results of the amputee subjects, although the classification accuracy was lower by more than 10% compared to the able-bodied subjects (see Figure 5.6 b). Contrary to the results on able-bodied subjects, LDACMA significantly outperformed LDAMA for amputees. LDAnew had worse performance than LDAMA for the first two days but showed a similar performance afterwards.

We performed a two-way repeated ANOVA test, including all eleven subjects, to compute the statistical differences between the methods. The statistical test results showed that LDAMA and LDACMA significantly outperform LDA ($p = 0.03$ and $p = 0.016$), whereas no significant difference was obtained when comparing LDA and LDAnew. We repeated the equivalent experiment for QDA, where the optimal shrinkage parameters for the mean adaptation (QDAMA: ($\tau = 0.8$, $\lambda = 0$)) and the covariance matrix adaptation (QDACMA $\tau = 0.7$, $\lambda = 0.9$)) were extracted by a grid search (similar to the results of the LDA grid search as shown in Figure 5.4). For the able-bodied subjects, we observed similar behaviors for QDA, QDAMA, QDACMA, and QDAnew compared to the LDA counterparts. The QDA performance dropped from 97% on the training day to 70%. Adapting the mean, QDAMA attained a

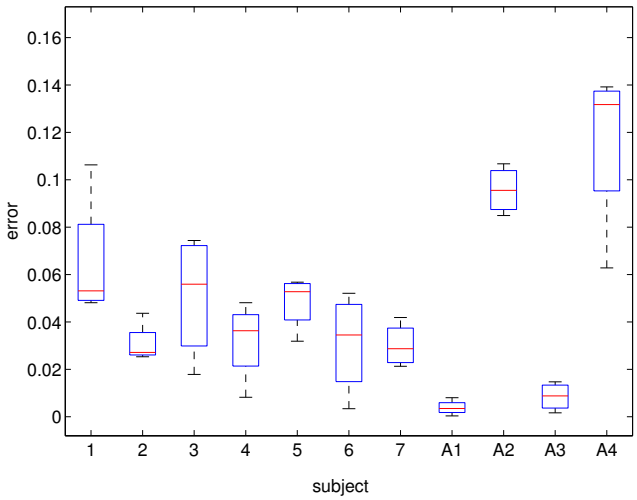
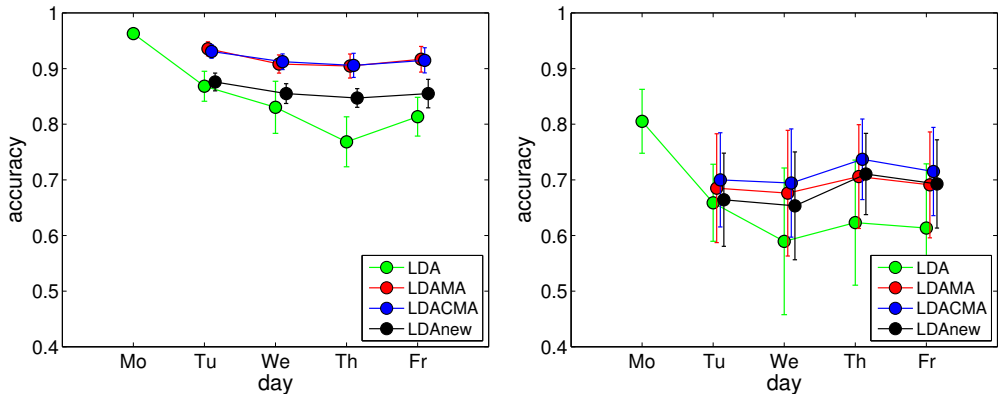


Fig. 5.5: Illustration of the subject specific errors *when adapting the LDA by the mean regularization parameter values extracted from the grid search including all subjects instead of the best parameter choice for the respective subject. The able-bodied subjects were tagged by 1-7 and the amputees by A1-A4.*

significant performance raise of 20%. Moreover, unlike the additional LDA covariance matrix adaptation, QDACMA significantly outperformed QDAMA. QDAnew reached no significant improvement (Figure 5.6 c)).
From the QDA results of the amputee data shown in Figure 5.6 d), we observe again that QDACMA performed best, followed by QDAMA. Compared to the LDA results, QDAnew performed significantly worse than QDAMA. Also in the case of QDA, the statistical test results on all subjects show that QDAMA and QDACMA outperform QDA ($p = 0.013$ and $p = 0.010$), and no significant difference was observed between QDA and QDAnew.
The most relevant difference between LDA and QDA was that LDA was more robust against inter-day non-stationarities. For this reason, the subsequent results are presented for LDA only.
The above results were presented for the adaptation of the initial classifier trained on the training day with the calibration set of the current day. However, we need to clarify whether it is worthwhile for an ongoing classifier adaptation to extend the training data set by the calibration data (LDADEA) or further adapt LDACMA instead of the initial LDA (LDAFA). The results showed that LDACMA significantly outperformed LDAFA ($p < 10^{-4}$, Figure 5.7 a)), whereas there was no significant difference between LDADEA and LDACMA (Figure 5.7 b)).

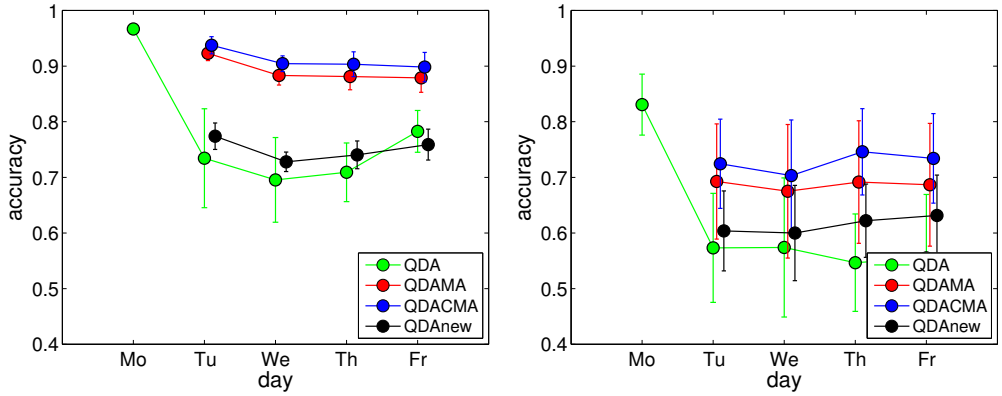
5.3.4 Influence of force level

We repeated the analyses described above, when adapting with the different contraction strengths (30, 60 and 90%). Additionally, we used each day once for training and all other days for testing.



(a) LDA able-bodied subjects

(b) LDA amputees

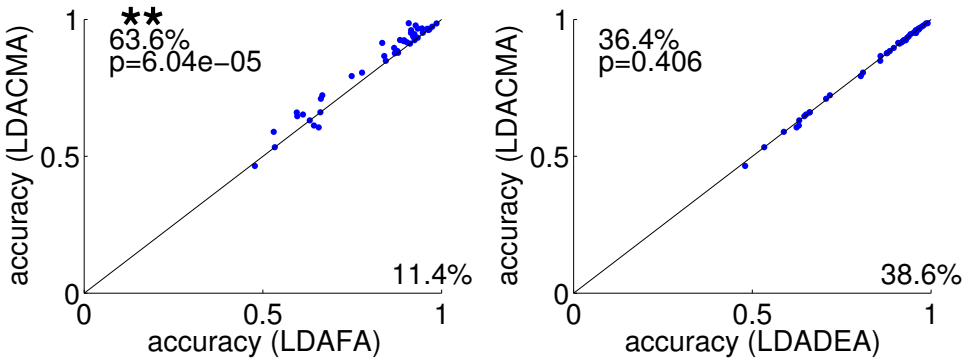


(c) QDA able-bodied subjects

(d) QDA amputees

Fig. 5.6: Illustration of LDA and QDA adaptation performance *for the able-bodied subjects in a) and c) as well as for the amputees in b) and d). In all cases, a remarkable improvement of classification accuracy across days is obtained by the mean adaptation. The additional adaptation of the covariance matrices only results in a slight improvement in the case of QDA. For comparison, the classifier trained exclusively on the new calibration data slightly improves the classification accuracy but performs worse than the adapted classifiers. Note that the variation between subjects is plotted as standard error.*

In Figure 5.8 the mean LDACMA performance of the test days are shown and compared to the baseline LDA performance. We performed a two-way repeated ANOVA test over all subjects to make a general statement about the statistical differences between the methods. We observe that an adaptation with 60% contraction strength significantly outperformed the baseline LDA ($p = 0.019$), whereas no significance difference was observed for an adaptation with 30 or 90% contraction strength ($p = 0.603$ and $p = 0.082$). Among the adaptation models, a 60% adaptation only outperformed the 30% adaptation ($p = 0.012$) but not the 90% one.



(a) LDACMA vs LDAFA (b) LDACMA vs LDADEA

Fig. 5.7: Comparison between the performance of three adaptation strategies *LDACMA, LDAFA, and LDADEA for all subjects and all days but the first day, which was used for training.*

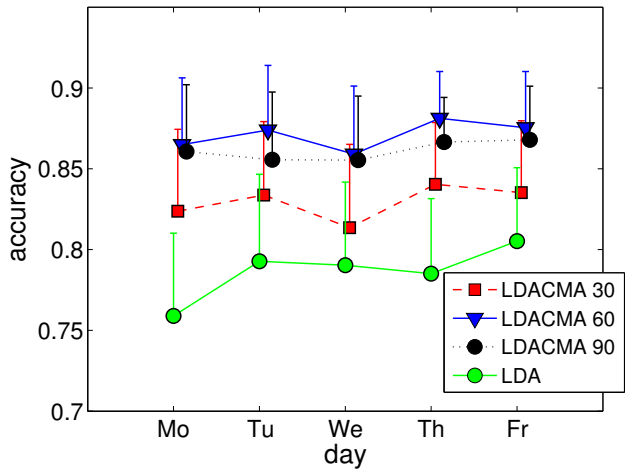
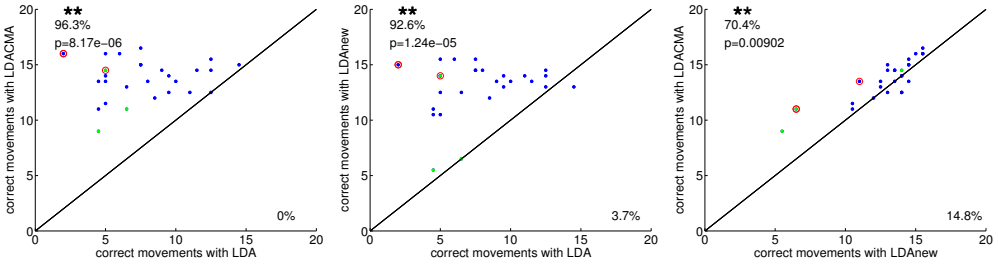


Fig. 5.8: The daily comparison of the LDACMA performance when adapting with the single contraction strengths of 30, 60 and 90%, respectively. *Using a calibration set with 60% contraction strength for adapting LDA indicated the best performance.*

5.3.5 Online Experiment

During the 1-min online tests, we counted the number of correctly imitated movements that should be held by a subject for 1 s. On three test days, each algorithm (LDA, LDACMA, and LDAnew) was tested twice.

First, we plotted the performance of the methods against each other as shown in Figure 5.9. From Figure 5.9 a) and b) we observe that LDACMA as well as LDAnew significantly outperformed LDA for all able-bodied subjects (blue dots) as well as for the amputee (green dots). When comparing LDACMA and LDAnew, we observe that for 70.4% of the subjects LDACMA performed better than LDAnew, where only



(a) LDACMA vs LDA (b) LDAnew vs LDA (c) LDACMA vs LDAnew

Fig. 5.9: Illustration of the 1-min online test results. *The subjects (8 able-bodied and 1 amputee) were asked to imitate certain movements for one second. In total, two tests (runs) were recorded on 3 different days (sessions). The scatter plots illustrate the number of correct classified movements by LDA, LDACMA, and LDAnew against each other. Each circle corresponds to the mean results of one session for one subject, where the amputee data is marked by green circles. The percentage on each side of the partition line represents the amount of results of the respective method.*

14.8% of the subjects got a performance gain (Figure 5.9 c)). The test run, which benefited the most from our approach (LDACMA) is highlighted in red, once for the able-bodied subjects and once for the amputee.

For further analysis, we additionally incorporated the performance of LDAnew as well as the user controllability ratings of each method. In order to be able to show simultaneously those joint effects, we standardized the data. This was done subject-wise by z-scoring, i.e. first removing the mean and then dividing by the standard deviation. Each line in the polar plot shown in Figure 5.10 indicates one run of one subject. The algorithm performance is reported on the x-axis and the user controllability rating on the y-axis. The majority of the subjects valued the controllability using LDA worse than using LDACMA or LDAnew. More precisely, 96% of the LDA results pointed in the direction of decreased user-rating, whereof 94% also pointed in the direction of decreased performance. On the contrary we record for LDACMA an increased user rating of 96% and 87% for LDAnew. Even when the LDACMA performance decreased, the user still perceived a pleasant controllability unlike LDAnew, where the user rated the algorithm lower although an increased performance was recorded.

In order to visualize the joint effects of performance and user-rating for each subject individually, we reported both against each other as shown in Figure 5.11. In Figure 5.11 a) the difference between achieved user-hits with LDACMA and LDA is reported on the x-axis and the difference between user controllability ratings of LDACMA and LDA on the y-axis. The test results of each subject were highlighted by different colors, where subject 2, colored in orange indicated the test results of the amputee. The user performance was positively associated to the user controllability across the subjects. Additionally, Figure 5.11 a) shows that for all test iterations LDACMA outperformed LDA (same as shown in Figure 5.9). We observed a similarity in

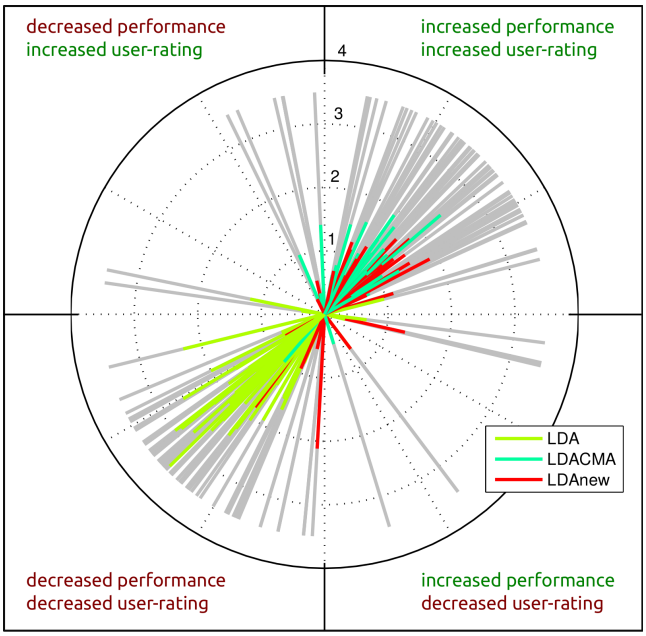
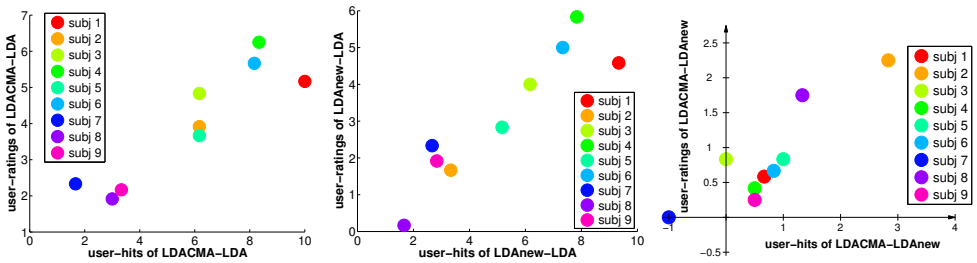


Fig. 5.10: Joint illustration of the user performance and user-rating. *The classifier performance (x-axis) is plotted against the user controllability rating (y-axis) for LDA, LDACMA, and LDAnew. To show simultaneously those joint effects, we standardized the data by subject-wise z-scoring. Each line in the polar plot indicates one session run of one subject.*

results between LDAnew and LDA as shown in Figure 5.11 b). When comparing LDACMA with LDAnew 7 of 9 subjects had both higher performance and higher user controllability rating with LDACMA. Although one able-bodied subject performed better with LDAnew, there was no difference between LDACMA and LDAnew in the user controllability rating. For one subject, there was no significant difference between the performance with LDACMA or LDAnew, but the subject preferred the controllability with LDACMA.

5.4 Discussion

LDA was more robust than QDA for both able-bodied subjects and amputees. This may be caused by the fact that QDA is highly dependent on the class-wise covariance matrices, which were estimated with a relatively small number of samples and therefore less stable and prone to overfitting. Also little changes in EMG signals may determine large changes in the data distributions, with the consequence that the QDA quadratic boundaries might fail in movement classification. Conversely, LDA uses the pooled covariance matrix, where the non-stationarities of the EMG signals have little influence. This was shown by the fact that the classification accuracy over days of LDA decreased less than for QDA on the one hand (Figure 5.6). On the other hand, the additional covariance matrix adaptation had less improvement for LDA than for QDA, since LDACMA performed similar to LDAMA. Our results are in line with the findings in



(a) LDACMA vs LDA (b) LDAnew vs LDA (c) LDACMA vs LDAnew

Fig. 5.11: Subject specific behavior when comparing the classifier performance with the controllability rating. *The results are shown as differences between the two methods, respectively, where the latter is subtracted from the first. Exemplary, in a), the x-axis show the hit difference between LDACMA and LDA, hence: hits of LDACMA minus hits of LDA (same for the user controllability ratings). Each circle indicates the average result of all session runs for one subject, where each subject is colored differently.*

Chen et al. (2013), where a self-enhancing mean and covariance matrix adaptation was presented, first separated from each other and then together. Also in that study, the class mean adaptation gave the largest effect in performance compared to the covariance matrix adaptation for LDA. The results suggest that for real-world EMG control applications LDA might be a better choice than QDA since it is more stable over time.

The presented adaptation method, where we adapted the trained classifier towards a short re-calibration set showed a high performance gain, both in offline and online analysis.

The LDACMA adaptation with 60% force level performed best since in general the overall mean, consisting of data with 30, 60, and 90% contraction strengths, is most closely to the data with 60% contraction strength. In the offline analysis LDAnew outperformed LDA but performed significantly worse than LDACMA. In the online analysis, although the performance of LDAnew was comparatively similar to LDACMA (in average LDACMA achieved one hit more than LDAnew), the user rated the controllability with LDACMA higher, which is shown in Figure 5.11 c). Although, one able-bodied subject performed better with LDAnew, there was no difference for him/her in the controllability between LDACMA and LDAnew. For one able-bodied subject there was no difference in the performance between LDACMA and LDAnew, however, he/she rated the controllability with LDACMA higher compared to LDAnew. Furthermore, the results from Figure 5.9 c) indicate that LDACMA significantly outperformed LDAnew ($p < 0.01$). But still, it is interesting that LDAnew performed much better in the online test than in the offline analysis. This can be explained by the adaptation of the user, which was possible with the given visual feedback in the online tests but not in the offline data collection.

The main difference in performance between able-bodied and amputee users was a lower classification accuracy for the amputees, which is also reported in Huang et al. (2005), Amsüss et al. (2014), and Hwang et al. (2014) and emphasized in Figure 5.3

by the false predicted class labels. Moreover, the variability of results was significantly greater for amputees. This is due to the fact that able-bodied subjects performed actual movements of the hand while amputees could only attempt it without a visual and sensory feedback from the hand. For amputees, the absence of visual feedback on the actual hand movement may also cause the impossibility of generating well distinguished muscle activations for the different tasks. It should be noted, that on average no absolute recognition accuracy difference was found between the experienced and inexperienced subjects, since the data set we used incorporated an extensive training (2 hours for 5 days) (Amsüss et al., 2014).
Unsupervised adaptation is sensitive to wrong adaptation, as observed, e.g., in He et al. (2012) where all the unsupervised adaptation variations tested showed an increased error rate over time. Thus, fully unsupervised adaptation is not a robust approach at least not until the influence of mis-classification is minimized by a high confidence in judging the accuracy of the test data.
Our approach thus provided a reasonable trade-off between high classification accuracy over days and a minimal effort by the user in re-training; we therefore consider it a practical method in real-world EMG control applications.
In this thesis, we considered the same adaptation for all classes. However, there are movements that are classified with lower accuracy than others as shown in Figure 5.3. Note that the non-stationarities influence the class distributions differently. In future work, it will be interesting to adapt parameters differently for each class and to determine subject-specific optimal shrinkage parameters (Figure 5.5). Moreover, subclass structures can be modeled in the classifier as presented in Höhne et al. (2016), which may additionally improve our presented adaptation technique and could be considered in future work. Also the extension of our approach to regression based myoelectric control (Hahne et al., 2014; Hwang et al., 2014) remain as future work.

5.5 Conclusion

Pattern recognition is of high benefit for controlling myoelectric prosthetic devices but can lack robustness when tested in daily-life conditions. Covariate shifts cause changes in the data distribution, which then may lead to a reduced classifier performance. Offline and online results of our proposed adaptation of the initial classifier towards a short, less than 1 min newly recorded data set demonstrated the significant gain in classification accuracy for both, able-bodied subjects and amputees over days. In offline experiments, LDA indicates a higher stability than QDA due to the pooled covariance matrix and might be the better choice for real-world applications. Furthermore, the relative improvement trends achieved with our proposed methodology were the same for all subjects, which underlines the relevance of our method for both experienced and novel users. In conclusion, the proposed adaptation approach can be used as a practically feasible method for improving the robustness of pattern recognition for myocontrol.

Chapter 6

Summary and Outlook

Due to the fact that most machine learning algorithms act like black-boxes and give no reasoning for their decisions, the first goal of thesis was to improve upon the interpretation of machine learning algorithms. We started by advancing POIMs, a method for visualizing the feature importances learned by an SVM with a WD kernel. Due to the fact that the size of POIMs grows exponential with the size of the motif, POIMs are only applicable to reveal small motifs. In this thesis, we developed a model-based method called motifPOIMs to extract the relevant motifs from POIMs, which are discriminative for the SVM decision. More precisely, motifPOIMs can extract arbitrary long and even overlapping motifs solving of a non-convex optimization problem. However, we found practical limitations due to the high number of local minima in the non-convex optimization problem. Due to the need of carefully chosen initialization parameters for the motifs and the highly varying results for different initializations, we improved on motifPOIMs and reached a convex optimization problem that provides us with more reliable results without the need of a proper motif value initialization. Besides the increased robustness, convex motifPOIMs also reduced the runtime by several orders of magnitude. We have demonstrated the usefulness of motifPOIMs and convex motifPOIMs for biological problems. The discriminative motifs could be correctly extracted by our methods, which was shown in various experiments on synthetic data and real-world human splice data and we could achieve even better and faster results than the state-of-the art competitor MEME.
So far, we treated model-based explanation methods for an SVM in combination with a WD kernel applicable to sequential data with categorical features.
Motivated by the need of a sample-wise explanation method, which is also applicable to arbitrary learning machines and arbitrary data and feature representations (not only WD kernel), we developed a method called measure of feature importance (MFI), which accomplishes all these demands. Thus, with MFI, we constructed a method for both instance-based and model-based explanations. MFI is applicable to arbitrary learning algorithms, such as CNNs, SVMs, decision trees without the need of knowing the underlying architecture for the learning machine, which makes it a convenient approach when comparing various algorithms regarding their discriminative features. Furthermore, all kinds of data, such as images, sequences, graphs or text can be processed by MFI as well as arbitrary feature representations. The kernelized version of MFI, allows also to search for non-linearly coupled features. Overall the experiments highlight the capacity of MFI to extract the truly relevant discriminative features, which on the one hand underlie the entire learning system and on the other hand drove single classification decisions. Unfortunately, due to the fact that MFI is based on sampling we face the shortcoming that the number of samples depends on the complexity of the problem. Therefore, future work remains to study the proposed

methodology for various sampling methods, such as min-max sampling or cluster sampling or the use of generative adversarial networks (GANs) to increase efficiency and speed up computation time.
The second part of this thesis alleviates the performance decrease of classification algorithms triggered by environmental changes. We extensively discussed the phenomenon of covariate shift and its impact on LDA and QDA classification accuracies for the example of myocontrol algorithms for prosthetic devices. Reducing the effect of the selection bias by including examples of non-stationarities in the training set, may increase the generalization ability of the classifier but requires a large training data set and an unacceptable effort on the user site. In addition, there are factors that cannot be easily included during training, such as changes in the way users perform the attempted tasks. In order to avoid extensive training while maintaining a consistent classification accuracy and a robust translation of EMG signals into accurate myoelectric control patterns, we presented a supervised adaptation technique. Our method adapts the parameters of the trained classifier towards the current data conditions, incorporated in a small, less than 1 min, calibration data set. Thereby, the adaptation parameters were estimated over all subjects. Empirical evaluations showed promising results for both able-bodied subjects and amputees. Interactive online experiments additionally underline the practical usefulness of our approach. The investigation of a subject specific parameter estimation may be subject to future research. Also an additionally coupling of the adaptation strength with the current calibration set could be considered by introducing a measure of confidence. Furthermore, a class specific adaptation could yield to a further increased classification accuracy. Besides classification, it would be interesting to extend our approach also to regression based myoelectric control.

To conclude, this thesis contributed several improvements for machine learning algorithms towards interpretability and accuracy. All contributions were theoretically founded and their performances were demonstrated experimentally on benchmark data as well as on real-world applications.

Bibliography

Abeel, T, Y Van de Peer, and Y Saeys (2009). "Toward a gold standard for promoter prediction evaluation". In: *Bioinformatics* 25.12, pp. i313–i320.

Alipanahi, B, A Delong, MT Weirauch, and BJ Frey (2015). "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning." In: *Nat Biotechnol* 33.8, pp. 831–838.

Alt, W (2011). *Nichtlineare Optimierung*. Springer Vieweg.

Amsüss, S, PM Göbel, N Jiang, B Graimann, L Paredes, and D Farina (2014). "Self-correcting pattern recognition system of surface EMG signals for upper limb prosthesis control". In: *Biomedical Engineering, IEEE Transactions on* 61.4, pp. 1167–1176.

Arbib, MA (2003). *The handbook of brain theory and neural networks*. MIT press.

Bach, S, A Binder, G Montavon, F Klauschen, KR Müller, and W Samek (2015). "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7, e0130140.

Baehrens, D, T Schroeter, S Harmeling, K Hansen, and KR Müller (2010). "How to Explain Individual Classification Decisions Motoaki Kawanabe". In: *Journal of Machine Learning Research* 11, pp. 1803–1831.

Bailey, TL, C Elkan, et al. (1994). "Fitting a mixture model by expectation maximization to discover motifs in bipolymers". In: *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, pp. 28–36.

Bailey, TL, J Johnson, CE Grant, and WS Noble (2015). *The MEME Suite*.

Ben-Hur, A, CS Ong, S Sonnenburg, B Schölkopf, and G Rätsch (2008). "Support vector machines and kernels for computational biology". In: *PLoS Comput Biol* 4.10, e1000173.

Bessa, IV de, RM Palhares, MFSV D'Angelo, and JE Chaves Filho (2016). "Data-driven fault detection and isolation scheme for a wind turbine benchmark". In: *Renewable Energy* 87, pp. 634–645.

Binder, A, S Bach, G Montavon, KR Müller, and W Samek (2016). "Layer-wise relevance propagation for deep neural network architectures". In: *Information Science and Applications (ICISA) 2016*. Springer, pp. 913–922.

Bishop, C (2006). *Pattern Recognition and Machine Learning*. Springer.

Blankertz, B, S Lemm, M Treder, S Haufe, and KR Müller (2011). "Single-trial analysis and classification of ERP components—a tutorial". In: *NeuroImage* 56.2, pp. 814–825.

Boser, B, I Guyon, and V Vapnik (1992). "A Training Algorithm for Optimal Margin Classifiers". In: *COLT*. Ed. by D Haussler. ACM, pp. 144–152.

Bousquet, O, S Boucheron, and G Lugosi (2004). "Introduction to statistical learning theory". In: *Advanced lectures on machine learning*. Springer, pp. 169–207.

Boyd, S and L Vandenberghe (2004). *Convex optimization*. Cambridge university press.

Chen, X, D Zhang, and X Zhu (2013). "Application of a self-enhancing classification method to electromyography pattern recognition for multifunctional prosthesis control". In: *Journal of neuroengineering and rehabilitation* 10.1, p. 44.

Chung, KL, YL Huang, and YW Liu (2007). "Efficient algorithms for coding Hilbert curve of arbitrary-sized image and application to window query". In: *Information sciences* 177.10, pp. 2130–2151.

Cortes, C and V Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Cortes, C, M Kloft, and M Mohri (2013). "Learning Kernels Using Local Rademacher Complexity". In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., pp. 2760–2768.

Dafner, R, D Cohen-Or, and Y Matias (2000). "Context-based Space Filling Curves". In: *Computer Graphics Forum*. Vol. 19. Wiley Online Library, pp. 209–218.

Duchi, J, E Hazan, and Y Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization". In: *Journal of Machine Learning Research* 12.Jul, pp. 2121–2159.

Englehart, K and B Hudgins (2003). "A robust, real-time control scheme for multifunction myoelectric control". In: *IEEE Trans. Biomed. Eng.* 50.7, pp. 848–854.

Farina, D, R Merletti, and RM Enoka (2004). "The extraction of neural strategies from the surface EMG". In: *Journal of Applied Physiology* 96.4, pp. 1486–1495.

Furey, TS, N Cristianini, N Duffy, DW Bednarski, M Schummer, and D Haussler (2000). "Support vector machine classification and validation of cancer tissue samples using microarray expression data". In: *Bioinformatics* 16.10, pp. 906–914.

Goodfellow, I, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, A Courville, and Y Bengio (2014). "Generative adversarial nets". In: *Advances in Neural Information Processing Systems*, pp. 2672–2680.

Görnitz, N, M Kloft, and U Brefeld (2009). "Active and semi-supervised data domain description". In: *ECML*. Springer, 407–422.

Görnitz, N, M Kloft, K Rieck, and U Brefeld (2009). "Active learning for network intrusion detection". In: *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*. ACM, pp. 47–54.

Görnitz, N, AK Porbadnigk, A Binder, C Sanelli, M Braun, KR Mueller, and M Kloft (2014). "Learning and Evaluation in Presence of Non-i . i . d . Label Noise". In: *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 33.

Görnitz, N, M Braun, and M Kloft (2015). "Hidden Markov Anomaly Detection". In: *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1833–1842.

Görnitz, N, M Braun, and M Kloft (2015). "Hidden markov anomaly detection". In: *International Conference on Machine Learning*, pp. 1833–1842.

Gretton, A, O Bousquet, A Smola, and B Schölkopf (2005). "Measuring statistical dependence with Hilbert-Schmidt norms". In: *ALT*. Vol. 16. Springer, pp. 63–78.

Hahne, JM, F Biessmann, N Jiang, H Rehbaum, D Farina, F Meinecke, KR Müller, and L Parra (2014). "Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22.2, pp. 269–279.

Hahne, JM, S Dähne, HJ Hwang, KR Müller, and LC Parra (2015). "Concurrent adaptation of human and machine improves simultaneous and proportional myoelectric control". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23.4, pp. 618–627.

Hahne, J, B Graimann, and KR Müller (2012). "Spatial filtering for robust myoelectric control". In: *IEEE Trans. Biomed. Eng.* 59.5, pp. 1436–1443.

Hargrove, LJ, EJ Scheme, KB Englehart, and BS Hudgins (2010). "Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis". In: *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 18.1, pp. 49–57.

Hastie, T, R Tibshirani, J Friedman, T Hastie, J Friedman, and R Tibshirani (2009). *The elements of statistical learning*. Vol. 2. Springer.

He, J, D Zhang, and X Zhu (2012). "Adaptive pattern recognition of myoelectric signal towards practical multifunctional prosthesis control". In: *Intelligent Robotics and Applications*. Springer, pp. 518–525.

Höhne, J, D Bartz, MN Hebart, KR Müller, and B Blankertz (2016). "Analyzing neuroimaging data with subclasses: A shrinkage approach". In: *NeuroImage* 124, pp. 740–751.

Huang, Y, KB Englehart, B Hudgins, and AD Chan (2005). "A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses". In: *Biomedical Engineering, IEEE Transactions on* 52.11, pp. 1801–1811.

Hudgins, B, P Parker, and RN Scott (1993). "A new strategy for multifunction myoelectric control". In: *Biomedical Engineering, IEEE Transactions on* 40.1, pp. 82–94.

Hull, JJ (1994). "A database for handwritten text recognition research". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16.5, pp. 550–554.

Hwang, HJ, JM Hahne, and KR Müller (2014). "Channel selection for simultaneous and proportional myoelectric prosthesis control of multiple degrees-of-freedom". In: *Journal of neural engineering* 11.5, p. 056008.

Ioffe, S and C Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of The 32nd International Conference on Machine Learning*, pp. 448–456.

Jiang, N, S Dosen, KR Müller, and D Farina (2012). "Myoelectric control of artificial limbs – Is there a need to change focus". In: *IEEE Signal Processing Mag.* 29.5, pp. 149–152.

Jiang, N, I Vujaklija, H Rehbaum, B Graimann, and D Farina (2014). "Is accurate mapping of EMG signals on kinematics needed for precise online myoelectric control?" In: *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 22.3, pp. 549–558.

Jones, E, T Oliphant, P Peterson, et al. (2001–). *SciPy: Open source scientific tools for Python*.

Kalchbrenner, N, L Espeholt, K Simonyan, Avd Oord, A Graves, and K Kavukcuoglu (2016). "Neural machine translation in linear time". In: *arXiv preprint arXiv:1610.10099*.

Kim, Y (2014). "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882*.

Kindermans, PJ, KT Schütt, M Alber, KR Müller, and S Dähne (2017). "PatternNet and PatternLRP–Improving the interpretability of neural networks". In: *arXiv preprint arXiv:1705.05598*.

Kloft, M, U Brefeld, S Sonnenburg, and A Zien (2011). "lp-Norm Multiple Kernel Learning". In: *JMLR* 12, 953—997.

Kloft, M and G Blanchard (2011). "The Local Rademacher Complexity of Lp-Norm Multiple Kernel Learning". In: *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc., pp. 2438–2446.

Kloft, M and P Laskov (2010). "Online anomaly detection under adversarial impact". In: *AISTATS*, pp. 405–412.

Kloft, M, U Brefeld, S Sonnenburg, P Laskov, KR Müller, and A Zien (2009). "Efficient and accurate lp-norm multiple kernel learning". In: *Advances in neural information processing systems* 22.22, pp. 997–1005.

Krizhevsky, A, I Sutskever, and GE Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

Lapuschkin, S, A Binder, G Montavon, KR Müller, and W Samek (2016a). "Analyzing classifiers: Fisher vectors and deep neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2912–2920.

Lapuschkin, S, A Binder, G Montavon, KR Müller, and W Samek (2016b). "The LRP toolbox for artificial neural networks". In: *Journal of Machine Learning Research* 17.114, pp. 1–5.

LeCun, Y, B Boser, JS Denker, D Henderson, RE Howard, W Hubbard, and LD Jackel (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4, pp. 541–551.

LeCun, Y, O Matan, B Boser, JS Denker, D Henderson, R Howard, W Hubbard, L Jacket, and H Baird (1990). "Handwritten zip code recognition with multilayer networks". In: *Pattern Recognition, 1990. Proceedings., 10th International Conference on*. Vol. 2. IEEE, pp. 35–40.

LeCun, Y, L Bottou, Y Bengio, and P Haffner (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

LeCun, Y, L Bottou, GB Orr, and KR Müller (2012). "Efficient backprop". In: *Neural networks: Tricks of the trade*. Springer, pp. 9–48.

LeCun, Y, Y Bengio, and G Hinton (2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444.

Lemm, S, B Blankertz, T Dickhaus, and KR Müller (2011). "Introduction to machine learning for brain imaging". In: *Neuroimage* 56.2, pp. 387–399.

Lesk, A (2013). *Introduction to bioinformatics*. Oxford University Press.

Lesk, A (2017). *Introduction to genomics*. Oxford University Press.

Leslie, CS, E Eskin, and WS Noble (2002). "The Spectrum Kernel: A String Kernel for SVM Protein Classification." In: *Pacific Symposium on Biocomputing*, pp. 566–575.

Liu, B, L Fang, R Long, X Lan, and KC Chou (2016). "iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition". In: *Bioinformatics* 32.3, pp. 362–369.

Liu, DC and J Nocedal (1989). "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* 45.1, pp. 503–528.

Mathelier, A et al. (2015). "JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles". In: *Nucleic acids research*, gkv1176.

Mieth, B, M Kloft, JA Rodríguez, S Sonnenburg, R Vobruba, C Morcillo-Suárez, X Farré, UM Marigorta, E Fehr, T Dickhaus, et al. (2016). "Combining Multiple Hypothesis Testing with Machine Learning Increases the Statistical Power of Genome-wide Association Studies". In: *Scientific Reports* 6.

Montavon, G and KR Müller (2012). "Better Representations: Invariant, Disentangled and Reusable". In: *Neural Networks: Tricks of the Trade*. Springer, pp. 559–560.

Montavon, G, S Lapuschkin, A Binder, W Samek, and KR Müller (2017). "Explaining nonlinear classification decisions with deep taylor decomposition". In: *Pattern Recognition* 65, pp. 211–222.

Müller, KR, S Mika, G Rätsch, K Tsuda, and B . Schölkopf (2001). "An introduction to kernel-based learning algorithms". In: *IEEE transactions on neural networks* 12.2, pp. 181–201.

Müller, KR, CW Anderson, and GE Birch (2003). "Linear and nonlinear methods for brain-computer interfaces". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11.2, pp. 165–169.

Müller, KR, M Tangermann, G Dornhege, M Krauledat, G Curio, and B Blankertz (2008). "Machine learning for real-time single-trial EEG-analysis: from brain–computer interfacing to mental state monitoring". In: *Journal of neuroscience methods* 167.1, pp. 82–90.

Nakajima, S, A Binder, C Müller, W Wojcikiewicz, M Kloft, U Brefeld, KR Müller, and M Kawanabe (2009). "Multiple kernel learning for object classification". In: *Proceedings of the 12th Workshop on Information-based Induction Sciences*. Vol. 24.

Nasir, JA, N Görnitz, and U Brefeld (2014). "An Off-the-shelf Approach to Authorship Attribution." In: *COLING*, pp. 895–904.

Oord, Avd, S Dieleman, H Zen, K Simonyan, O Vinyals, A Graves, N Kalchbrenner, A Senior, and K Kavukcuoglu (2016). "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499*.

Porbadnigk, AK, N Görnitz, C Sannelli, A Binder, M Braun, M Kloft, and KR Müller (2015). "Extracting latent brain states — Towards true labels in cognitive neuroscience experiments". In: *NeuroImage* 120, pp. 225–253.

Rätsch, G and S Sonnenburg (2004). "13 Accurate Splice Site Detection for Caenorhabditis elegans". In: *Kernel methods in computational biology*, p. 277.

Rätsch, G, S Sonnenburg, J Srinivasan, H Witte, KR Müller, RJ Sommer, and B Schölkopf (2007). "Improving the Caenorhabditis elegans genome annotation using machine learning". In: *PLoS Computational Biology* 3.2, pp. 0313–0322.

Ribeiro, MT, S Sameer, and C Guestrin (2016). ""Why Should I Trust You?" Explaining the Predictions of Any Classifier". In: *ACM*, pp. 1135–1144.

Samek, W, A Binder, G Montavon, S Lapuschkin, and KR Müller (2017). "Evaluating the visualization of what a deep neural network has learned". In: *IEEE Transactions on Neural Networks and Learning Systems* 99, pp. 1–14.

Sandelin, A, A Höglund, B Lenhardd, and WW Wasserman (2003). "Integrated analysis of yeast regulatory sequences for biologically linked clusters of genes". In: *Functional & integrative genomics* 3.3, pp. 125–134.

Sandelin, A, W Alkema, P Engström, WW Wasserman, and B Lenhard (2004). "JASPAR: an open-access database for eukaryotic transcription factor binding profiles". In: *Nucleic Acids Research* 32.Database-Issue, pp. 91–94.

Scheme, E and K Englehart (2011). "Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use." In: *Journal of Rehabilitation Research & Development* 48.6, pp. 643–653.

Schölkopf, B, A Smola, and KR Müller (1998). "Nonlinear component analysis as a kernel eigenvalue problem". In: *Neural computation* 10.5, pp. 1299–1319.

Schütt, KT, F Arbabzadah, S Chmiela, KR Müller, and A Tkatchenko (2017). "Quantum-chemical insights from deep tensor neural networks". In: *Nature communications* 8, p. 13890.

Sensinger, JW, BA Lock, and TA Kuiken (2009). "Adaptive pattern recognition of myoelectric signals: exploration of conceptual framework and practical algorithms". In: *IEEE Trans. Neural Syst. Rehab. Eng.* 17.3, pp. 270–278.

Shawe-Taylor, J and N Cristianini (2004). *Kernel Methods for Pattern Analysis*. Cambridge.

Shenoy, P, M Krauledat, B Blankertz, RP Rao, and KR Müller (2006). "Towards adaptive classification for BCI". In: *Journal of neural engineering* 3.1, R13.

Shimodaira, H (2000). "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of statistical planning and inference* 90.2, pp. 227–244.

Simard, PY, D Steinkraus, JC Platt, et al. (2003). "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis." In: *ICDAR*. Vol. 3. Citeseer, pp. 958–962.

Sonnenburg, S, G Rätsch, C Schäfer, and B Schölkopf (2006a). "Large Scale Multiple Kernel Learning". In: *Journal of Machine Learning Research* 7. Ed. by K Bennett and EP Hernandez, pp. 1531–1565.

Sonnenburg, S, G Rätsch, S Henschel, C Widmer, J Behr, A Zien, FD Bona, A Binder, C Gehl, and V Franc (2010). "The SHOGUN Machine Learning Toolbox". In: *Journal of Machine Learning Research* 11, pp. 1799–1802.

Sonnenburg, S (2008). "Machine Learning for Genomic Sequence Analysis-Dissertation". PhD thesis. Berlin Institute of Technology.

Sonnenburg, S and V Franc (2010). "COFFIN: A Computational Framework for Linear SVMs". In: *ICML*, pp. 999–1006.

Sonnenburg, S, G Rätsch, A Jagota, and KR Müller (2002). "New methods for splice site recognition". In: *Artificial Neural Networks?ICANN 2002*. Springer, pp. 329–336.

Sonnenburg, S, A Zien, and G Rätsch (2006b). "ARTS: Accurate Recognition of Transcription Starts in Human". In: *Bioinformatics* 22.14, e472–480.

Sonnenburg, S, G Schweikert, P Philips, J Behr, and G Rätsch (2007). "Accurate splice site prediction using support vector machines". In: *BMC Bioinformatics* 8.Suppl 10, S7.

Sonnenburg, S, A Zien, P Philips, and G Rätsch (2008). "POIMs: Positional oligomer importance matrices - Understanding support vector machine-based signal detectors". In: *Bioinformatics* 24.13, pp. 6–14.

Steinwart, I and A Christmann (2008). *Support Vector Machines*. Springer.

Sugiyama, M, M Krauledat, and KR Müller (2007). "Covariate shift adaptation by importance weighted cross validation". In: *The Journal of Machine Learning Research* 8, pp. 985–1005.

Sugiyama, M and M Kawanabe (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

Sugiyama, M and KR Müller (2005). "Input-dependent estimation of generalization error under covariate shift". In: *Statistics and Decisions-International Journal Stochastic Methods and Models* 23.4, pp. 249–280.

Sugiyama, M, T Suzuki, S Nakajima, H Kashima, P Bünau, and M Kawanabe (2008). "Direct importance estimation for covariate shift adaptation". In: *Annals of the Institute of Statistical Mathematics* 60.4, pp. 699–746.

Sugiyama, M, M Yamada, and MC du Plessis (2013). "Learning under nonstationarity: covariate shift and class-balance change". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 5.6, pp. 465–477.

Tsuda, K, M Kawanabe, G Rätsch, S Sonnenburg, and KR Müller (2002). "A new discriminative kernel from probabilistic models". In: *Neural Computation* 14.10, pp. 2397–2414.

Vapnik, V (1995). *A Nature of Statistical Learning Theory*. Springer.

Vert, JP, K Tsuda, and B Schölkopf (2004). "A primer on kernel methods". In: *Kernel Methods in Computational Biology*, pp. 35–70.

Vidaurre, C, M Kawanabe, P Von Bünau, B Blankertz, and KR Müller (2011). "Toward unsupervised adaptation of LDA for brain–computer interfaces". In: *Biomedical Engineering, IEEE Transactions on* 58.3, pp. 587–597.

Vidovic, MMC, LP Paredes, HJ Hwang, S Amsüss, J Pahl, JM Hahne, B Graimann, D Farina, and KR Müller (2014). "Covariate shift adaptation in EMG pattern recognition for prosthetic device control". In: *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE, pp. 4370–4373.

Vidovic, MMC, N Görnitz, KR Müller, G Rätsch, and M Kloft (2015a). "Opening the Black Box: Revealing Interpretable Sequence Motifs in Kernel-Based Learning Algorithms". In: *ECML PKDD*. Vol. 6913, pp. 175–190.

Vidovic, MMC, N Görnitz, KR Müller, G Rätsch, and M Kloft (2015b). "SVM2Motif — Reconstructing Overlapping DNA Sequence Motifs by Mimicking an SVM Predictor". In: *PloS one* 10.12, e0144782.

Vidovic, MMC, N Görnitz, KR Müller, and M Kloft (2016a). "Feature Importance Measure for Non-linear Learning Algorithms". In: *arXiv preprint arXiv:1611.07567*.

Vidovic, MMC, HJ Hwang, S Amsüss, JM Hahne, D Farina, and KR Müller (2016b). "Improving the robustness of myoelectric pattern recognition for upper limb prostheses by covariate shift adaptation". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24.9, pp. 961–970.

Vidovic, MMC, M Kloft, KR Müller, and N Görnitz (2017). "ML2Motif—Reliable extraction of discriminative sequence motifs from learning machines". In: *PloS one* 12.3, e0174392.

Zeller, G, N Görnitz, A Kahles, J Behr, P Mudrakarta, S Sonnenburg, and G Rätsch (2013). "mTim: rapid and accurate transcript reconstruction from RNA-Seq data". In: *arXiv preprint arXiv:1309.5211*.

Zien, A, P Philips, and S Sonnenburg (2007). *Computing Positional Oligomer Importance Matrices (POIMs)*. Research Report; Electronic Publication. Fraunhofer Institute FIRST.

Zien, A, G Rätsch, S Mika, B Schölkopf, T Lengauer, and KR Müller (2000). "Engineering support vector machine kernels that recognize translation initiation sites". In: *Bioinformatics* 16.9, pp. 799–807.

Zien, A, N Krämer, S Sonnenburg, and G Rätsch (2009). "The feature importance ranking measure". In: *Machine Learning and Knowledge Discovery in Databases*, pp. 694–709.

Chapter A

Appendix

A.1 Biological Background

This section introduces the biological background of genes, where we explain the structure and function of genes until their final products, the proteins (Lesk, 2013, 2017).

The Anatomy of a Genome

The cell is the smallest functional unit of all biological life. Humans, for example, have about 10 trillion (10^{13}) cells. Every cell contains the entirety of an organism's heritable information, the so-called *genome*. The most important functions of this complex cell-unit are the following:

- It is able to replicate itself by cell division.
- It has a metabolism processing nutrient.
- It is able to build or manufacture proteins by protein synthesis.

The genome is stored in DNA molecules, which can be thought of long chains composed of linearly linked nucleotides. Nucleotides are chemical compounds, where a distinction is made between the four essentially nucleotides called A (adenine), C (cytosine), G (guanine), and T (thymine).
Due to the property that nucleotides are able to bind their complementary nucleotides (A with T as well as C with G), DNA forms a double helix structure as shown in Figure A.1. This redundancy has two advantages. On the one hand the cell is able to detect and repair so-called point mutations, i.e., where one nucleotide was been accidentally replaced by another one in a strand. On the other hand the redundancy enables DNA replication. During this process, the double helix is separated into two single strands of DNA, each of them serves as a template for synthesizing its complementary strand. The results are two identical DNA double helix structures. Due to the fact that a human genome consists of more than six billion nucleotides, the DNA double helix is separated into 46 pieces, where each of them is packed into a structure known as chromosome. Organisms are divided into the two groups eukaryotes and prokaryotes, depending on their cell types. Eukaryote cells have a nucleus containing the genome, whereas prokaryote cells are much simpler without any nucleus. In this thesis, we focus on the more complex eukaryotic organism.

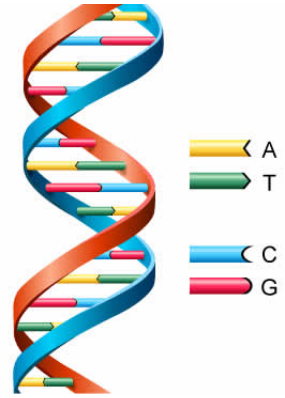


Fig. A.1: *Figurative illustration of a DNA double helix structure. It is composed of the four nucleotides A (adenine), C (cytosine), G (guanine), and T (thymine), where A binds with T and C with G. The figure was taken from* `http://education.technyou.edu.au/view/91/155/what-does-dna-look`

Genes and Proteins

Proteins are basically molecules with specific shapes and tasks. They consist of chains of amino acids. All known organisms use the same 20 basic amino acids. The specific composition of an amino acid chain determines shape and function of the protein. There are various functions of proteins and each cell needs to produce thousands of them to guarantee its functionality in the organism. They are responsible for most of the cellular actions, for example:

- Enzymes (a type of protein) catalyse chemical reactions, and control functions of the metabolism, DNA replication, and DNA reparation.
- Proteins enable communication between cells, recognize small signal molecules, and induce muscle contraction.

Cells can produce various proteins. The 'recipe' for each protein is given by a gene. The process during which cells produce proteins from genes is called protein biosynthesis. Genes are DNA subsequences, their lengths depend on the organism. Located a few nucleotides before the gen, there exists a promoter region, a small DNA subsequence like the TATA-box. The promoter region signalizes the start of the gene in the DNA. Figure A.2 shows the central dogma of the protein biosynthesis:

$$DNA \longrightarrow RNA \longrightarrow Protein$$

Particular proteins, called transcription factors detect the promoter region in the DNA and recruit an RNA polymerase, an enzyme that starts one of three steps of the protein synthesis called transcription step. First, the DNA double helix is unwound and RNA polymerase progresses along the template strand and synthesizes a complementary RNA strand called messenger RNA (mRNA) till it reaches a termination sequence. Unlike DNA, RNA has the nucleobase uracil (U) instead of thymine (T). The next step is known as splice step and is necessary because in eucaryotic organisms, genes

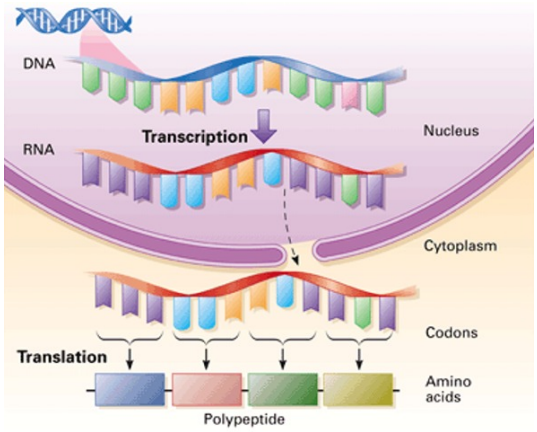


Fig. A.2: *Visualization of the protein biosynthesis process. First, a gene is recognized in the DNA and synthesized as RNA. After some processing steps the protein will be produced. Taken from http://terra.dadeschools.net/books/Biology/BiologyexploringLife04/0-13-115075-8/text/chapter11/concept11.4.html*

include coding regions (exons) and non-coding regions (introns), which are identified by splice site sequences. The introns are removed and the exons are joined together at the splice sites. For the third step, the spliced mRNA migrates from the nucleus to the cytoplasm. Right there, the ribosome, the cellular machine for synthesizing proteins, starts the synthesis step. Therefore, the ribosome docks at the start codon[1] and begins reading the codons one by one, translating them into the right amino acids and attaching them to the growing polypeptide (protein).The process is stopped if the ribosome reads one of three different stop codons. Finding the true stop codons, splice sites, or the transcription factors is a challenging learning problem.

A.2 Derivation of the Margin

For determining the margin, the hyperplane equations are written in the hesse normal form (HNF):

$$
\begin{aligned}
H \text{ in } HNF: \quad & \left\langle x, \tfrac{w}{\|w\|} \right\rangle = -\frac{b}{\|w\|} \\
H_1 \text{ in } HNF: \quad & \left\langle x_1, \tfrac{w}{\|w\|} \right\rangle = \frac{1-b}{\|w\|} \\
H_2 \text{ in } HNF: \quad & \left\langle x_2, \tfrac{w}{\|w\|} \right\rangle = \frac{-1-b}{\|w\|}
\end{aligned}
$$

[1] Codons are entities consisting of three nucleotides. The most common start codon is ATG.

The computation of the distance d_+ results in:

$$\begin{aligned} d_+ &= \left| \left\langle x_1, \frac{w}{\|w\|} \right\rangle + \frac{b}{\|w\|} \right| \\ &= \left| \frac{1-b}{\|w\|} + \frac{b}{\|w\|} \right| \\ &= \frac{1}{\|w\|} \end{aligned}$$

The analogue computation of d_- results in $\frac{1}{\|w\|}$, so that the margin is $\frac{2}{\|w\|}$.

A.3 Derivation of the Dual Problem

With the method of the Lagrange multipliers the constrained primal optimization problem (see (2.5)) can be handled by integrating the constraints through the Lagrange multipliers α, β with $\alpha_i \geq 0$ and $\beta_i \geq 0$ for $i = 1, \ldots, n$ in the objective function

$$L_p(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\langle w, \Phi(x_i) \rangle + b \right) - 1 + \xi_i \right) - \sum_{i=1}^{n} \beta_i \xi_i,$$

where L_p is known as the Lagrangian function (Boyd and Vandenberghe, 2004; Alt, 2011), and then solve

$$\min_{w,b,\xi,\alpha,\beta} L_p(w, b, \xi, \alpha, \beta) \ . \tag{A.1}$$

The optimality conditions for a potential solution are given by the so-called *Karush-Kuhn-Tucker* (KKT) conditions. Due to the fact that the optimization problem (2.5) is convex and the objective function is differentiable, the KKT-conditions are necessary and also sufficient for optimality and are given by the KKT-System (Boyd and Vandenberghe, 2004; Alt, 2011):

$$\begin{aligned} \frac{\partial L_p(w, b, \xi, \alpha, \beta)}{\partial w} &\stackrel{!}{=} 0 && (A.2) \\ \frac{\partial L_p(w, b, \xi, \alpha, \beta)}{\partial b} &\stackrel{!}{=} 0 && (A.3) \\ \frac{\partial L_p(w, b, \xi, \alpha, \beta)}{\partial \xi_i} &\stackrel{!}{=} 0 \quad i = 1, \ldots, n \\ y_i(\langle \Phi(x_i), w \rangle + b) &\geq 1 - \xi_i \quad i = 1, \ldots, n && (A.4) \\ \langle \alpha_i, -y_i(\langle \Phi(x_i), w \rangle + b) + 1 - \xi_i \rangle &\stackrel{!}{=} 0 \quad i = 1, \ldots, n && (A.5) \\ \langle \beta_i, -\xi_i \rangle &\stackrel{!}{=} 0 \quad i = 1, \ldots, n && (A.6) \\ \alpha_i &\geq 0 \quad i = 1, \ldots, n \\ \beta_i &\geq 0 \quad i = 1, \ldots, n \\ \xi_i &\geq 0 \quad i = 1, \ldots, n \, . \end{aligned}$$

Equation A.5 and A.6 are known as complementary slackness, where the first one plays an important role in SVM-theory: Only a few Lagrangian multiplier α_i are unequal zero namely, if and only if the mapping $\Phi(x_i)$ lies on or in the margin, we than say that x_i is a *supporting vector*. Substituting and converting the derivations

$$\frac{\partial L_p\left(w, b, \xi, \alpha, \beta\right)}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i)$$

and

$$\frac{\partial L_p\left(w, b, \xi, \alpha, \beta\right)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i$$

in (A.2) and (A.3) results in:

$$w = \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i) \tag{A.7}$$

and

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{A.8}$$

We obtain the dual optimization problem by substituting (A.7) and (A.8) into (2.5):

$$\begin{aligned}
&\frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i \left(\langle w, x_i\rangle + b\right) - 1\right) + \sum_{i=1}^{N} \alpha_i \\
&= \frac{1}{2}\left\|\sum_{i=1}^{N} \alpha_i y_i x_i\right\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i \left(\left\langle \sum_{j=1}^{N} \alpha_j y_j x_j, x_i \right\rangle + b\right) - 1\right) + \sum_{i=1}^{N} \alpha_i \\
&= -\sum_{i=1}^{N} \alpha_i y_i \left\langle \sum_{i=j}^{N} \alpha_j y_j x_j, x_i \right\rangle - \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \alpha_i \\
&= -\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\langle x_i, x_j\rangle\right) + 2\sum_{i=1}^{N} \alpha_i.
\end{aligned}$$

A.4 Extension of Theorem 12 and 13 to Multiple Motifs

Theorem 17 *Suppose uppose that the objective function* Π *of the following optimization problem*

$$\begin{aligned}
\min_{r} \quad & \Pi((m_{k,t})_{t=1,\dots,T_k, k\in\mathcal{K}}) = \frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{y\in\Sigma^{\tilde{k}}}\sum_{j=1}^{L-\tilde{k}+1}\left(\sum_{t=1}^{T_k}\left(R_{y,j}(m_{k,t}) - S_{\tilde{k},y,j} + c\right)\right)^2 \\
s.t. \quad & 0 \le r_{k,t,o,s} \le 1 \qquad t = 1,\dots,T_k,\ k \in \mathcal{K}, o = 1,\dots,4, s = 1,\dots,k\,, \\
& \sum_{o} r_{k,t,o,s} = 1 \quad t = 1,\dots,T_k,\ k \in \mathcal{K}, s = 1,\dots,k\,,
\end{aligned}$$

is convex and let r_c^* be the optimal solution, then $\forall c' \in \mathbb{R}\ r_{c'}^* = r_c^*$.

Proof 5 *Let r_c^* be the optimal solution of the objective function Π (A.9) with the inequality constraints $h_{k,t,o,s,1} = -r_{k,t,o,s}$ and $h_{k,t,o,s,2} = r_{k,t,o,s} - 1,\ k \in \mathcal{K}, t = 1,\dots,T_k, o = 1,\dots,4, s = 1,\dots,k, i = 1,2$ and the equality constraints $g_k, t, s = \sum_o r_{o,s} - 1,\ k \in \mathcal{K}, t = 1,\dots,T_k, s = 1,\dots,k$, and let η and ξ be the Lagrangian multipliers, then the Lagrangian function is as follows*

$$\mathcal{L}(r,\eta,\xi) = \Pi(r_c^*;\mu) + \sum_{k\in\mathcal{K}}\sum_{t=1}^{T_k}\sum_{o=1}^{4}\sum_{s=1}^{k}\sum_{i=1}^{2}\eta_{k,t,o,s,i}h_{k,t,o,s,i} + \sum_{k\in\mathcal{K}}\sum_{t=1}^{T_k}\sum_{s=1}^{k}\xi_{k,t,s}g_{k,t,s}\,.$$

The Karush-Kuhn-Tucker(KKT) conditions are satisfied for r_c^*: The primal feasibility conditions ($g_{k,t,s} = 0,\ \mathcal{K}, t = 1,\dots,T_k, s = 1,\dots,k$ and $h_{k,t,o,s,i} \le 0,\ \mathcal{K}, t = 1,\dots,T_k, o = 1,\dots,4, s = 1,\dots,k, i = 1,2$) are trivially fulfilled, since r_c^* is a stochastic matrix. Together with the dual feasibility conditions ($\eta \ge 0$) the complementary slackness condition ($\eta_{k,t,o,s,i}h_{k,t,o,s,i} = 0,\ \mathcal{K}, t = 1,\dots,T_k, o = 1,\dots,4, s = 1,\dots,k, i = 1,2$) are trivially fulfilled as well, which leaves us to show that the stationarity condition

$$\nabla\Pi(r_c^*;\mu) + \sum_{k\in\mathcal{K}}\sum_{t=1}^{T_k}\sum_{i=1}^{2}\sum_{o}\sum_{s=1}^{k}\eta_{k,t,o,s,1}\nabla h_{k,t,o,s,i} + \sum_{k\in\mathcal{K}}\sum_{t=1}^{T_k}\sum_{s}^{k}\xi_{k,t,s}\nabla g_{k,t,s} = 0$$

is satisfied. Therefore we insert the derivations and reorganize for the Lagrange multipliers ξ , which leads to

$$\begin{aligned}\xi_{k,t,s} \;=\; -\sum_{k\in\mathcal{K}}\sum_{y}\sum_{j}1_{\{i\in\mathcal{U}(\mu)\}}\Bigg(&\sum_{t=1}^{T_k}\prod_{l=1}^{\tilde{k}}r^*_{c,k,t,y_l,j+l}\prod_{\substack{l=1\\l\neq t}}^{\tilde{k}}r^*_{c,k,t,y_l,j+l}\\ &-(S_{\tilde{k},y,j+\mu} - c)\prod_{\substack{l=1\\l\neq t}}^{\tilde{k}}r^*_{c,k,t,y_l,j+l}\Bigg) + \sum_{k\in\mathcal{K}}\sum_{t=1}^{T_k}\sum_{i=1}^{2}\eta_{k,t,o,s,i}.\end{aligned}$$

With $\xi \in \mathbb{R}$ it holds, that for any $c' \in \mathbb{R}\ r_{c'}^* = r_c^*$. The fact that Π is convex, h is convex and g is affine denotes the KKT conditions as sufficient and concludes the proof.

Theorem 18 (Convexity for multiple motifs) *Let D be a convex set, $m_k \in D$ a probabilistic motif, S a gPOIM, such that $S_{\tilde{k},y,j} \in \mathbb{R}$ for $y \in \Sigma^{\tilde{k}}$ and $j = 1,\dots,L-\tilde{k}+1$, $\mu \in [1, L-k+1]$, $c \in \mathbb{R}$ and $S_\lfloor$ the element wise minimum of S then, if $c \ge \mathbb{1}_{\{S_\lfloor<0\}}S_\lfloor + \mathbb{1}_{\{S_\lfloor<T_k\}}T_k$ it holds that*

$$\Pi((m_{k,t})_{t=1,\dots,T_k,k\in\mathcal{K}}) \;=\; \frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{y\in\Sigma^{\tilde{k}}}\sum_{j=1}^{L-\tilde{k}+1}\left(\sum_{t=1}^{T_k}R_{y,j}(m_{k,t}) - (S_{\tilde{k},y,j} + c)\right)^2$$

is convex.

Proof 6 *We have to proof the following inequality to show convexity of* $f(m_k)$

$$||\sum_{t=1}^{T_k} R(\Phi r_{k,t} + (1-\Phi)s_{k,t};\mu) - (S + c')||_2^2 \leq \Phi||\sum_{t=1}^{T_k} R(r_{k,t};\mu) - (S + c')||_2^2 + (1-\Phi)||\sum_{t=1}^{T_k} R(s_{k,t};\mu) - (S + c')||_2^2$$

which is, for the case $j \notin \mathbb{1}_{\{i \in \mathcal{U}(\mu)\}}$*, trivially fulfilled for* $c' \in \mathbb{R}$*. This, due to the fact, that a sum of convex functions is convex, leaves us with showing the following inequality*

$$\big(\sum_{t=1}^{T_k} \Phi a_t + (1-\Phi)b_t - (S_{\tilde{k},y,j} + c')\big)^2 \leq \Phi\big(\sum_{t=1}^{T_k} a_t - (S_{\tilde{k},y,j} + c')\big)^2 + (1-\Phi)\big(\sum_{t=1}^{T_k} b_t - (S_{\tilde{k},y,j} + c')\big)^2, \tag{A.9}$$

where we replaced the PWM products $\prod_{l=j}^{k+j} r_{k,t,y_l,l}$ *and* $\prod_{l=j}^{k+j} s_{k,t,y_l,l}$ *by* a_t *and* b_t *for more transparency. After resolving and transforming Eq. (A.9) shortens to*

$$\Phi^2 \sum_{t=1}^{T_k} a_t^2 + 2\Phi \sum_{t=1}^{T_k} a_t b_t - 2\Phi^2 \sum_{t=1}^{T_k} a_t b_t \leq \Phi \sum_{t=1}^{T_k} a_t^2 + 2\Phi(S_{\tilde{k},y,j} + c')^2. \tag{A.10}$$

Since $-2\Phi^2 \sum_{t=1}^{T_k} a_t b_t \leq 0$ *and* $\Phi^2 \sum_{t=1}^{T_k} a_t^2 \leq \Phi \sum_{t=1}^{T_k} a_t^2$*, Eq. (A.10) reduces to* $\sum_{t=1}^{T_k} a_t b_t \leq (S_{\tilde{k},y,j} + c')^2$*. The fact that the maximum of* $\sum_{t=1}^{T_k} a_t b_t$ *is* T_k*, concludes the proof for* $c \geq c'$ *with* $c' = \mathbb{1}_{\{\min(S)<0\}} S_{\lfloor} + \mathbb{1}_{\{S_{\lfloor} < T_k\}} T_k$*.*