The Feature Importance Ranking Measure

Alexander Zien[1,2], Nicole Krämer[3],
Sören Sonnenburg[2], and Gunnar Rätsch[2]

[1] Fraunhofer FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany
[2] Friedrich Miescher Laboratory, Max Planck Society,
Spemannstr. 39, 72076 Tübingen, Germany
[3] Machine Learning Group, Berlin Institute of Technology,
Franklinstr. 28/29, 10587 Berlin, Germany

Abstract. Most accurate predictions are typically obtained by learning machines with complex feature spaces (as e.g. induced by kernels). Unfortunately, such decision rules are hardly accessible to humans and cannot easily be used to gain insights about the application domain. Therefore, one often resorts to linear models in combination with variable selection, thereby sacrificing some predictive power for presumptive interpretability. Here, we introduce the *Feature Importance Ranking Measure* (FIRM), which by retrospective analysis of arbitrary learning machines allows to achieve both excellent predictive performance and superior interpretation. In contrast to standard raw feature weighting, FIRM takes the underlying correlation structure of the features into account. Thereby, it is able to discover the most relevant features, even if their appearance in the training data is entirely prevented by noise. The desirable properties of FIRM are investigated analytically and illustrated in simulations.

1   Introduction

A major goal of machine learning — beyond providing accurate predictions — is to gain understanding of the investigated problem. In particular, for researchers in application areas, it is frequently of high interest to unveil which features are indicative of certain predictions. Existing approaches to the identification of important features can be categorized according to the restrictions that they impose on the learning machines.

The most convenient access to features is granted by linear learning machines. In this work we consider methods that express their predictions via a real-valued output function $s : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the space of inputs. This includes standard models for classification, regression, and ranking. Linearity thus amounts to

$$s(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b \ . \tag{1}$$

One popular approach to finding important dimensions of vectorial inputs ($\mathcal{X} = \mathbb{R}^d$) is *feature selection*, by which the training process is tuned to make sparse use of the available d candidate features. Examples include ℓ_1-regularized methods like Lasso [13] or ℓ_1-SVMs [1] and heuristics for non-convex ℓ_0-regularized formulations. They all find feature weightings $\mathbf{w}$ that have few non-zero components,

for example by eliminating redundant dimensions. Thus, although the resulting predictors are economical in the sense of requiring few measurements, it can not be concluded that the other dimensions are unimportant: a different (possibly even disjoint) subset of features may yield the same predictive accuracy. Being selective among correlated features also predisposes feature selection methods to be unstable. Last but not least, the accuracy of a predictor is often decreased by enforcing sparsity (see e.g. [10]).

In multiple kernel learning (MKL; e.g. [5,10]) a sparse linear combination of a small set of kernels [8] is optimized concomitantly to training the kernel machine. In essence, this lifts both merits and detriments of the selection of individual features to the coarser level of feature spaces (as induced by the kernels). MKL thus fails to provide a principled solution to assessing the importance of sets of features, not to speak of individual features. It is now urban knowledge that ℓ_1-regularized MKL can even rarely sustain the accuracy of a plain uniform kernel combination [2].

Alternatively, the sparsity requirement may be dropped, and the j-th component w_j of the trained weights $\mathbf{w}$ may be taken as the importance of the j-th input dimension. This has been done, for instance, in cognitive sciences to understand the differences in human perception of pictures showing male and female faces [4]; here the resulting weight vector $\mathbf{w}$ is relatively easy to understand for humans since it can be represented as an image.

Again, this approach may be partially extended to kernel machines [8], which do not access the features explicitly. Instead, they yield a kernel expansion

$$s(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \ , \tag{2}$$

where $(\mathbf{x}_i)_{i=1,...,n}$ are the inputs of the n training examples. Thus, the weighting $\boldsymbol{\alpha} \in \mathbb{R}^n$ corresponds to the training examples and cannot be used directly for the interpretation of features. It may still be viable to compute explicit weights for the features $\Phi(\mathbf{x})$ induced by the kernel via $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$, provided that the kernel is benign: it must be guaranteed that only a finite and limited number of features are used by the trained machine, such that the equivalent linear formulation with

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i)$$

can efficiently be deduced and represented.

A generalization of the feature weighting approach that works with general kernels has been proposed by Üstün et. al. [14]. The idea is to characterize input variables by their correlation with the weight vector $\boldsymbol{\alpha}$. For a linear machine as given by (1) this directly results in the weight vector $\mathbf{w}$; for non-linear functions s, it yields a projection of $\mathbf{w}$, the meaning of which is less clear.

A problem that all above methods share is that the weight that a feature is assigned by a learning machine is not necessarily an appropriate measure of

its importance. For example, by multiplying any dimension of the inputs by a positive scalar and dividing the associated weight by the same scalar, the conjectured importance of the corresponding feature can be changed arbitrarily, although the predictions are not altered at all, i.e. the trained learning machine is unchanged. An even more practically detrimental shortcoming of the feature weighting is its failure to take into account correlations between features; this will be illustrated in a computational experiment below (Section 3).

Further, all methods discussed so far are restricted to linear scoring functions or kernel expansions. There also exists a range of customized importance measures that are used for building decision trees and random forests (see e.g. [11,12] for an overview).

In this paper, we reach for an importance measure that is "universal": it shall be applicable to any learning machine, so that we can avoid the clumsiness of assessing the relevance of features for methods that produce suboptimal predictions, and it shall work for any feature. We further demand that the importance measure be "objective", which has several aspects: it may not arbitrarily choose from correlated features as feature selection does, and it may not be prone to misguidance by feature rescaling as the weighting-based methods are. Finally, the importance measure shall be "intelligent" in that it exploits the connections between related features (this will become clearer below).

In the next section, we briefly review the state of the art with respect to these goals and in particular outline a recent proposal, which is, however, restricted to sequence data. Section 2 exhibits how we generalize that idea to continuous features and exhibits its desirable properties. The next two sections are devoted to unfolding the math for several scenarios. Finally, we present a few computational results illustrating the properties of our approach in the different settings. The relevant notation is summarized in Table 1.

Table 1. Notation

symbol	definition	reference
$\mathcal{X}$	input space	
$s(x)$	scoring function $\mathcal{X} \to \mathbb{R}$	
w	weight vector of a linear scoring function s	equation (1)
f	feature function $\mathcal{X} \to \mathbb{R}$	equation (6)
$q_f(t)$	conditional expected score $\mathbb{R} \to \mathbb{R}$	definition 1
Q_f	feature importance ranking measure (firm) $\in \mathbb{R}$	definition 2
$\mathbf{Q}$	vector $\in \mathbb{R}^d$ of firms for d features	subsection 2.4
$\mathbf{\Sigma}, \mathbf{\Sigma}_{j\bullet}$	covariance matrix, and its jth column	

1.1   Related Work

A few existing feature importance measures satisfy one or more of the above criteria. One popular "objective" approach is to assess the importance of a variable by measuring the decrease of accuracy when retraining the model based on

a random permutation of a variable. However, it has only a narrow application range, as it is computationally expensive and confined to input variables.

Another approach is to measure the importance of a feature in terms of a sensitivity analysis [3]

$$I_j = \mathbb{E}\left[\left(\frac{\partial s}{\partial x_j}\right)^2 \mathbb{V}\mathrm{ar}\,[X_j]\right]^{1/2} . \tag{3}$$

This is both "universal" and "objective". However, it clearly does not take the indirect effects into account: for example, the change of X_j may imply a change of some X_k (e.g. due to correlation), which may also impact s and thereby augment or diminish the net effect.

Here we follow the related but more "intelligent" idea of [17]: to assess the importance of a feature by estimating its total impact on the score of a trained predictor. While [17] proposes this for binary features that arise in the context of sequence analysis, the purpose of this paper is to generalize it to real-valued features and to theoretically investigate some properties of this approach. It turns out (proof in Section 2.2) that under normality assumptions of the input features, FIRM generalizes (3), as the latter is a first order approximation of FIRM, and because FIRM also takes the correlation structure into account.

In contrast to the above mentioned approaches, the proposed *feature importance ranking measure* (FIRM) also takes the dependency of the input features into account. Thereby it is even possible to assess the importance of features that are not observed in the training data, or of features that are not directly considered by the learning machine.

1.2 Positional Oligomer Importance Matrices [17]

In [17], a novel feature importance measure called Positional Oligomer Importance Matrices (POIMs) is proposed for substring features in string classification. Given an alphabet Σ, for example the DNA nucleotides $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, let $\mathbf{x} \in \Sigma^L$ be a sequence of length L. The kernels considered in [17] induce a feature space that consists of one binary dimension for each possible substring $\mathbf{y}$ (up to a given maximum length) at each possible position i. The corresponding weight $w_{\mathbf{y},i}$ is added to the score if the substring $\mathbf{y}$ is incident at position i in $\mathbf{x}$. Thus we have the case of a kernel expansion that can be unfolded into a linear scoring system:

$$s(\mathbf{x}) = \sum_{\mathbf{y},i} w_{\mathbf{y},i}\mathbb{I}\left\{\mathbf{x}[i] = \mathbf{y}\right\} , \tag{4}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. Now POIMs are defined by

$$Q'(\mathbf{z}, j) := \mathbb{E}\left[\,s(\mathbf{X}) \mid \mathbf{X}[j] = \mathbf{z}\right] - \mathbb{E}\left[s(\mathbf{X})\right] , \tag{5}$$

where the expectations are taken with respect to a D-th order Markov distribution.

Intuitively, Q' measures how a feature, here the incidence of substring $\mathbf{z}$ at position j, would change the score s as compared to the average case (the unconditional expectation). Although positional sub-sequence incidences are binary features (they are either present or not), they posses a very particular correlation structure, which can dramatically aid in the identification of relevant features.

2   The Feature Importance Ranking Measure (FIRM)

As explained in the introduction, a trained learner is defined by its output or scoring function $s : \mathcal{X} \to \mathbb{R}$. The goal is to quantify how important any given feature

$$f : \mathcal{X} \to \mathbb{R} \tag{6}$$

of the input data is to the score. In the case of vectorial inputs $\mathcal{X} = \mathbb{R}^d$, examples for features are simple coordinate projections $f_j(\mathbf{x}) = x_j$, pairs $f_{jk}(\mathbf{x}) = x_j x_k$ or higher order interaction features, or step functions $f_{j,\tau}(\mathbf{x}) = \mathbb{I}\{x_j > \tau\}$ (where $\mathbb{I}\{\cdot\}$ is the indicator function).

We proceed in two steps. First, we define the expected output of the score function under the condition that the feature f attains a certain value.

Definition 1 (conditional expected score). *The conditional expected score of s for a feature f is the expected score $q_f : \mathbb{R} \to \mathbb{R}$ conditional to the feature value t of the feature f:*

$$q_f(t) = \mathbb{E}\left[\, s(X) \,\middle|\, f(X) = t \,\right]\ . \tag{7}$$

We remark that this definition corresponds — up to normalization — to the marginal variable importance studied by van der Laan [15]. A flat function q_f corresponds to a feature f that has no or just random effect on the score; a variable function q_f indicates an important feature f.

Consequently, the second step of FIRM is to determine the importance of a feature f as the variability of the corresponding expected score $q_f : \mathbb{R} \to \mathbb{R}$.

Definition 2 (feature importance ranking measure). *The feature importance $Q_f \in \mathbb{R}$ of the feature f is the standard deviation of the function q_f:*

$$Q_f := \sqrt{\mathbb{V}ar[q_f(f(X))]} = \left(\int_{\mathbb{R}} \big(q_f(t) - \bar{q}_f\big)^2 \mathbb{P}r\left(f(X) = t\right) dt\right)^{\frac{1}{2}}\ , \tag{8}$$

where $\bar{q}_f := \mathbb{E}\left[q_f(f(X))\right] = \int_{\mathbb{R}} q_f(t)\mathbb{P}r\left(f(X) = t\right) dt$ is the expectation of q_f.

In case of (i) known linear dependence of the score on the feature under investigation or (ii) an ill-posed estimation problem (8) — for instance, due to scarce data —, we suggest to replace the standard deviation by the more reliably estimated slope of a linear regression. As we will show later (Section 2.3), for binary features identical feature importances are obtained by both ways anyway.

2.1   Properties of FIRM

FIRM generalizes POIMs. As we will show in Section Section 2.3, FIRM indeed contains POIMs as special case. POIMs, as defined in (5), are only meaningful for binary features. FIRM extends the core idea of POIMs to continuous features.

FIRM is "universal". Note that our feature importance ranking measure (FIRM) can be applied to a very broad family of learning machines. For instance, it works in both classification, regression and ranking settings, as long as the task is modeled via a real-valued output function over the data points. Further, it is not constrained to linear functions, as is the case for l_1-based feature selection. FIRM can be used with any feature space, be it induced by a kernel or not. The importance computation is not even confined to features that are used in the output function. For example, one may train a kernel machine with a polynomial kernel of some degree and afterwards determine the importance of polynomial features of higher degree. We illustrate the ability of FIRM to quantify the importance of unobserved features in Section 3.3.

FIRM is robust and "objective". In order to be sensible, an importance measure is required to be robust with respect to perturbations of the problem and invariant with respect to irrelevant transformations. Many successful methods for classification and regression are translation-invariant; FIRM will immediately inherit this property. Below we show that FIRM is also invariant to rescaling of the features in some analytically tractable cases (including all binary features), suggesting that FIRM is generally well-behaved in this respect. In Section 2.4 we show that FIRM is even robust with respect to the choice of the learning method. FIRM *is* sensitive to rescaling of the scoring function s. In order to compare different learning machines with respect to FIRM, s should be standardized to unit variance; this yields importances $\tilde{Q}_f = Q_f / \mathbb{V}\mathrm{ar}\left[s(X)\right]^{1/2}$ that are to scale. Note, however, that the relative importance, and thus the ranking, of all features for any single predictor remains fixed.

Computation of FIRM. It follows from the definition of FIRM that we need to assess the distribution of the input features and that we have to compute conditional distributions of nonlinear transformations (in terms of the score function s). In general, this is infeasible. While in principle one could try to estimate all quantities empirically, this leads to an estimation problem due to the limited amount of data. However, in two scenarios, this becomes feasible. First, one can impose additional assumptions. As we show below, for normally distributed inputs and linear features, FIRM can be approximated analytically, and we only need the covariance structure of the inputs. Furthermore, for linear scoring functions (1), we can compute FIRM for (a) normally distributed inputs (b) binary data with known covariance structure and (c) — as shown before in [16] — for sequence data with (higher-order) Markov distribution. Second, one can approximate the conditional expected score q_f by a linear function, and to then estimate the feature importance Q_f from its slope. As we show in Section 2.3, this approximation is exact for binary data.

2.2   Approximate FIRM for Normally Distributed Features

For general score functions s and arbitrary distributions of the input, the computation of the conditional expected score (7) and the FIRM score (8) is in general intractable, and the quantities can at best be estimated from the data. However, under the assumption of normally distributed features, we can derive an analytical approximation of FIRM in terms of first order Taylor approximations. More precisely, we use the following approximation.

Approximation. *For a normally random variable $\widetilde{X} \sim \mathcal{N}\left(\widetilde{\mu}, \widetilde{\Sigma}\right)$ and a differentiable function $g : \mathbb{R}^d \to \mathbb{R}^p$, the distribution of $g(X)$ is approximated by its first order Taylor expansion:*

$$g(X) \sim \mathcal{N}\left(g(\widetilde{\mu}), J\widetilde{\Sigma}J^\top\right)$$

with

$$J = \left.\frac{\partial g}{\partial \mathbf{x}}\right|_{\mathbf{x}=\widetilde{\mu}}$$

Note that if the function g is linear, the distribution is exact.

In the course of this subsection, we consider feature functions $f_j(\mathbf{x}) = \mathbf{x}_j$ (an extension to linear feature functions $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{a}$ is straightforward.)

First, recall that for a normally distributed random variable $X \sim \mathcal{N}(\mathbf{0}, \Sigma)$, the conditional distribution of $X|X_j = t$ is again normal, with expectation

$$\mathbb{E}\left[\, X \mid X_j = t\,\right] = \frac{t}{\Sigma_{jj}}\Sigma_{j\bullet} =: \widetilde{\mu}_j \,.$$

Here $\Sigma_{j\bullet}$ is the jth column of Σ.

Now, using the above approximation, the conditional expected score is

$$q_f(t) \approx s\left(\widetilde{\mu}_j\right) = s((t/\Sigma_{jj})\,\Sigma_{j\bullet})$$

To obtain the FIRM score, we apply the approximation again, this time to the function $t \mapsto s(((t/\Sigma_{jj})\,\Sigma_{j\bullet})$. Its first derivative at the expected value $t = 0$ equals

$$J = \frac{1}{\Sigma_{jj}}\Sigma_{j\bullet}^\top \left.\frac{\partial s}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{0}}$$

This yields

$$Q_j \approx \sqrt{\frac{1}{\Sigma_{jj}}\left(\Sigma_{j\bullet}^\top \left.\frac{\partial s}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{0}}\right)^2} \tag{9}$$

Note the correspondence to (3) in Friedman's paper [3]: If the features are uncorrelated, (9) simplifies to

$$Q_j \approx \sqrt{\Sigma_{jj}\left(\left.\frac{\partial s}{\partial \mathbf{x}_j}\right|_{\mathbf{x}_j=0}\right)^2}$$

(recall that $0 = E[X_j]$). Hence FIRM adds an additional weighting that corresponds to the dependence of the input features. These weightings are based on the true covariance structure of the predictors. In applications, the true covariance matrix is in general not known. However, it is possible to estimate it reliably even from high-dimensional data using mean-squared-error optimal shrinkage [7].

Note that the above approximation can be used to compute FIRM for the kernel based score functions (2). E.g., for Gaussian kernels

$$k_\gamma(\mathbf{x}, \mathbf{x}_i) = exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\gamma^2}\right)$$

we have

$$\left.\frac{\partial k_\gamma(\mathbf{x}, \mathbf{x}_i)}{\partial \mathbf{x}}\right|_{\mathbf{x}=0} = \frac{2k(\mathbf{0}, \mathbf{x}_i)}{\gamma^2}\mathbf{x}_i^\top = \frac{2e^{-(\|\mathbf{x}_i\|^2/\gamma^2)}}{\gamma^2}\mathbf{x}_i^\top$$

and hence obtain

$$\left.\frac{\partial s}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{0}} = \sum_{i=1}^{N} \alpha_i y_i \frac{2e^{-(\|\mathbf{x}_i\|^2/\gamma^2)}}{\gamma^2}\mathbf{x}_i^\top \quad .$$

2.3   Exact FIRM for Binary Data

Binary features are both analytically simple and, due to their interpretability and versatility, practically highly relevant. Many discrete features can be adequately represented by binary features, even if they can assume more than two values. For example, a categorical feature can be cast into a sparse binary encoding with one indicator bit for each value; an ordinal feature can be encoded by bits that indicate whether the value is strictly less than each of its possibilities. Therefore we now try to understand in more depth how FIRM acts on binary variables.

For a binary feature $f : \mathcal{X} \to \{a, b\}$ with feature values $t \in \{a, b\}$, let the distribution be described by

$$p_a = \mathbb{Pr}\left(f(X) = a\right), \, p_b = 1 - p_a \,,$$

and let the conditional expectations be $q_a = q_f(a)$ and $q_b = q_f(b)$. Simple algebra shows that in this case $\mathbb{Var}\left[q(f(X))\right] = p_a p_b (q_a - q_b)^2$. Thus we obtain the feature importance

$$Q_f = (q_a - q_b)\sqrt{p_a p_b} \,. \tag{10}$$

(By dropping the absolute value around $q_a - q_b$ we retain the directionality of the feature's impact on the score.) Note that we can interpret firm in terms of the slope of a linear function. If we assume that $a, b \in \mathbb{R}$, the linear regression fit

$$(w_f, c_f) = \arg\min_{w_f, c_f} \int_{\mathbb{R}} \left((w_f t + c_f) - q_f(t)\right)^2 d\mathbb{Pr}\left(t\right)$$

the slope is $w_f = \frac{q_a - q_b}{a - b}$. The variance of the feature value is $\mathbb{V}\mathrm{ar}\,[f(X)] = p_a p_b (a - b)^2$. (10) is recovered as the increase of the linear regression function along one standard deviation of feature value. As desired, the importance is independent of feature translation and rescaling (provided that the score remains unchanged). In the following we can thus (without loss of generality) constrain that $t \in \{-1, +1\}$.

Let us reconsider POIMS Q', which are defined in equation (5). We note that $Q'(b) := q_b - \bar{q} = p_a(q_b - q_a) = \sqrt{p_a/p_b}\,Q(b)$; thus $Q(\mathbf{z}, j)$ can be recovered as

$$Q(\mathbf{z}, j) = Q'(\mathbf{z}, j)\sqrt{\Pr\left(\mathbf{X}[j] \neq \mathbf{z}\right)/\Pr\left(\mathbf{X}[j] = \mathbf{z}\right)}.$$

Thus, while POIMs are not strictly a special case of FIRM, they differ only in a scaling factor which depends on the distribution assumption. For a uniform Markov model (as empirically is sufficient according to [17]), this factor is constant.

2.4   FIRM for Linear Scoring Functions

To understand the properties of the proposed measure, it is useful to consider it in the case of linear output functions (1).

Independently Distributed Binary Data. First, let us again consider the simplest scenario of uniform binary inputs, $X \sim unif(\{-1, +1\}^d)$; the inputs are thus pairwise independent.

First we evaluate the importance of the input variables as features, i.e. we consider projections $f_j(\mathbf{x}) = x_j$. In this case, we immediately find for the conditional expectation $q_j(t)$ of the value t of the j-th variable that $q_j(t) = tw_j + b$. Plugged into (10) this yields $Q_j = w_j$, as expected. When the features are independent, their impact on the score is completely quantified by their associated weights; no side effects have to be taken into account, as no other features are affected.

We can also compute the importances of conjunctions of two variables, i.e.

$$f_{j \wedge k}(\mathbf{x}) = \mathbb{I}\left\{x_j = +1 \wedge x_k = +1\right\}.$$

Here we find that $q_{j \wedge k}(1) = w_j + w_k + b$ and $q_{j \wedge k}(0) = -\frac{1}{3}(w_j + w_k) + b$, with $\Pr\left(f_{j \wedge k}(X) = 1\right) = \frac{1}{4}$. This results in the feature importance $Q_{j \wedge k} = (w_j + w_k)/\sqrt{3}$. This calculation also applies to negated variables and is easily extended to higher order conjunctions.

Another interesting type of feature derives from the xor-function. For features $f_{j \otimes k}(\mathbf{x}) = \mathbb{I}\left\{x_j \neq x_k\right\}$ the conditional expectations vanish, $q_{j \otimes k}(1) = q_{j \otimes k}(0) = 0$. Here the FIRM exposes the inability of the linear model to capture such a dependence.

Binary Data With Empirical Distribution. Here we consider the empirical distribution as given by a set $\{\,\mathbf{x}_i \mid i = 1, \dots, n\,\}$ of n data points $\mathbf{x}_i \in \{-1, +1\}^d$: $\Pr(X) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{X = \mathbf{x}_i\}$. For input features $f_j(\mathbf{x}) = x_j$, this leads to $q_j(t) =$

$\frac{1}{n_{jt}} \sum_{i:\mathbf{x}_{ij}=t} \mathbf{w}^\top \mathbf{x}_i + b$, where $n_{jt} := |\{ i \mid \mathbf{x}_{ij} = t \}|$ counts the examples showing the feature value t. With (10) we get

$$Q_j = (q_j(+1) - q_j(-1)) \sqrt{\Pr(X_j = +1)\Pr(X_j = -1)}$$

$$= \sum_{i=1}^{n} \frac{\mathbf{x}_{ij}}{n_{j,\mathbf{x}_{ij}}} (\mathbf{w}^\top \mathbf{x}_i) \sqrt{\frac{n_{j,+1}n_{j,-1}}{n^2}}$$

It is convenient to express the vector $\mathbf{Q} \in \mathbb{R}^d$ of all feature importances in matrix notation. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the data matrix with the data points $\mathbf{x}_i$ as rows. Then we can write

$$\mathbf{Q} = \mathbf{M}^\top \mathbf{X}\mathbf{w} \ \text{ with } \ \mathbf{M} \in \mathbb{R}^{n \times d} = \mathbf{1}_{n \times d}\mathbf{D}_0 + \mathbf{X}\mathbf{D}_1$$

with diagonal matrices $\mathbf{D}_0, \mathbf{D}_1 \in \mathbb{R}^{d \times d}$ defined by

$$(\mathbf{D}_1)_{jj} = \frac{1}{2\sqrt{n_{j,+1}n_{j,-1}}} \ , \ (\mathbf{D}_0)_{jj} = \frac{n_{j,+1} - n_{j,-1}}{2n\sqrt{n_{j,+1}n_{j,-1}}} \ . \tag{11}$$

With the empirical covariance matrix $\hat{\mathbf{\Sigma}} = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$, we can thus express $\mathbf{Q}$ as $\mathbf{Q} = \mathbf{D}_0\mathbf{1}_{d \times n}\mathbf{X}\mathbf{w} + n\mathbf{D}_1\hat{\mathbf{\Sigma}}\mathbf{w}$. Here it becomes apparent how the FIRM, as opposed to the plain $\mathbf{w}$, takes the correlation structure of the features into account. Further, for a uniformly distributed feature j (i.e. $\Pr(X_j = t) = \frac{1}{2}$), the standard scaling is reproduced, i.e. $(\mathbf{D}_1)_{jj} = \frac{1}{n}\mathbf{I}$, and the other terms vanish, as $(\mathbf{D}_0)_{jj} = 0$.

For $\mathbf{X}$ containing each possible feature vector exactly once, corresponding to the uniform distribution and thus independent features, $\mathbf{M}^\top\mathbf{X}$ is the identity matrix (the covariance matrix), recovering the above solution of $\mathbf{Q} = \mathbf{w}$.

Continuous Data With Normal Distribution. If we consider normally distributed input features and assume a linear scoring function (1), the approximations above (Section 2.2) are exact. Hence, the expected conditional score of an input variable is

$$q_j(t) = \frac{t}{\mathbf{\Sigma}_{jj}}\mathbf{w}^\top\mathbf{\Sigma}_{j\bullet} + b \ . \tag{12}$$

With the diagonal matrix $\mathbf{D}$ of standard deviations of the features, i.e. with entries $\mathbf{D}_{jj} = \sqrt{\mathbf{\Sigma}_{jj}}$, this is summarized in

$$\mathbf{q} = b\mathbf{1}_d + t\mathbf{D}^{-2}\mathbf{\Sigma}\mathbf{w} \ .$$

Exploiting that the marginal distribution of X with respect to the j-th variable is again a zero-mean normal, $X_j \sim \mathcal{N}(0, \mathbf{\Sigma}_{jj})$, this yields $\mathbf{Q} = \mathbf{D}^{-1}\mathbf{\Sigma}\mathbf{w}$. For uncorrelated features, $\mathbf{D}$ is the square root of the diagonal covariance matrix $\mathbf{\Sigma}$, so that we get $\mathbf{Q} = \mathbf{D}\mathbf{w}$. Thus rescaling of the features is reflected by a corresponding rescaling of the importances — unlike the plain weights, FIRM cannot be manipulated this way.

As FIRM weights the scoring vector by the correlation $D^{-1}\Sigma$ between the variables, it is in general more stable and more reliable than the information obtained by the scoring vector alone. As an extreme case, let us consider a two-dimensional variable (X_1, X_2) with almost perfect correlation $\rho = cor(X_1, X_2) \approx 1$. In this situation, L1-type methods like lasso tend to select randomly only one of these variables, say $\mathbf{w} = (w_1, 0)$, while L2-regularization tends to give almost equal weights to both variables. FIRM compensates for the arbitrariness of lasso by considering the correlation structure of X: in this case $q = (w_1, \rho w_1)$, which is similar to what would be found for an equal weighting $\mathbf{w} = \frac{1}{2}(w, w)$, namely $q = (w(1 + \rho)/2, w(1 + \rho)/2)$.

Linear Regression. Here we assume that the scoring function s is the solution of an unregularized linear regression problem, $\min_{\mathbf{w},b} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$; thus $\mathbf{w} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}$.

Plugging this into the expression for $\mathbf{Q}$ from above yields

$$\mathbf{Q} = \mathbf{D}^{-1}\mathbf{\Sigma}\left(n\hat{\mathbf{\Sigma}}\right)^{-1}\mathbf{X}^\top\mathbf{y} \ . \tag{13}$$

For infinite training data, $\hat{\mathbf{\Sigma}} \longrightarrow \mathbf{\Sigma}$, we thus obtain $\mathbf{Q} = \frac{1}{n}\mathbf{D}^{-1}\mathbf{X}^\top\mathbf{y}$. Here it becomes apparent how the normalization makes sense: it renders the importance independent of a rescaling of the features. When a feature is inflated by a factor, so is its standard deviation $\mathbf{D}_{jj}$, and the effect is cancelled by multiplying them.

3   Simulation Studies

We now illustrate the usefulness of FIRM in a few preliminary computational experiments on artificial data.

3.1   Binary Data

We consider the problem of learning the Boolean formula $x_1 \vee (\neg x_1 \wedge \neg x_2)$. An SVM with polynomial kernel of degree 2 is trained on all 8 samples that can be drawn from the Boolean truth table for the variables $(x_1, x_2, x_3) \in \{0, 1\}^3$. Afterwards, we compute FIRM both based on the trained SVM ($\mathbf{w}$) and based on the true labelings (y). The results are displayed in Figure 1.

Note that the raw SVM $\mathbf{w}$ can assign non-zero weights only to feature space dimensions (here, input variables and their pairwise conjunctions, corresponding to the quadratic kernel); all other features, here for example pairwise disjunctions, are implicitly assigned zero. The SVM assigns the biggest weight to x_2, followed by $x_1 \wedge x_2$. In contrast, for the SVM-based FIRM the most important features are $x_1 \wedge \neg x_2$ followed by $\neg x_{1/2}$, which more closely resembles the truth. Note that, due to the low degree of the polynomial kernel, the SVM not capable of learning the function "by heart"; in other words, we have an underfitting situation. In fact, we have $s(\mathbf{x}) = 1.\bar{6}$ for $(x_1, x_2) = (0, 1)$.
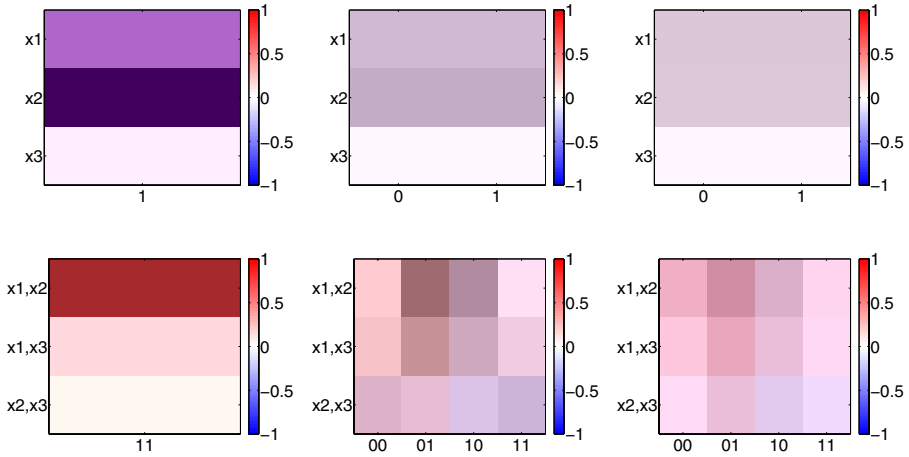
Fig. 1. FIRMs and SVM-w for the Boolean formula $x_1 \vee (\neg x_1 \wedge \neg x_2)$. The figures display heat maps of the scores, blue denotes negative label, red positive label, white is neutral. The upper row of heat maps shows the scores assigned to a single variable, the lower row shows the scores assigned to pairs of variables. The first column shows the SVM-w assigning a weight to the monomials x_1, x_2, x_3 and $x_1 x_2, x_1 x_3, x_2 x_3$ respectively. The second column shows FIRMs obtained from the trained SVM classifier. The third column shows FIRMs obtained from the true labeling.

The difference in $y-$FIRM and SVM-FIRM underlines that — as intended — FIRM helps to understand the learner, rather than the problem. Nevertheless a quite good approximation to the truth is found as displayed by FIRM on the true labels, for which all seven 2-tuples that lead to true output are found (black blocks) and only $\neg x_1 \wedge x_2$ leads to a false value (stronger score). Values where $\neg x_1$ and x_2 are combined with x_3 lead to a slightly negative value.

3.2   Gaussian Data

Here, we analyze a toy example to illustrate FIRM for real valued data. We consider the case of binary classification in three real-valued dimensions. The first two dimensions carry the discriminative information (cf. Figure 2a), while the third only contains random noise. The second dimension contains most discriminative information and we can use FIRM to recover this fact. To do so, we train a linear SVM classifier to obtain a classification function $s(\mathbf{x})$. Now we use the linear regression approach to model the conditional expected scores q_i (see Figure 2b-d for the three dimensions). We observe that dimension two indeed shows the strongest slope indicating the strongest discriminative power, while the third (noise) dimension is identified as uninformative.
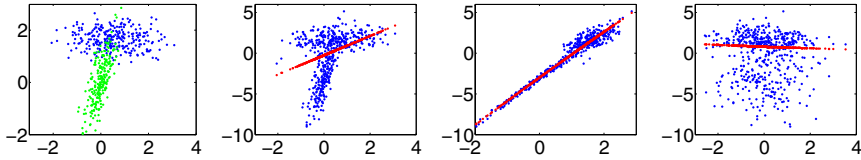
Fig. 2. Binary classification performed on continuous data that consists of two 3d Gaussians constituting the two classes (with x_3 being pure noise). From left to right a) Of the raw data set x_1, x_2 are displayed. b) Score of the linear discrimination function $s(\mathbf{x}_i)$ (blue) and conditional expected score $q_1((\mathbf{x}_i)_1)$ (red) for the first dimension of $\mathbf{x}$. c) $s(\mathbf{x}_i)$ and $q_2((\mathbf{x}_i)_2)$ for varying x_2. As the variance of q is highest here, this is the discriminating dimension (closely resembling the truth). d) $s(\mathbf{x}_i)$ and $q_3((\mathbf{x}_i)_3)$ for varying x_3. Note that x_3 is the noise dimension and does not contain discriminating information (as can be seen from the small slope of q_3).

3.3   Sequence Data

As shown above (Section 1.2), for sequence data FIRM is essentially identical to the previously published technique POIMs [17]. To illustrate its power for sequence classification, we use a toy data set from [9]: random DNA sequences are generated, and for the positive class the sub-sequence `GATTACA` is planted at a random position centered around 35 (rounded normal distribution with SD=7). As biological motifs are typically not perfectly conserved, the planted consensus sequences are also mutated: for each planted motif, a single position is randomly chosen, and the incident letter replaced by a random letter (allowing for no change for $\sim 25\%$ of cases). An SVM with WDS kernel [6] is trained on 2500 positive and as many negative examples.

Two analyses of feature importance are presented in Figure 3: one based on the feature weights $\mathbf{w}$ (left), the other on the feature importance Q (right). It is apparent that FIRM identifies the `GATTACA` feature as being most important at positions between 20 and 50, and it even attests significant importance to the strings with edit distance 1. The feature weighting $\mathbf{w}$, on the other hand, fails completely: sequences with one or two mutations receive random importance, and even the importance of the consensus `GATTACA` itself shows erratic behavior.

The reason is that the appearance of the exact consensus sequence is not a reliable feature, as is mostly occurs mutated. More useful features are substrings of the consensus, as they are less likely to be hit by a mutation. Consequently there is a large number of such features that are given high weight be the SVM. By taking into account the correlation of such short substrings with longer ones, in particular with `GATTACA`, FIRM can recover the "ideal" feature which yields the highest SVM score. Note that this "intelligent" behavior arises automatically; no more domain knowledge than the Markov distribution (and it is only 0-th order uniform!) is required. The practical value of POIMs for real world biological problems has been demonstrated in [17].
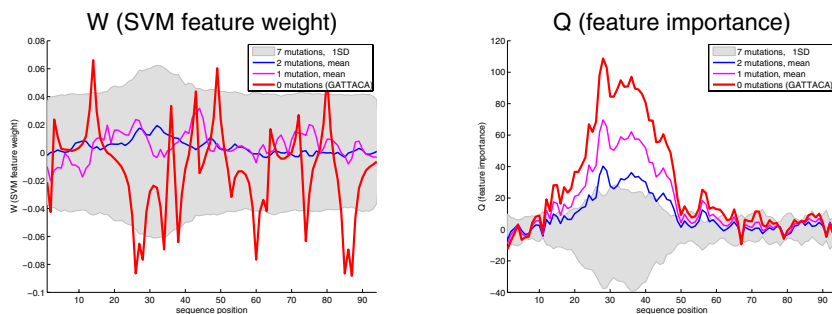
Fig. 3. Feature importance analyses based on (left) the SVM feature weighting **w** and (right) FIRM. The shaded area shows the ±1 SD range of the importance of completely irrelevant features (length 7 sequences that disagree to `GATTACA` at every position). The red lines indicate the positional importances of the exact motif `GATTACA`; the magenta and blue lines represent average importances of all length 7 sequences with edit distances 1 and 2, respectively, to `GATTACA`. While the feature weighting approach cannot distinguish the decisive motiv from random sequences, FIRM identifies it confidently.

4   Summary and Conclusions

We propose a new measure that quantifies the relevance of features. We take up the idea underlying a recent sequence analysis method (called POIMs, [17]) — to assess the importance of substrings by their impact on the expected score — and generalize it to arbitrary continuous features. The resulting *feature importance ranking measure* FIRM has invariance properties that are highly desirable for a feature ranking measure. First, it is "objective": it is invariant with respect to translation, and reasonably invariant with respect to rescaling of the features. Second, to our knowledge FIRM is the first feature ranking measure that is totally "universal", i.e. which allows for evaluating *any* feature, irrespective of the features used in the primary learning machine. It also imposes no restrictions on the learning method. Most importantly, FIRM is "intelligent": it can identify features that are not explicitly represented in the learning machine, due to the correlation structure of the feature space. This allows, for instance, to identify sequence motifs that are longer than the considered substrings, or that are not even present in a single training example.

By definition, FIRM depends on the distribution of the input features, which is in general not available. We showed that under various scenarios (e.g. binary features, normally distributed features), we can obtain approximations of FIRM that can be efficiently computed from data. In real-world scenarios, the underlying assumptions might not always be fulfilled. Nevertheless, e.g. with respect to the normal distribution, we can still interpret the derived formulas as an estimation based on first and second order statistics only.

While the quality of the computed importances does depend on the accuracy of the trained learning machine, FIRM can be used with any learning framework. It can even be used without a prior learning step, on the raw training data. Usually, feeding training labels as scores into FIRM will yield similar results as using a learned function; this is natural, as both are supposed to be highly correlated.

However, the proposed indirect procedure may improve the results due to three effects: first, it may smooth away label errors; second, it extends the set of labeled data from the sample to the entire space; and third, it allows to explicitly control and utilize distributional information, which may not be as pronounced in the training sample. A deeper understanding of such effects, and possibly their exploitation in other contexts, seems to be a rewarding field of future research.

Based on the unique combination of desirable properties of FIRM, and the empirical success of its special case for sequences, POIMs [17], we anticipate FIRM to be a valuable tool for gaining insights where alternative techniques struggle.

References

1. Bennett, K., Mangasarian, O.: Robust linear programming discrimination of two linearly inseparable sets. Optimization Methods and Software 1, 23–34 (1992)
2. Cortes, C., Gretton, A., Lanckriet, G., Mohri, M., Rostamizedeh, A.: Outcome of the NIPS 2008 workshop on kernel learning: Automatic selection of optimal kernels (2008)
3. Friedman, J.: Greedy function approximation: a gradient boosting machine. Annals of Statistics 29, 1189–1232 (2001)
4. Graf, A., Wichmann, F., Bülthoff, H.H., Schölkopf, B.: Classification of faces in man and machine. Neural Computation 18, 143–165 (2006)
5. Lanckriet, G.R.G., Cristianini, N., Ghaoui, L.E., Bartlett, P., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research 5, 27–72 (2004)
6. Rätsch, G., Sonnenburg, S., Schölkopf, B.: RASE: Recognition of alternatively spliced exons in C. elegans. Bioinformatics 21(suppl. 1), i369–i377 (2005)
7. Schäfer, J., Strimmer, K.: A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. Statistical Applications in Genetics and Molecular Biology 4(1), 32 (2005)
8. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)
9. Sonnenburg, S., Rätsch, G., Schäfer, C.: Learning interpretable SVMs for biological sequence classification. In: Miyano, S., Mesirov, J., Kasif, S., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) RECOMB 2005. LNCS (LNBI), vol. 3500, pp. 389–407. Springer, Heidelberg (2005)

10. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large Scale Multiple Kernel Learning. Journal of Machine Learning Research 7, 1531–1565 (2006)
11. Strobl, C., Boulesteix, A., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. BMC Bioinformatics 9(1), 307 (2008)
12. Strobl, C., Boulesteix, A., Zeileis, A., Hothorn, T.: Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC bioinformatics 8(1), 25 (2007)
13. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society, Series B 58(1), 267–288 (1996)
14. Üstün, B., Melssen, W.J., Buydens, L.M.: Visualisation and interpretation of support vector regression models. Analytica Chimica Acta 595(1-2), 299–309 (2007)
15. van der Laan, M.: Statistical inference for variable importance. The International Journal of Biostatistics 2(1), 1008 (2006)
16. Zien, A., Philips, P., Sonnenburg, S.: Computing Positional Oligomer Importance Matrices (POIMs). Res. Report; Electronic Publ. 2, Fraunhofer FIRST (December 2007)
17. Zien, A., Sonnenburg, S., Philips, P., Rätsch, G.: POIMs: Positional Oligomer Importance Matrices – Understanding Support Vector Machine Based Signal Detectors. In: Proceedings of the 16th International Conference on Intelligent Systems for Molecular Biology (2008)