arXiv:0912.1128v1 [stat.ML] 6 Dec 2009

How to Explain Individual Classification Decisions

David Baehrens* baehrens@cs.tu-berlin.de
Timon Schroeter* timon@cs.tu-berlin.de
Technische Universität Berlin
Franklinstr. 28/29, FR 6-9
10587 Berlin, Germany
Stefan Harmeling* stefan.harmeling@tuebingen.mpg.de
MPI for Biological Cybernetics
Spemannstr. 38
72076 Tübingen, Germany
Motoaki Kawanabe motoaki.kawanabe@first.fraunhofer.de
Fraunhofer Institute FIRST.IDA
Kekulestr.7
12489 Berlin, Germany
and
Technische Universität Berlin
Franklinstr. 28/29, FR 6-9
10587 Berlin, Germany
Katja Hansen khansen@cs.tu-berlin.de
Klaus-Robert Müller klaus-robert.mueller@tu-berlin.de
Technische Universität Berlin
Franklinstr. 28/29, FR 6-9
10587 Berlin, Germany

Editor: Carl Edward Rasmussen

Abstract

After building a classifier with modern tools of machine learning we typically have a black box at hand that is able to predict well for unseen data. Thus, we get an answer to the question *what* is the most likely label of a given unseen data point. However, most methods will provide no answer *why* the model predicted the particular label for a single instance and what features were most influential for that particular instance. The only method that is currently able to provide such explanations are decision trees. This paper proposes a procedure which (based on a set of assumptions) allows to explain the decisions of *any* classification method.

Keywords: explaining, nonlinear, black box model, kernel methods, Ames mutagenicity

∗. The first three authors contributed equally.

1. Introduction

Automatic nonlinear classification is a common and powerful tool in data analysis. Machine learning research has created methods that are practically useful and that can classify unseen data after being trained on a limited training set of labeled examples.

Nevertheless, most of the algorithms do not *explain* their decision. However in practical data analysis it is essential to obtain an instance based explanation, i.e. we would like to gain an understanding what input features made the nonlinear machine give its answer for each individual data point.

Typically, explanations are provided jointly for all instances of the training set, for example feature selection methods (including Automatic Relevance Determination) find out which inputs are salient for a good generalization (see for a review Guyon and Elisseeff, 2003). While this can give a coarse impression about the global usefulness of each input dimension, it is still an ensemble view and does not provide an answer on an instance basis.[1] In the neural network literature also solely an ensemble view was taken in algorithms like input pruning (e.g. Bishop, 1995; LeCun, Bottou, Orr, and Müller, 1998). The only classification which does provide individual explanations are decision trees (e.g. Hastie, Tibshirani, and Friedman, 2001).

This paper proposes a simple framework that provides local explanation vectors applicable to *any* classification method in order to help understanding prediction results for single data instances. The local explanation yields the features being relevant for the prediction at the very points of interest in the data space and is able to spot local peculiarities which are neglected in the global view e.g. due to cancellation effects.

The paper is organized as follows: We define local explanation vectors as class probability gradients in Section 2 and give an illustration for Gaussian Process Classification (GPC). Some methods output a prediction without a direct probability interpretation. For these we propose in Section 3 a way to estimate local explanations. In Section 4 we will apply our methodology to learn distinguishing properties of Iris flowers by estimating explanation vectors for a k-NN classifier applied to the classic Iris data set. Section 5 will discuss how our approach applied to a SVM classifier allows us to explain how digits "two" are distinguished from digit "8" in the USPS data set. In Section 6 we discuss a more real-world application scenario where the proposed explanation capabilities prove useful in drug discovery: Human experts regularly decide how to modify existing lead compounds in order to obtain new compounds with improved properties. Models capable of explaining predictions can help in the process of choosing promising modifications. Our automatically generated explanations match with chemical domain knowledge about toxifying functional groups of the compounds in question. Section 7 contrasts our approach with related work and Section 8 discusses characteristic properties and limitations of our approach, before we conclude the paper in Section 9.

1. This point is illustrated in Figure 1 (Section 2). Applying feature selection methods to the training set (a) will lead to the (correct) conclusion that both dimensions are equally important for accurate classification. As an alternative to this ensemble view, one may ask: Which features (or combinations thereof) are most influential in the vicinity of each particular instance. As can be seen in Figure 1 (c), the answer depends on where the respective instance is located. On the hypotenuse and at the corners of the triangle, both features contribute jointly, whereas along each of the remaining two edges the classification depends almost completely on just one of the features.

2. Definitions of Explanation Vectors

In this Section we will give definitions for our approach of local explanation vectors in the classification setting. We start with a theoretical definition for multi-class Bayes classification and then give a specialized definition being more practical for the binary case.

For the multi-class case, suppose we are given data points $x_1, \ldots, x_n \in \Re^d$ with labels $y_1, \ldots, y_n \in \{1, \ldots, C\}$ and we intend to learn a function that predicts the labels of unlabeled data points. Assuming that the data could be modeled as being IID-sampled from some unknown joint distribution $P(X, Y)$, in theory, we can define the Bayes classifier,

$$g^*(x) = \arg \min_{c \in \{1,\ldots,C\}} P(Y \neq c \mid X = x)$$

which is optimal for the 0-1 loss function (see Devroye, Györfi, and Lugosi, 1996).

For the Bayes classifier we define the *explanation vector* of a data point x_0 to be the derivative with respect to x at $x = x_0$ of the conditional probability of $Y \neq g^*(x_0)$ given $X = x$, or formally,

Definition 1

$$\zeta(x_0) := \frac{\partial}{\partial x} P(Y \neq g^*(x_0) \mid X = x)\Big|_{x=x_0}$$

Note that $\zeta(x_0)$ is a d-dimensional vector just like x_0 is. The classifier g^* partitions the data space $\Re^d$ into up to C parts on which g^* is constant. We assume that the conditional distribution $P(Y = c \mid X = x)$ is first-order differentiable w.r.t. x for all classes c and over the entire input space. For instance, the assumption holds, if $P(X = x \mid Y = c)$ is for all c first-order differentiable in x and the supports of the class densities overlap around the boarder for all the neighboring pairs in the partition by the Bayes classifier. The vector $\zeta(x_0)$ defines on each of those parts a vector field that characterizes the flow away from the corresponding class. Thus entries in $\zeta(x_0)$ with large absolute values highlight features that will influence the class label decision of x_0. A positive sign of such an entry implies that increasing that feature would lower the probability that x_0 is assigned to $g^*(x_0)$. Ignoring the orientations of the explanation vectors, ζ forms a continuously changing (orientation-less) vector field along which the class labels change. This vector field lets us *locally* understand the Bayes classifier.

We remark that $\zeta(x_0)$ becomes a zero vector, e.g. when $P(Y \neq g^*(x_0) \mid X = x)|_{x=x_0}$ is equal to one in some neighborhood of x_0. Our explanation vector fits well to probabilistic classifiers such as Gaussian Process Classification (GPC), where the conditional distribution $P(Y = c \mid X = x)$ is usually not completely flat in some regions. In the case of deterministic classifiers, despite of this issue, Parzen window estimators with appropriate widths (Section 3) can provide meaningful explanation vectors for many samples in practice (see also Section 8).

For the case of binary classification we directly define local explanation vectors as local gradients of the probability function $p(x) = P(Y = 1 \mid X = x)$ of the learned model for the positive class.

So for a probability function $p : \Re^d \to [0, 1]$ of a classification model learned from examples $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in \Re^d \times \{-1, +1\}$ the explanation vector for a classified test point x_0 is the local gradient of p at x_0:

Definition 2

$$\eta_p(x_0) := \nabla p(x)|_{x=x_0}$$

By this definition the explanation η is again a d-dimensional vector just like the test point x_0 is. The sign of each of its individual entries indicates whether the prediction would increase or decrease when the corresponding feature of x_0 is increased locally and each entry's absolute value give the amount of influence in the change in prediction. As a vector η gives the direction of the steepest ascent from the test point to higher probabilities for the positive class. For binary classification the negative version $-\eta_p(x_0)$ indicates the changes in features needed to increase the probability for the negative class which may be especially useful for x_0 predicted in the positive class.

For an example we apply Definition 2 to model predictions learned by Gaussian Process Classification (GPC), see Rasmussen and Williams (2006). GPC is used here for three reasons:
(i) In our real-world application we are interested in classifying data from drug discovery, which is an area where Gaussian processes have proven to show state-of-the-art performance, see e.g. Obrezanova, Csányi, Gola, and Segall (2007); Schroeter, Schwaighofer, Mika, ter Laak, Sülzle, Ganzer, (2007c); Schroeter, Schwaighofer, Mika, Laak, Suelzle, Ganzer, Heinrich, and Müller (2007a,b); Schwaighofer, Schroeter, Mika, Laub, ter Laak, Sülzle, Ganzer, Heinrich, and Müller (2007); Schwaighofer, Schroeter, Mika, Hansen, ter Laak, Lienau, Reichel, Heinrich, and Müller (2008); Obrezanova, Gola, Champness, and Segall (2008). It is natural to expect a model with high prediction accuracy on a complex problem to capture relevant structure of the data which is worth explaining and may give domain specific insights in addition to the values predicted. For an evaluation of the explaining capabilities of our approach on a complex problem from chemoinformatics see Section 6.
(ii) GPC does model the class probability function used in Definition 2 directly. For other classification methods such as Support Vector Machines which do not provide a probability function as its output in Section 3 we give an example for an estimation method starting from Definition 1.
(iii) The local gradients of the probability function can be calculated analytically for differentiable kernel as we discuss next.

Let $\overline{f}(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$ be a GP model trained on sample points $x_1, \ldots, x_n \in \Re^d$ where k is a kernel function and α_i are the learned weights of each sample point. For a test point $x_0 \in \Re^d$ let $\mathrm{var}_f(x_0)$ be the variance of $f(x_0)$ under the GP posterior for f. Because the posterior cannot be calculated analytically for GP classification models, we used an approximation by expectation propagation (EP) (Kuss and Ramussen, 2005). In the case of the probit likelihood term defined by the error function, the probability for being of the positive class $p(x_0)$ can be computed easily from this approximated posterior as

$$p(x_0) = \frac{1}{2}\mathrm{erfc}\left(\frac{-\overline{f}(x_0)}{\sqrt{2} * \sqrt{1 + \mathrm{var}_f(x_0)}}\right),$$

where erfc denotes the complementary error function (see Equation 6 in Schwaighofer, Schroeter, Mika, Hansen, 2008).

Then the local gradient of $p(x_0)$ is given by

$$
\begin{aligned}
&\nabla p(x)|_{x=x_0} \\
=&\nabla \frac{1}{2}\text{erfc}\left(\frac{-\overline{f}(x)}{\sqrt{2}*\sqrt{1+\text{var}_f(x)}}\right)\Bigg|_{x=x_0} \\
=&\nabla \frac{1}{2}\left(1-\text{erf}\left(\frac{-\overline{f}(x)}{\sqrt{2}*\sqrt{1+\text{var}_f(x)}}\right)\right)\Bigg|_{x=x_0} \\
=&-\frac{1}{2}\nabla \text{ erf}\left(\frac{-\overline{f}(x)}{\sqrt{2}*\sqrt{1+\text{var}_f(x)}}\right)\Bigg|_{x=x_0} \\
=&-\frac{\exp\left(\frac{-\overline{f}(x_0)^2}{2(1+\text{var}_f(x_0))}\right)}{\sqrt{\pi}}\nabla\left(\frac{-\overline{f}(x)}{\sqrt{2}*\sqrt{1+\text{var}_f(x)}}\right)\Bigg|_{x=x_0} \\
=&-\frac{\exp\left(\frac{-\overline{f}(x_0)^2}{2(1+\text{var}_f(x_0))}\right)}{\sqrt{\pi}}\left(-\frac{1}{\sqrt{2}}\nabla\left(\frac{\overline{f}(x)}{\sqrt{1+\text{var}_f(x)}}\right)\Bigg|_{x=x_0}\right) \\
=&\frac{\exp\left(\frac{-\overline{f}(x_0)^2}{2(1+\text{var}_f(x_0))}\right)}{\sqrt{2\pi}}\left(\frac{\nabla\overline{f}(x)|_{x=x_0}}{\sqrt{1+\text{var}_f(x_0)}}+\overline{f}(x_0)\left(\nabla \text{ var}_f(x)|_{x=x_0}*-\frac{1}{2}\left(1+\text{var}_f(x_0)\right)^{-\frac{3}{2}}\right)\right) \\
=&\frac{\exp\left(\frac{-\overline{f}(x_0)^2}{2(1+\text{var}_f(x_0))}\right)}{\sqrt{2\pi}}\left(\frac{\nabla\overline{f}(x)|_{x=x_0}}{\sqrt{1+\text{var}_f(x_0)}}-\frac{1}{2}\frac{\overline{f}(x_0)}{\left(1+\text{var}_f(x_0)\right)^{\frac{3}{2}}}\nabla\text{var}_f(x)|_{x=x_0}\right).
\end{aligned}
$$

As a kernel function choose e.g. the RBF-kernel $k(x_0,x_1)=\exp(-w(x_0-x_1)^2)$, which has the derivative $(\partial/\partial x_{0,j})k(x_0,x_1)=-2w\exp(-w(x_0-x_1)^2)(x_{0,j}-x_{1,j})$ for $j\in\{1,\ldots,d\}$. Then the elements of the local gradient $\nabla\overline{f}(x)|_{x=x_0}$ are

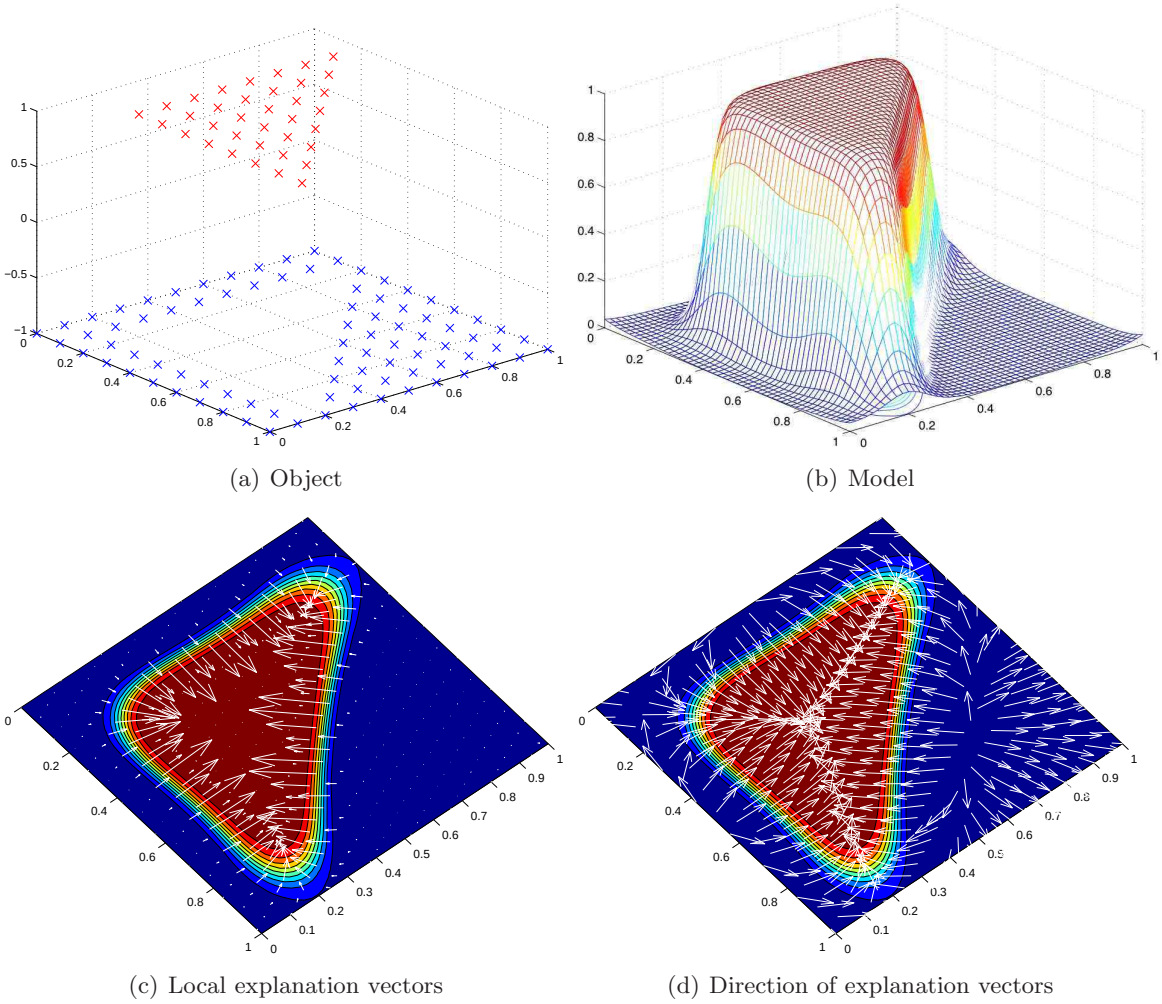
$$\frac{\partial\overline{f}}{\partial x_{0,j}}=-2w\sum_{i=1}^{n}\alpha_i\exp(-w(x_0-x_i)^2)(x_{0,j}-x_{i,j}) \quad \text{for } j\in\{1,\ldots,d\}.$$

For $\text{var}_f(x_0)=k(x_0,x_0)-k_*^T(K+\Sigma)^{-1}k_*$ the derivative is given by[2]

$$\nabla\text{var}_f(x)|_{x=x_0}=\frac{\partial\text{var}_f}{\partial x_{0,j}}=\left(\frac{\partial}{\partial x_{0,j}}k(x_0,x_0)\right)-2*k_*^T(K+\Sigma)^{-1}\frac{\partial}{\partial x_{0,j}}k_* \quad \text{for } j\in\{1,\ldots,d\}.$$

Panel (a) of Figure 1 shows the training data of a simple object classification task and panel (b) shows the model learned using GPC[3]. The data is labeled -1 for the blue points and $+1$ for the red points. As illustrated in panel (b) the model is a probability function for the positive class which gives every data point a probability of being in this

2. Here $k_*=(k(x_0,x_1),\ldots,k(x_0,x_n))^T$ is the evaluation of the kernel function between the test point x_0 and every training point. Σ is the diagonal matrix of the variance site parameter. For details see Rasmussen and Williams (2006, Chapter 3)
3. Hyperparameters were tuned by a gradient ascend on the evidence.



(a) Object (b) Model

(c) Local explanation vectors (d) Direction of explanation vectors

Figure 1: Explaining simple object classification with Gaussian Processes

class. Panel (c) shows the probability gradient of the model together with the local gradient explanation vectors. On the hypotenuse and at the corners of the triangle explanations from both features interact towards the triangle class while along the edges the importance of one of the two feature dimensions singles out. At the transition from the negative to the positive class the length of the local gradient vectors represents the increased importance of the relevant features. In panel (d) we see that explanations close to the edges of the plot (especially in the right hand side corner) point away from the positive class. However, panel (c) shows that their magnitude is very small. For discussion of this issue, see Section 8.

3. Estimating Explanation Vectors

Several classifier methods estimate directly the decision rule, which often has no interpretation as a probability function which is used in our Definition 2 in Section 2. For example

Support Vector Machines estimate a decision function of the form

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) + b,$$

$\alpha_i, b \in \Re$. Suppose we have two classes (each with one cluster) in one dimension (see Figure 2) and train a SVM with RBF kernel. For points outside the data clusters $f(x)$ tends to zero. Thus, the derivative of $f(x)$ (shown as arrows above the curves) for points on the very left or on the very right side of the axis will point to the wrong side. In the

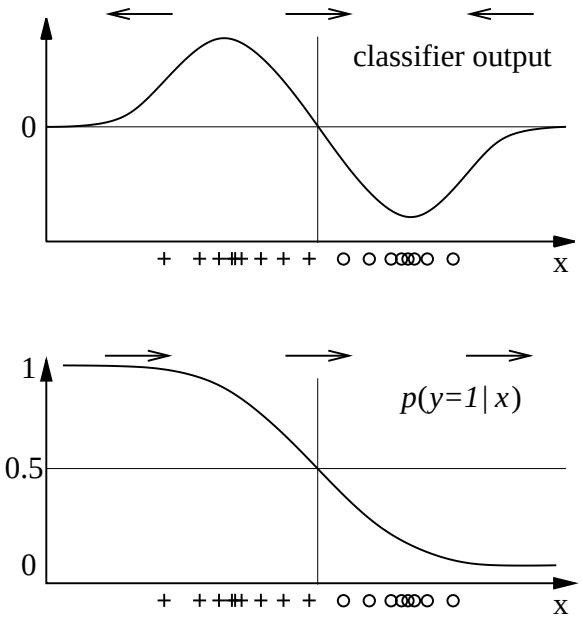


Figure 2: Classifier output of an SVM (top) compared to $p(y=1|x)$ (bottom).

following, we will explain how explanations can be obtained for such classifiers.

In practice we do not have access to the true underlying distribution $P(X, Y)$. Consequently, we have no access to the Bayes classifier as defined in Section 2. Instead we can apply sophisticated learning machinery like Support Vector Machines (Vapnik, 1995; Schölkopf and Smola, 2002; Müller, Mika, Rätsch, Tsuda, and Schölkopf, 2001) that estimates some classifier g that tries to mimic g^*. For test data points $z_1, \ldots, z_m \in \Re^d$ which are assumed to be sampled from the same unknown distribution as the training data, g estimates labels $g(z_1), \ldots, g(z_m)$. Now, instead of trying to explain g^* to which we have no access, we will define explanation vectors that help us understand the classifier g on the test data points.

Since we do not assume that we have access to some intermediate real-valued classifier output here (of which g might be a thresholded version and which further might not be an estimate of $P(Y = c \mid X = x)$), we suggest to approximate g by another classifier $\hat{g}$ the actual form of which resembles the Bayes classifier. There are several choices for $\hat{g}$, e.g.

GPC, logistic regression and Parzen windows.[4] In this paper we apply Parzen windows to the training points to estimate the weighted class densities $P(Y=c) \cdot P(X \mid Y=c)$,

$$\hat{p}_\sigma(x, y=c) = \frac{1}{n} \sum_{i \in I_c} k_\sigma(x - x_i) \tag{1}$$

for the index set $I_c = \{i \mid g(x_i) = c\}$ and with $k_\sigma(z)$ being a Gaussian kernel $k_\sigma(z) = \exp(-0.5\, z^\top z/\sigma^2)/\sqrt{2\pi\sigma^2}$ (as always other kernels are also possible). This estimates $P(Y = c \mid X = x)$ for all c,

$$\hat{p}_\sigma(y=c|x) = \frac{\hat{p}_\sigma(x, y=c)}{\hat{p}_\sigma(x, y=c) + \hat{p}_\sigma(x, y \neq c)} \approx \frac{\sum_{i \in I_c} k_\sigma(x - x_i).}{\sum_i k_\sigma(x - x_i),} \tag{2}$$

and thus an estimate of the Bayes classifier (that mimics g),

$$\hat{g}_\sigma(x) = \arg \min_{c \in \{1,\ldots,C\}} \hat{p}_\sigma(y \neq c \mid x).$$

This approach has the advantage, that we can use our estimated classifier g to generate any amount of labeled data for for constructing $\hat{g}$. The single hyper-parameter σ is chosen, such that $\hat{g}$ approximates g (which we want to explain), i.e.

$$\hat{\sigma} := \arg\min_\sigma \sum_{j=1}^{m} I\left\{g(z_j) \neq \hat{g}_\sigma(z_j)\right\},$$

where $I\{\cdots\}$ is the indicator function. σ is assigned the constant value $\hat{\sigma}$ from here on and omitted as a subscript. For $\hat{g}$ it is straightforward to define explanation vectors:

Definition 3

$$\hat{\zeta}(z) := \frac{\partial}{\partial x}\, \hat{p}(y \neq g(z) \mid x)\Big|_{x=z} = \frac{\Big(\sum_{i \notin I_{g(z)}} k(z - x_i)\Big)\Big(\sum_{i \in I_{g(z)}} k(z - x_i)(z - x_i)\Big)}{\sigma^2 \left(\sum_{i=1}^n k(z - x_i)\right)^2}$$
$$- \frac{\Big(\sum_{i \notin I_{g(z)}} k(z - x_i)(z - x_i)\Big)\Big(\sum_{i \in I_{g(z)}} k(z - x_i)\Big)}{\sigma^2 \left(\sum_{i=1}^n k(z - x_i)\right)^2}$$

which is easily derived using Eq. (2) and the derivative of Eq. (1), see Appendix A.2.1. Note that we use g instead of $\hat{g}$. This choice ensures that the orientation of $\hat{\zeta}(z)$ fits to the labels assigned by g, which allows better interpretations.

In summary, we imitate the classifier g which we would like to explain locally, by a Parzen window classifier $\hat{g}$ that has the same form as the Bayes estimator and for which we can thus easily estimate the explanation vectors using Definition 3. Practically there are some caveats: the mimicking classifier $\hat{g}$ has to be estimated from g even in high dimensions; this needs to be done with care. However, in principle we have an arbitrary amount of training data available for constructing $\hat{g}$ since we may use our estimated classifier g to generate labeled data.

4. For Support Vector Machines Platt (1999) fits a sigmoid function to map the outputs to probabilities. In the following, we will present a more general method for estimating explanation vectors.

4. Explaining Iris Flower Classification by k-Nearest Neighbors

The Iris flower data set (introduced in Fisher, 1936) describes 150 flowers from the genus Iris by 4 features: sepal length, sepal width, petal length, and petal width, which are easily measured properties of certain leaves of the corolla of the flower. There are three clusters in that data which correspond to three different species: Iris setosa, Iris virginica, and Iris versicolor.

Let us consider the problem of classifying the data points of Iris versicolor (class 0) against the other two species (class 1). We applied some standard classification machinery to this problem as detailed in the following:

- Class 0 consists of all examples of Iris versicolor.
- Class 1 consists of all examples of Iris setosa and Iris virginica.
- Randomly split 150 data points into 100 training and 50 test examples.
- Normalize training and test set using the mean and variance of the training set.
- Apply k-nearest neighbor classification with $k = 4$ (chosen by leave-one-out cross validation on the training data).
- Training error is 3% (i.e. 3 mistakes in 100).
- Test error is 8% (i.e. 4 mistakes in 50).

In order to estimate explanation vectors we mimic the classification results with a Parzen window classifier. The best fit (3% error) is obtained with a kernel width of $\sigma = 0.26$ (chosen by leave-one-out cross validation on the training data).

Since the explanation vectors live in the input space we can visualize them with scatter plots of the initially measured features. The resulting *explanations* (i.e. vectors) for the test set are shown in Figure 3. The blue dots correspond to explanation vectors for Iris setosa and the red dots for Iris virginica (both class 1). Both groups of dots point to the green dots of Iris versicolor. The most important feature is the combination of petal length and petal width (see the corresponding panel), the product of which corresponds roughly to the area of the petals. However, the resulting explanations for the two species in class 1 are different:

- Iris setosa (class 1) is different from Iris versicolor (class 0) because its petal area is *smaller*.
- Iris virginica (class 1) is different from Iris versicolor (class 0) because its petal area is *larger*.

Also the dimensions of the sepal (another part of the blossom) is relevant, but not as distinguishing.

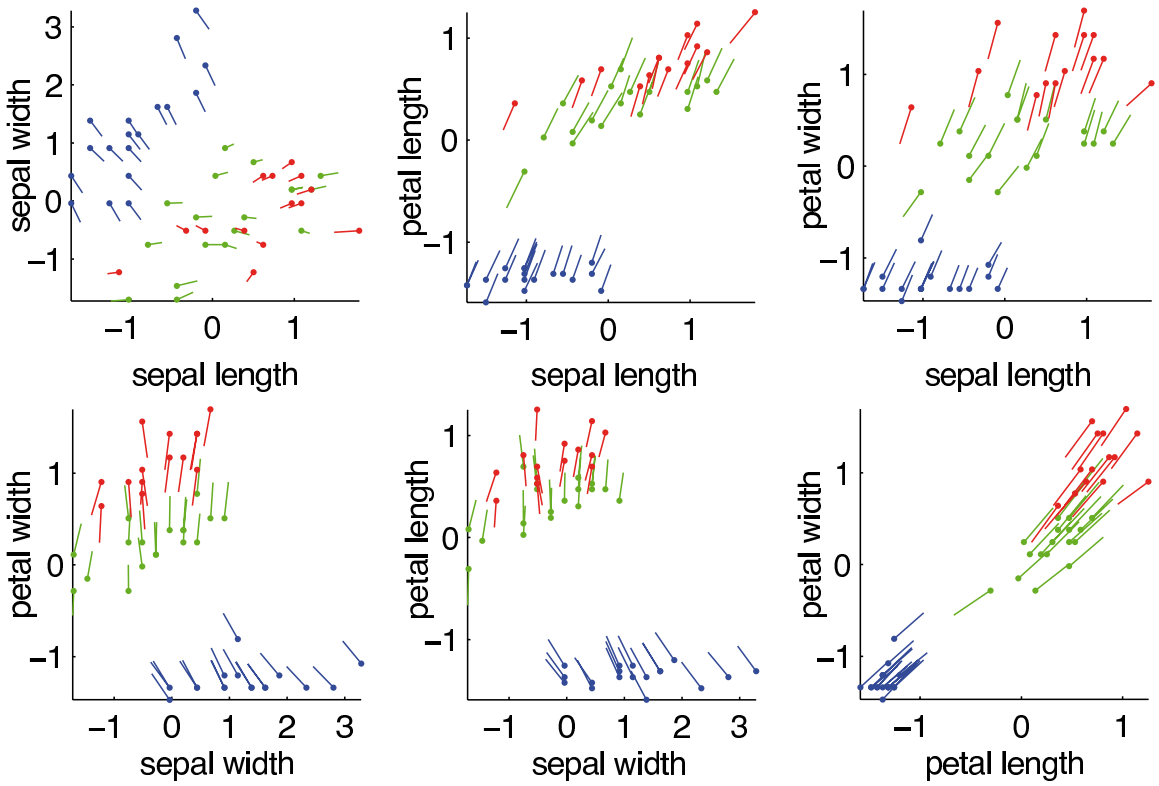


Figure 3: Scatter plots of the explanation vectors for the test data. Shown are all explanation vectors for both classes: class 1 containing Iris setosa (shown in blue) and Iris virginica (shown in red) versus class 0 containing only one species Iris versicolor (shown in green). Note that the explanations why an Iris flower is not an Iris versicolor is different for Iris setosa and Iris virginica.

Figure 4: USPS digits (training set): 'twos' (left) and 'eights' (right) with correct classification. For each digit from left to right: (i) explanation vector (with black being negative, white being positive), (ii) the original digit, (iii-end) artificial digits along the explanation vector towards the other class.

Figure 5: USPS digits (test set bottom part): 'twos' (left) and 'eights' (right) with correct classification. For each digit from left to right: (i) explanation vector (with black being negative, white being positive), (ii) the original digit, (iii-end) artificial digits along the explanation vector towards the other class.

5. Explaining USPS Digit Classification by Support Vector Machine

We now apply the framework of estimating explanation vectors to a high dimensional data set, the USPS digits. The classification problem that we designed for illustration purposes is detailed in the following list:

- all digits are 16×16 images which are reshaped to 256×1 dimensional column vectors
- classifier: SVM from Schwaighofer (2002) with RBF-kernel-width $\sigma = 1$ and regularization constant $C = 10$ (chosen by grid search in cross validation on the training data).
- training set: 47 'twos', 53 'eights'; training error 0.00
- test set: 48 'twos', 52 'eights'; test error 0.05

We approximated the estimated class labels obtained by the SVM with the Parzen window classifier (Parzen window size $\sigma = 10.2505$, chosen by grid search in cross validation on the training data). The SVM and the Parzen window classifier only disagreed on 2% of the test examples, so a good fit was achieved. Figures 4 and 5 show our results. All parts show three examples per row. For each example we display from left to right: (i) the explanation vector, (ii) the original digit, (iii-end) artificial digits along the explanation vector towards the other class.[5] These artificial digits should help to understand and interpret the explanation vector. Let us first have a look at the results on the training set:

Figure 4 (left panel): Let us focus on the top example framed in red. The line that forms the 'two' is part of some 'eight' from the data set. Thus the parts of the lines that are missing show up in the explanation vector: if the dark parts (which correspond to the missing lines) are added to the 'two' digit then it will be classified as an 'eight'. Or in other words, because of the lack of those parts the digit was classified as a 'two' and not as an 'eight'. A similar explanation holds for the middle example framed in red of the same Figure. Not all examples transform easily to 'eights': besides adding parts of black lines, some existing black spots (that the digit has to be a 'two') must be removed. This is reflected in the explanation vector by white spots/lines. Curious is the bottom 'two' framed in red, which is actually a dash and is in the data set by mistake. However, its explanation vector shows nicely which parts have to be added and which have to be removed.

Figure 4 (right panel): we see similar results for the 'eights' class. The explanation vectors again tell us how the 'eights' must change to become classified as 'twos'. However, sometimes the transformation does not reach the 'twos'. This is probably due to the fact that some of the 'eights' are inside the cloud of 'eights'.

On the test set the explanation vectors are not as pronounced as on the training set. However, they show similar tendencies:

5. For the sake of simplicity, no intermediate updates were performed, i.e. artificial digits were generated by taking equal sized steps in the direction given by the original explanation vector calculated fot the original digit.

Figure 5 (left panel): we see the correctly classified 'twos'. Let's focus on the example framed in red. Again the explanation vector shows us how to edit the image of the 'two' to make it some of the 'eights', i.e. exactly what parts of the digit have been important for the classification result. For several other 'twos' the explanation vectors do not directly lead to the 'eights' but weight the different parts of the digits which have been relevant for the classification.

Figure 5 (right panel): similarly to the training data, we see that also these explanation vectors are not bringing all 'eights' to 'two'. Their explanation vectors mainly suggest to remove most of the eights (the dark parts) and add some in the lower part (the light parts, which look like a white shadow).

Overall, our findings can be summarized, that the explanation vectors tell us how to edit our example digits to change the assigned class label. Hereby, we get a better understanding of the reasons why the chosen classifier classified the way it did.

6. Explaining Mutagenicity Classification by Gaussian Processes

In the following Section we describe an application of our local gradient explanation methodology to a complex real world data set. Our aim is to find structure specific to the problem domain that has *not* been fed into training explicitly but is captured implicitly by the GPC model in the high-dimensional feature space used to determine its prediction. We investigate the task of predicting Ames mutagenic activity of chemical compounds. Not being mutagenic (i.e. not able to cause mutations in the DNA) is an important requirement for compounds under investigation in drug discovery and design. The Ames test (Ames, Gurney, Miller, and Bartsch, 1972) is a standard experimental setup for measuring mutagenicity. The following experiments are based on a set of Ames test results for 6512 chemical compounds that we published previously.[6]

GPC was applied as detailed in the following:

- Class 0 consists of non-mutagenic compounds
- Class 1 consists of mutagenic compounds
- Randomly split 6512 data points into 2000 training and 4512 test examples such that:
  - The training set consists of equally many class 0 and class 1 examples.
  - For the steroid compound class the balance in the train and test set is enforced.
- 10 additional random splits were investigated individually. This confirmed the results presented below.
- Each example (chemical compound) is represented by a vector of counts of 142 molecular substructures calculated using the Dragon software (Todeschini, Consonni, Mauri, and Pavan, 2006).

6. See Hansen, Mika, Schroeter, Sutter, Laak, Steger-Hartmann, Heinrich, and Müller (2009) for results of modeling this set using different machine learning methods. The data itself is available online at `http://ml.cs.tu-berlin.de/toxbenchmark`

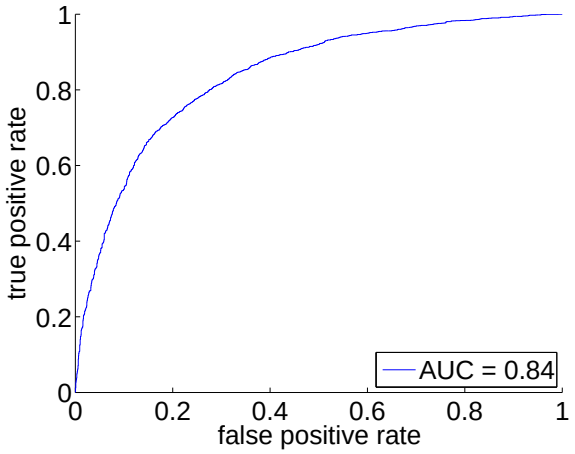


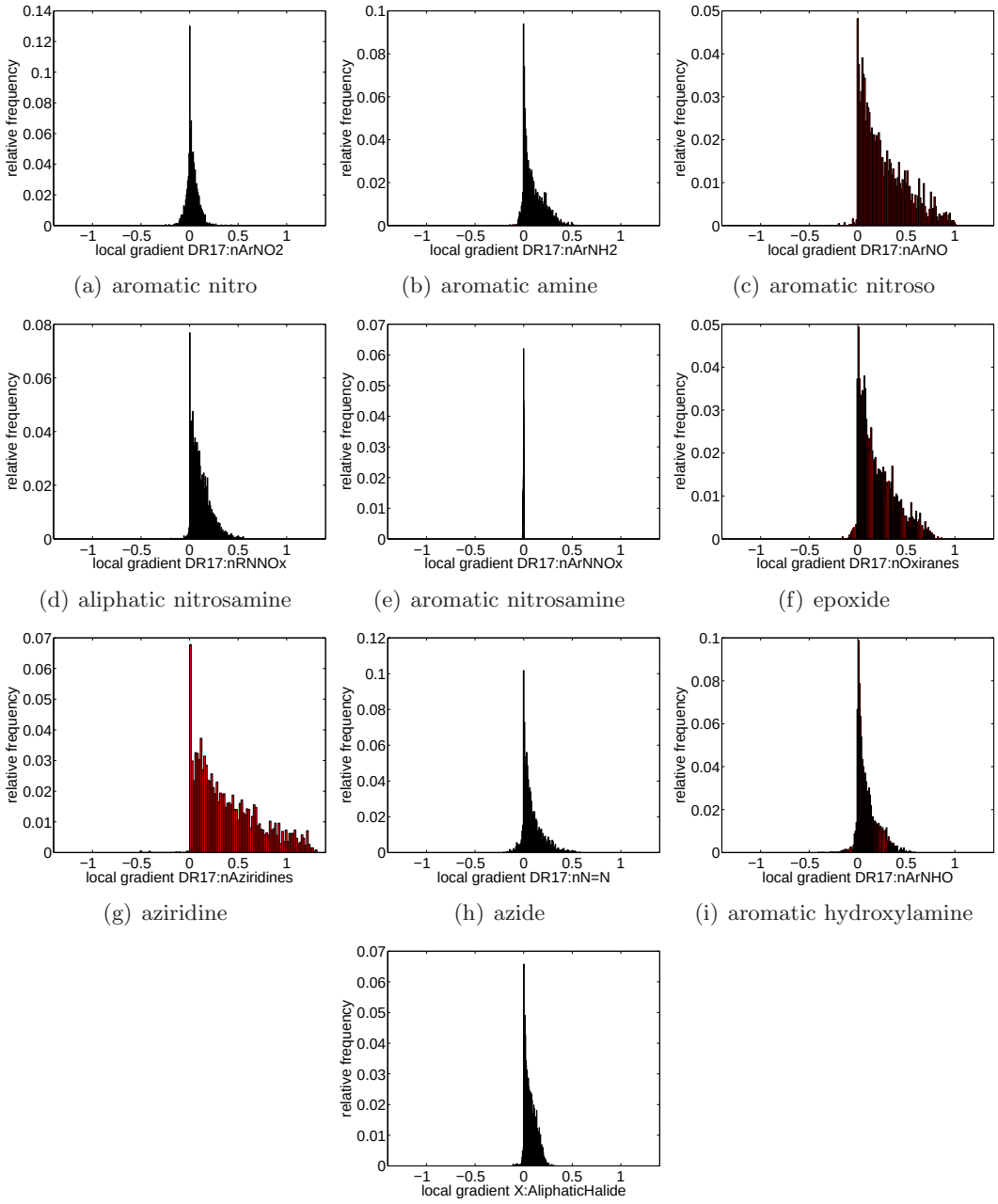
Figure 6: Receiver operating curve of GPC model for mutagenicity prediction

- Normalize training and test set using the mean and variance of the training set.
- Apply GPC model with RBF kernel
- Performance (84 % area under curve) confirms our previous results (Hansen, Mika, Schroeter, Sutter, Laa 2009). Error rates can be obtained from Figure 6.

Together with the prediction we calculated the explanation vector (as introduced in Section 2 with Definition 2) for each test point. The remainder of this Section is an evaluation of these local explanations.

In Figures 7 and 8 we show the distribution of the local importance of selected features across the test set: For each input feature we generate a histogram of local importance values, as indicated by its corresponding entry in the explanation vector of each of the 4512 test compounds. The features examined in Figure 7 are counts of substructures known to cause mutagenicity. We show all approved "specific toxicophores" introduced by Kazius, McGuire, and Bursi (2005) that are also represented in the Dragon set of features. The features shown in Figure 8 are known to detoxify certain toxicophores (again see Kazius, McGuire, and Bursi, 2005). With the exception of 7(e) the toxicophores also have a toxifying influence according to our GPC prediction model. Feature 7(e) seems to be mostly irrelevant for the prediction of the GPC model on the test points. In contrast the detoxicophores show overall negative influence on the prediction outcome of the GPC model. Modifying the test compounds by adding toxicophores will increase the probability of being mutagenic as predicted by the GPC model while adding detoxicophores will decrease this predicted probability.

So we have seen that the conclusions drawn from our explanation vectors agree with established knowledge about toxicophores and detoxicophores. While this is reassuring, such a sanity check required existing knowledge about which compounds are toxicophores and detoxicophores and which are not. Thus it is interesting to ask, whether we also could have *discovered* that knowledge from the explanation vectors. To answer this question



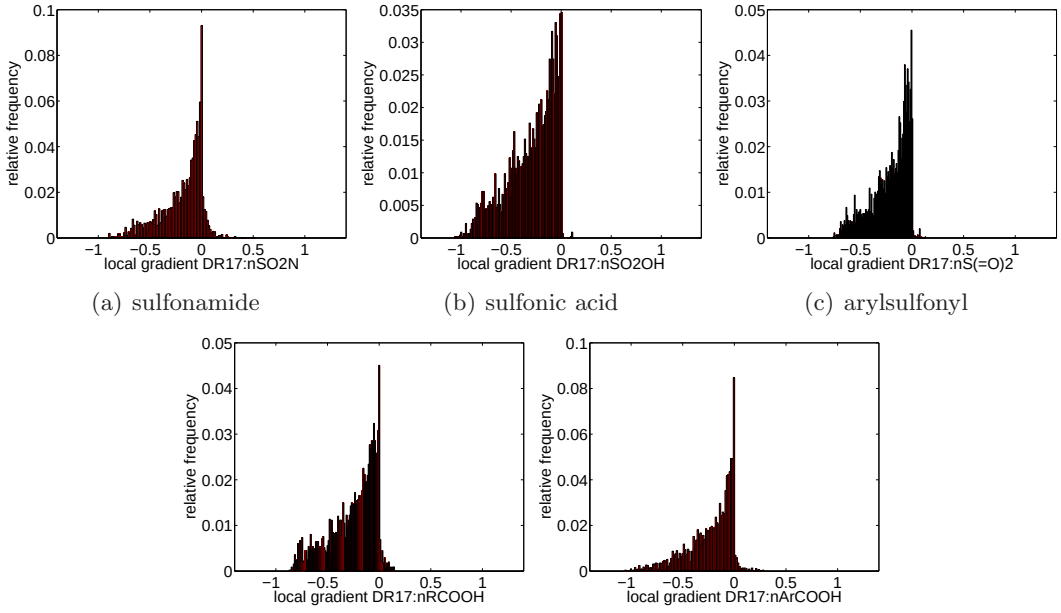
(a) aromatic nitro (b) aromatic amine (c) aromatic nitroso

(d) aliphatic nitrosamine (e) aromatic nitrosamine (f) epoxide

(g) aziridine (h) azide (i) aromatic hydroxylamine

(j) aliphatic halide

Figure 7: Distribution of local importance of selected features across the test set of 4512 compounds. Nine out of ten known toxicophores (Kazius, McGuire, and Bursi, 2005) indeed exhibit positive local gradients.



(a) sulfonamide (b) sulfonic acid (c) arylsulfonyl

(d) aliphatic carboxylic acid (e) aromatic carboxylic acid

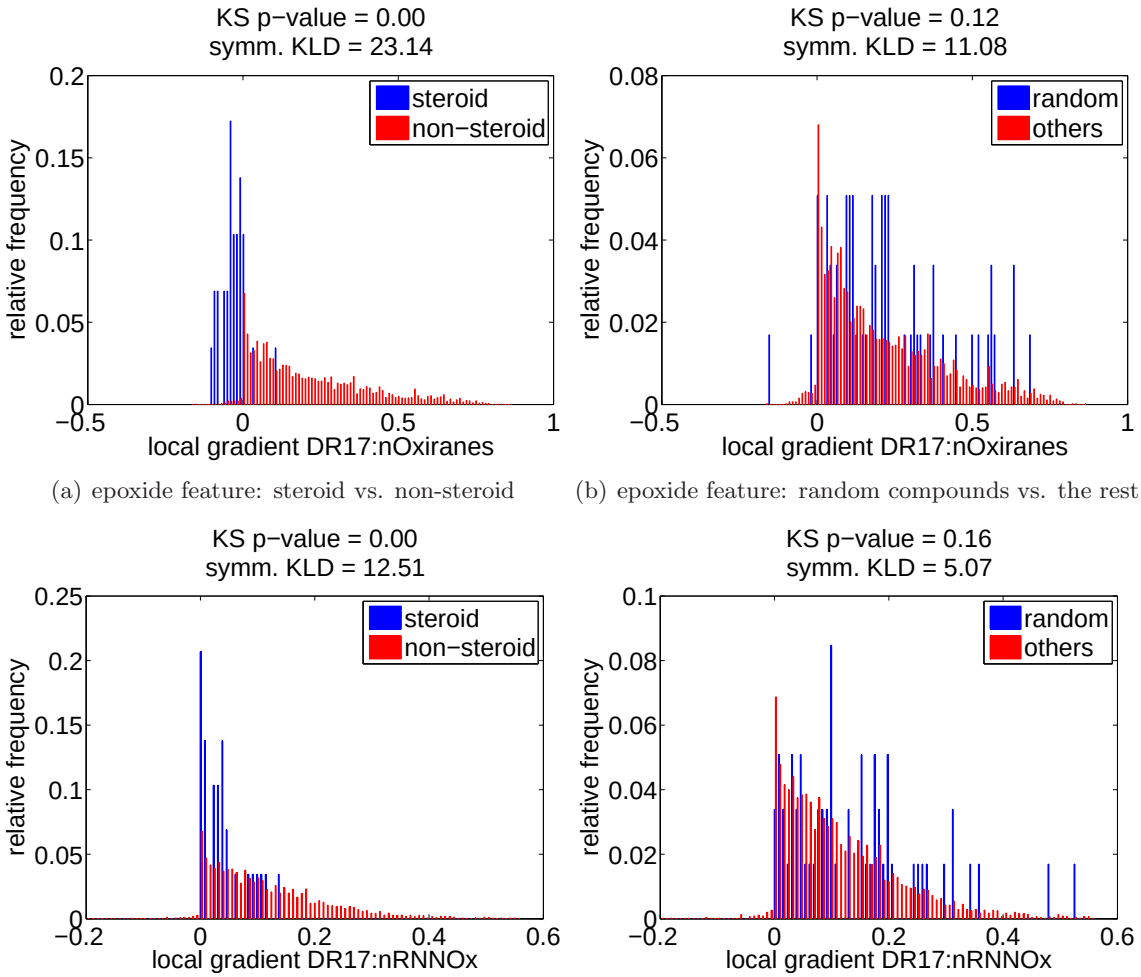
Figure 8: Distribution of local importance of selected features across the test set of 4512 compounds. All five known detoxicophores exhibit negative local gradients

we ranked all 142 features by the means of their local gradients[7]. Clear trends result: 9 out of 10 known toxicophores can be found close the top of the list (mean rank of 19). The only exception (rank 81) is the aromatic nitrosamine feature.[8] This trend is even stronger for the detoxicophores: The mean rank of these five features is 138 (out of 142), i.e. they consistently exhibit the largest negative local gradients. Consequently, the established knowledge about toxicophores and detoxicophores could indeed have been *discovered* using our methodology.

In the following paragraph we will discuss steroids[9] as an example of an important compound class for which the meaning of features differs from this global trend, so that local explanation vectors are needed to correctly identify relevant features.

Figure 9 displays the difference in relevance of epoxide (a) and aliphatic nitrosamine (c) substructures for the predicted mutagenicity of steroids and non-steroid compounds. For

7. Tables resulting from this ranking are made available as a supplement to this paper and can be downloaded from the journals website.
8. This finding agrees with the result obtained by visually inspecting Figure 7(e). We found that only very few compounds with this feature are present in the data set. Consequently, detection of this feature is only possible if enough of these few compounds are included in the training data. This was not the case in the random split used to produce the results presented above.
9. Steroids are natural products and occur in humans, animals and plants. They have a characteristic backbone containing four fused carbon-rings. Many hormones important to the development of the human body are steroids, including androgens, estrogens, progestagens, cholesterol and natural anabolics. These have been used as starting points for the development of many different drugs, including the most reliable contraceptives currently on the market.



(a) epoxide feature: steroid vs. non-steroid

(b) epoxide feature: random compounds vs. the rest

(c) aliphatic nitrosamine feature: steroid vs. non-steroid

(d) aliphatic nitrosamine feature: random compounds vs. the rest

Figure 9: The local distribution of feature importance to steroids and random non-steroid compounds significantly differs for two known toxicophores. The small local gradients found for the steroids (shown in blue) indicate that the presence of each toxicophore is irrelevant to the molecules toxicity. For non-steroids (shown in red) the known toxicophores indeed exhibit positive local gradients.

comparison we also show the distributions for compounds chosen at random from the test set (b,d). Each subfigure contains two measures of (dis-)similarity for each pair of distributions. The p-value of the Kolmogorov-Smirnoff test (KS) gives the probability of error when rejecting the hypothesis that both relative frequencies are drawn from the same underlying distribution. The symmetrized Kullback-Leibler divergence (KLD) gives a metric of the distance between the two distributions.[10] While containing epoxides generally tends to make molecules mutagenic (see discussion above), we do not observe this effect for steroids: In Figure 9(a), almost all epoxide containing non-steroids exhibit positive gradients, thereby following the global distribution of epoxide containing compounds as shown in Figure 7(f). In contrast, almost all epoxide containing steroids exhibit gradients just below zero. "Immunity" of steroids to the epoxide toxicophore is an established fact and has first been discussed by Glatt, Jung, and Oesch (1983). This peculiarity in chemical space is clearly exhibited by the local explanation given by our approach. For aliphatic nitrosamine, the situation in the GPC model is less clear but still the toxifying influence seems to be less in steroids than in many other compounds. To our knowledge, this phenomenon has not yet been discussed in the pharmaceutical literature.

In conclusion, we can learn from the explanation vectors that:

- toxicophores tend to make compounds mutagenic (class 1)
- detoxicophores tend to make compounds non-mutagenic (class 0)
- steroids are immune to the presence of some toxicophores (epoxide, possibly also aliphatic nitrosamine)

7. Related Work

Assigning potentially different explanations to individual data points distinguishes our approach from conventional feature extraction methods that extract global features that are relevant for all data points, i.e. those features that allow to achieve a small overall prediction error. Our notion of explanation is not related to the prediction error, but only to the label provided by the prediction algorithm. Even though the error is large, our framework is able to answer the question *why* the algorithm has decided on a data point the way it did.

The explanation vector proposed here is similar in spirit to sensitivity analysis which is common to various areas of information science. A classical example is the outlier sensitivity in statistics (Hampel, Ronchetti, Rousseeuw, and Stahel, 1986). In this case, the effects of removing single data points on estimated parameters are evaluated by an influence function. If the influence for a data point is significantly large, it is detected as an outlier and should be removed for the following analysis. In regression problems, leverage analysis is a procedure along similar lines. It detects leverage points which have potential to give large impact on the estimate of the regression function. In contrast to the influential points (outliers), removing a leverage sample may not actually change the regressor, if its response is very close to the predicted value. E.g. for linear regression the samples whose inputs are far from

10. Symmetry is achieved by averaging the two Kullback-Leibler divergences: $\frac{KL(P1,P2)+KL(P2,P1)}{2}$, cf. Johnson and Sinanovic (2000). To prevent zero-values in the histograms which would lead to infinite KL distances, an $\varepsilon > 0$ has been added to each bin count.

the mean are the leverage points. Our framework of explanation vectors considers a different view. It describes the influence of *moving* single data points locally and it thus answers the question which directions are locally most influential to the prediction. The explanation vectors are used for extracting sensitive features which are relevant to the prediction results, rather than detecting/eliminating the influential samples.

In recent decades, explanation of results by expert systems have been an important topic in the AI community. Especially, for those based on Bayesian belief networks, such explanation is crucial in practical use. In this context sensitivity analysis has also been used as a guiding principle (Horvitz, Breese, and Henrion, 1988). There the influence is evaluated by removing a set of variables (features) from evidences and the explanation is constructed from those variables which affect inference (relevant variables). For example, Suermondt (1992) measures the cost of omitting a single feature E_i by the cross-entropy

$$H^{-}(E_i) = H(p(D|E); P(D|E\backslash E_i)\,) = \sum_{j=1}^{N} P(d_j|E) \log \frac{P(d_j|E)}{p(d_j|E\backslash E_i)},$$

where E denotes evidences and $D = (d_1, \ldots, d_N)^T$ is the target variable. The cost of a subset $F \subset E$ can be defined similarly. This line of research is more connected to our work, because explanation can depend on the assigned values of the evidences E, and is thus local.

Similarly Robnik-Šikonja and Kononenko (2008) and Štrumbelj and Kononenko (2008) try to explain the decision of trained kNN-, SVM- and ANN-models for individual instances by measuring the difference in their prediction with sets of features omitted. The cost of omitting features is evaluated as the information difference, the log-odds ratio or the difference of probabilities between the model with knowledge about all features and with omissions respectively. To know what the prediction would be without the knowledge of a certain feature the model is retrained for every choice of features whose influence is to be explained. To save the time of combinatorial training Robnik-Šikonja and Kononenko (2008) propose to use neutral values which have to be estimated by a known prior distribution of all possible parameter values. As a theoretical framework for considering feature interactions, Štrumbelj and Kononenko (2008) propose to calculate the differences between model predictions for every choice of feature subset.

For multi-layer perceptrons Féraud and Clérot (2002) measure the importance of individual input variables on clusters of test points. Therefore the change in the model output is evaluated for the change of a single input variable in a chosen interval while all other input variables are fixed. Lemaire and Feraud (2007) use a similar approach on an instance by instance basis. By considering each input variable in turn there is no way to measure input feature interactions on the model output (see LeCun, Bottou, Orr, and Müller, 1998).

The principal differences between our approach and these frameworks are: (i) We consider continuous features and no structure among them is required, while some other frameworks start from binary features and may require discretization steps with the need to estimate parameters for it. (ii) We allow changes in any direction, i.e. any weighted combination of variables, while other approaches only consider one feature at a time or the omission of a set of variables.

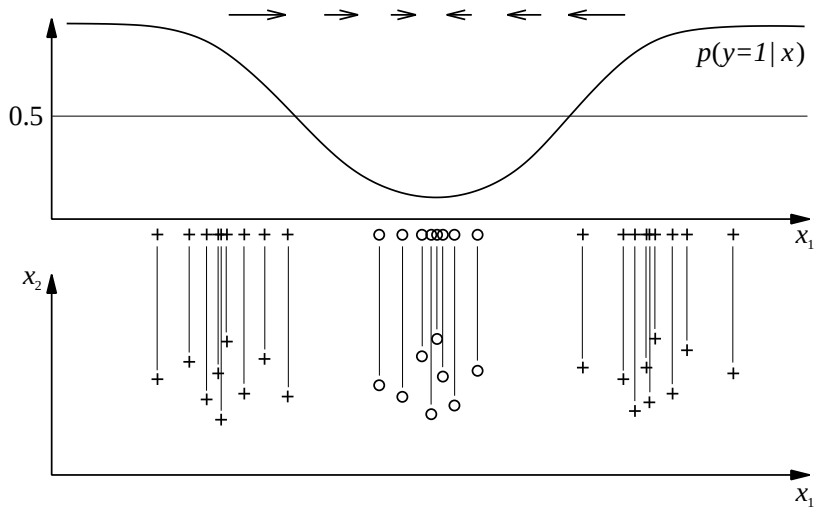


Figure 10: $\zeta(x)$ is the zero vector in the middle of the cluster in the middle.

8. Discussion

By now we have shown that our methods for calculating / estimating explanation vectors are useful in a variety of situations. In the following we discuss their limitations.

What can we do, if the derivative is zero? This situation is depicted in Figure 10. In the lower panel we see a two-dimensional data set consisting of three clusters. The middle cluster has a different class than the clusters on the left and on the right. Relevant for the classification is only the horizontal coordinate (i.e. x_1). The upper panel shows the projected data and a representative slice of $\zeta(x)$. However, the explanation $\zeta(x)$ for the center point of the middle cluster is the zero vector, because at that point $p(Y=1|X=x)$ is maximal. What can we do in such situations? Actually, the (normalized) explanation vector is derived from the following optimization problem for finding the locally most influential direction: $\mathrm{argmax}_{\|\varepsilon\|=1} \{p(Y \neq g^*(x_0)|X = x_0 + \varepsilon) - p(Y \neq g^*(x_0)|X = x_0)\}$. In case that the first derivative of the above criterion is zero, its Taylor expansion starts from the second order term, which is a quadratic form in its Hessian matrix. In the example data set with three clusters, the explanation vector is constant along the second dimension. The most *interesting* direction is given by the eigenvector corresponding to the largest eigenvalue of the Hessian. This direction will be in our example along the first dimension. Thus, we can learn from the Hessian that the first coordinate is relevant for the classification, but we do not obtain an orientation for it. Instead it means that both directions (left and right) will influence the classification. However, if the conditional distribution $P(Y = 1 \mid X = x)$ is flat in some regions, no meaningful explanation can be obtained by the gradient-based approach with the remedy mentioned above. Practically, by using Parzen window estimators with larger widths, the explanation vector can capture coarse structures of the classifier at the points which are not so far from the boarders. In A.2.2 we give an illustration of this point. In the future, we would like to work on global approaches, e.g. based on distances to the boarders, or extensions of the approach by Robnik-Šikonja and Kononenko (2008). Since

these procedures are expected to be computationally demanding, our proposal is useful in practice, in particular for probabilistic classifiers.

Does our framework generate different explanations for different prediction models? When using the local gradient of the model prediction directly as in Definition 2 and Section 6, the explanation follows the given model precisely by definition. For the estimation framework this depends on whether the different classifiers classify the data differently. In that case the explanation vectors will be different, which makes sense, since they should explain the classifier at hand, even if its estimated labels were not all correct. On the other hand, if the different classifiers agree on all labels, the explanation will be exactly equal.

Which implicit limitations do analytical gradients inherit from Gaussian Process models? A particular phenomenon can be observed at the boundaries of the training data: Far from the training data, Gaussian Process Classification models predict a probability of 0.5 for the positive class. When querying the model in an area of the feature space where predictions are negative, and one approaches the boundaries of the space populated with training data, explanation vectors will point away from any training data and therefore also away from areas of positive prediction. This behavior can be observed in Figure 1(d), where unit length vectors indicate the direction of explanation vectors. In the right hand side corner, arrows point away from the triangle. However, we can see that the length of these vectors is so small, that they are not even visible in Figure 1(c). Consequently, this property of GPC models does not pose a restriction for identifying the locally most influential features by investigating the features with the highest absolute values in the respective partial derivatives, as shown in Section 6.

Stationarity of the data. Since explanation vectors are defined as local gradients of the model prediction (see Definition 2), no assumption on the data is made: The local gradients follow the predictive model in any case. If, however, the model to be explained assumes stationarity of the data, the explanation vectors will inherit this limitation and reflect any shortcomings of the model (e.g. when the model is applied to non-stationary data). Our method for estimating explanation vectors, on the other hand, assumes stationarity of the data.

When modeling data that is in fact non-stationary, appropriate measures to deal with such data sets should be taken. One option is to separate the feature space into stationary and non-stationary parts using Stationary Subspace Analysis as introduced by von Bünau, Meinecke, Király, and Müller (2009). For further approaches to data set shift see Sugiyama, Nakajima, Kashima, von Buenau, and Kawanabe (2007b), Sugiyama, Krauledat, and Müller (2007a) and the book by Quionero-Candela, Sugiyama, Schwaighofer, and Lawrence (2009).

9. Conclusion

This paper proposes a method that sheds light into the black boxes of nonlinear classifiers. In other words, we introduce a method that can explain the local decisions taken by arbitrary (possibly) nonlinear classification algorithms. In a nutshell, the estimated explanations are local gradients that characterize how a data point has to be moved to change its predicted

label. For models where such gradient information cannot be calculated explicitly, we employ a probabilistic approximate mimic of the learning machine to be explained.

To validate our methodology we show how it can be used to draw new conclusions on how the various Iris flowers in Fisher's famous data set are different from each other and how to identify the features with which certain types of digits 2 and 8 in the USPS data set can be distinguished. Furthermore, we applied our method to a challenging drug discovery problem. The results on that data fully agree with existing domain knowledge, which was not available to our method. Even local peculiarities in chemical space (the extraordinary behavior of steroids) was discovered using the local explanations given by our approach.

Future directions are two-fold: First we believe that our method will find its way into the tool boxes of practitioners who not only want to automatically classify their data but who also would like to understand the learned classifier. Thus using our explanation framework in computation biology (see Sonnenburg, Zien, Philips, and Rätsch, 2008) and in decision making experiments in psychophysics (e.g. Kienzle, Franz, Schölkopf, and Wichmann, 2009) seems most promising. The second direction is to generalize our approach to other prediction problems such as regression.

Acknowledgments

This work was supported in part by the FP7-ICT Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-216886 and by DFG Grant MU 987/4-1. We would like to thank Andreas Sutter, Antonius Ter Laak, Thomas Steger-Hartmann and Nikolaus Heinrich for publishing the Ames mutagenicity data set (Hansen, Mika, Schroeter, Sutter, Laak, Stege 2009).

A. Appendix

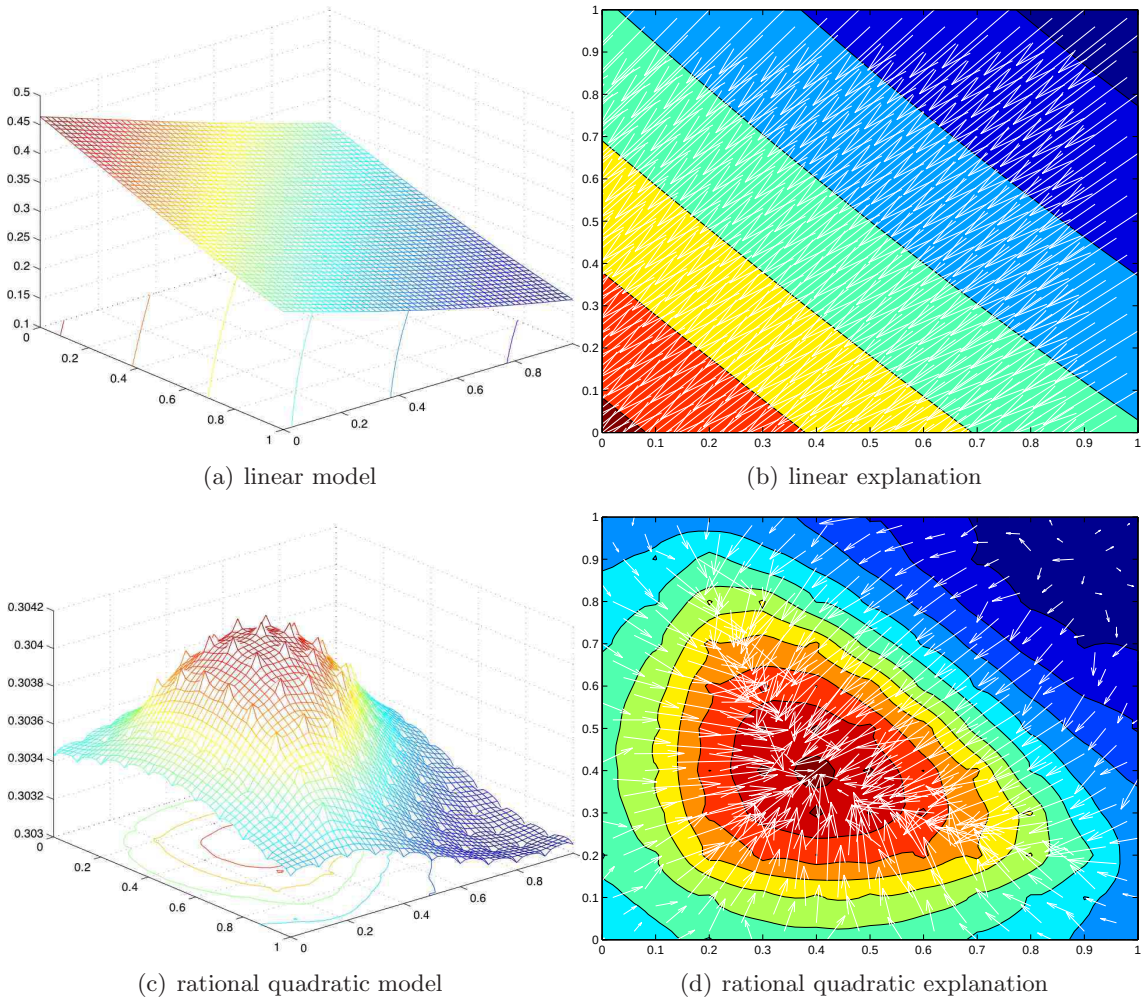
A.1 Illustration of direct local gradients

In the following we give some illustrative examples of our method to explain models using local gradients. Since the explanation is derived directly from the respective model, it is interesting to investigate its acuteness depending on different model parameters and in instructive scenarios. We examine the effects that local gradients exhibit when choosing different kernel functions, when introducing outliers, and when the classes are not linearly separable locally.

A.1.1 Choice of kernel function

Figure 11 shows the effect of different kernel functions on the triangle toy data from Figure 1 in Section 2. The following observations can be made:

- In any case note that the local gradients explain the model, which in turn may or may not capture the true situation.
- In Subfigure 11(a) the linear kernel leads to a model which fails to capture the non-linear class separation. This model misspecification is reflected by the explanations given for this model in Subfigure 11(b).



(a) linear model

(b) linear explanation

(c) rational quadratic model

(d) rational quadratic explanation

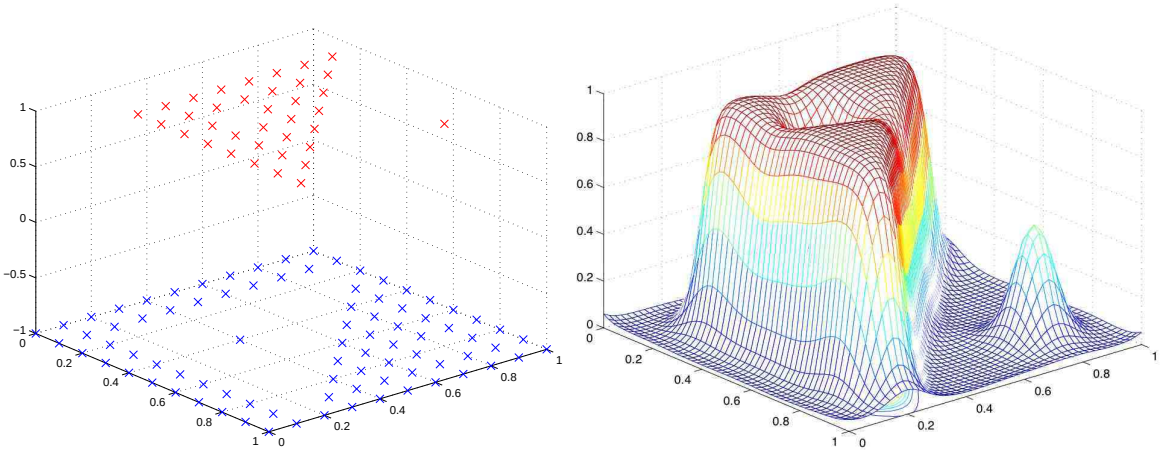
Figure 11: The effect of different kernel functions to the local gradient explanations

- The rational quadratic kernel is able to more accurately model the non-linear separation. In Subfigure 11(c) a non-optimal degree parameter has been chosen for illustrative purposes. For other parameter values the rational quadratic kernel leads to similar results as the RBF kernel function used in Figure 1.

- The explanations in Subfigure 11(d) obtained for this model show local perturbations at the small "bumps" of the model but the trends towards the positive class are still clear. As previously observed in Figure 1, the explanations make clear that both features interact at the corners and on the hypotenuse of the triangle class.

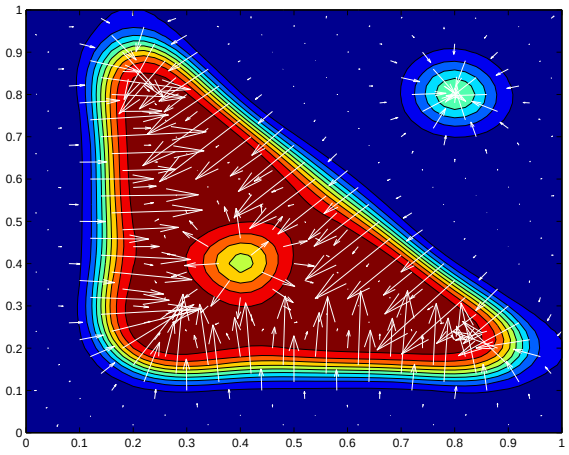
A.1.2 Outliers

In Figure 12 the effects of two outliers in the classification data to GPC with RBF kernel are shown. Once more, note that the local gradients explain the model, which in turn may or



(a) outliers in classes

(b) outliers in model



(c) outlier explanation

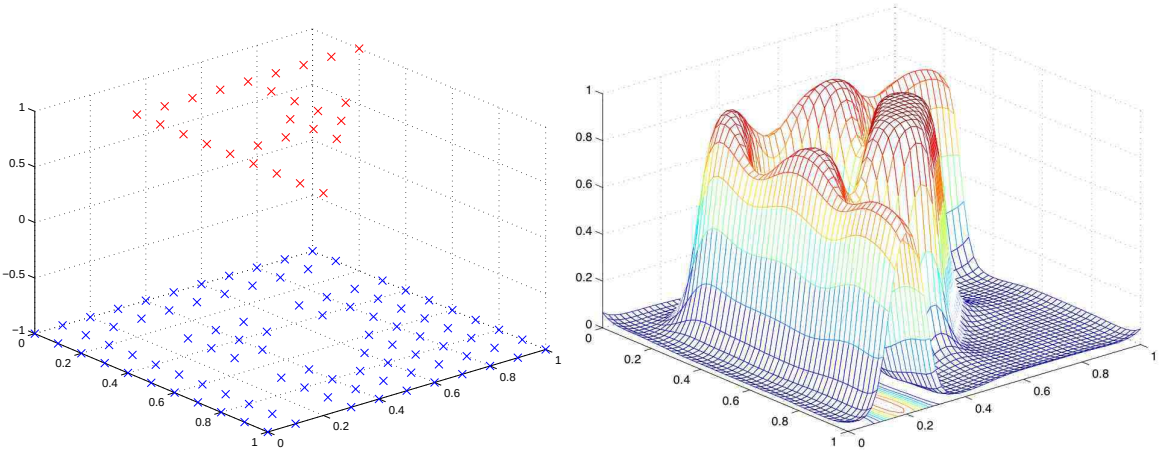
Figure 12: The effect of outliers to the local gradient explanations

may not capture the true situation. The size of the region affected by the outliers depends on the kernel width parameter. We consider the following items:

- Local gradients are in the same way sensitive to outliers as the model which they try to explain. Here a single outlier deforms the model and with it the explanation which may be extracted from it.
- Being derivatives the sensitivity of local gradients to a nearby outlier is increased over the sensitivity of the model prediction itself.
- Thus the local gradient of a point near an outlier may not reflect a true explanation of the features important in reality. Nevertheless it is the model here which is wrong around an outlier in the first place.
- The histograms in the Figures 7, 8, and 9 in Section 6 show the trends of the respective features in the distribution of all test points and are thus not affected by single outliers.

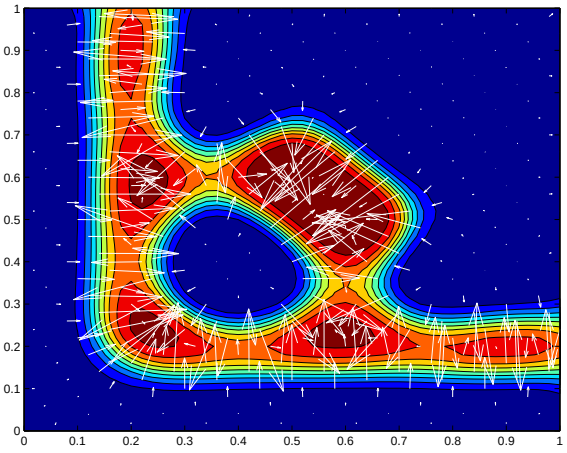
To compensate for the effect of outliers to the local gradients of points in the affected region we propose to use a sliding window method to smooth the gradients around each point of interest. Thus for each point use the mean of all local gradients in the hypercube centered at this point and of appropriate size. This way the disrupting effect of an outlier is averaged out for an appropriately chosen window size.

A.1.3 Local non-linearity



(a) locally non-linear object

(b) locally non-linear model



(c) locally non-linear explanation

Figure 13: The effect of local non-linearity to the local gradient explanations

The effect of locally non-linear class boundaries in the data is shown in Figure 13 again for GPC with an RBF kernel. The following points can be observed:

- All the non-linear class boundaries are accurately followed by the local gradients
- The circle shaped region of negative examples surrounded by positive ones shows the full range of feature interactions towards the positive class

- On the ridge of single positive instances the model introduces small valleys which are reflected by the local gradients

A.2 Estimating by Parzen window

Finally we elaborate on some details of our estimation approach of local gradients by Parzen window approximation. First we give the derivation to obtain the explanation vector and second we examine how the explanation varies with the goodness of fit of the Parzen window method.

A.2.1 Derivation of explanation vectors

These are more details on the derivation of Eq. (3). We use the index set $I_c = \{i \mid g(x_i) = c\}$:

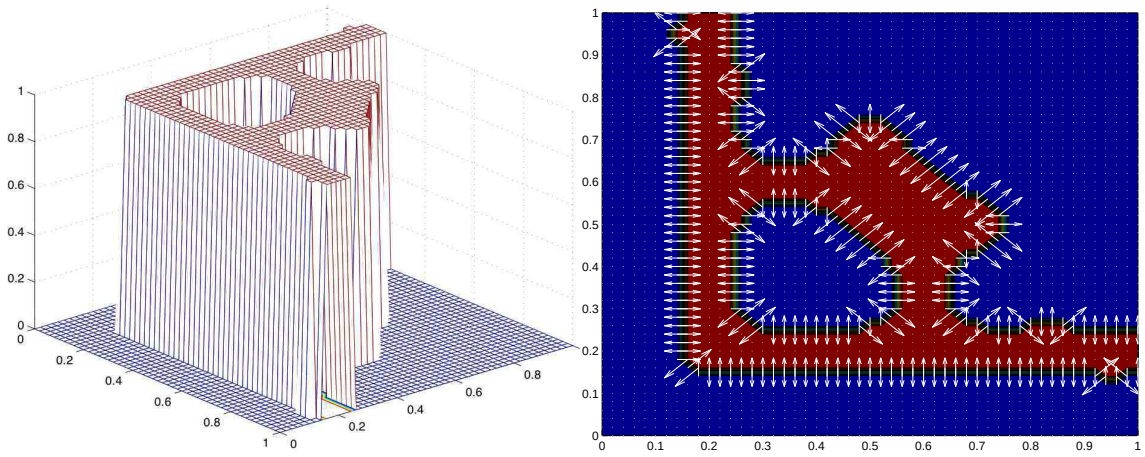
$$
\begin{aligned}
&\frac{\partial}{\partial x} k_\sigma(x) = -\frac{x}{\sigma^2} k_\sigma(x) \\
&\frac{\partial}{\partial x} \hat{p}_\sigma(x, y \neq c) = \frac{1}{n} \sum_{i \notin I_c} k_\sigma(x - x_i) \frac{-(x - x_i)}{\sigma^2} \\
&\frac{\partial}{\partial x} \hat{p}_\sigma(y \neq c | x) \\
&= \frac{\Big(\sum_{i \notin I_c} k(x - x_i)\Big)\Big(\sum_{i=1}^n k(x - x_i)(x - x_i)\Big)}{\sigma^2 \left(\sum_{i=1}^n k(z - x_i)\right)^2} \\
&- \frac{\Big(\sum_{i \notin I_c} k(x - x_i)(x - x_i)\Big)\Big(\sum_{i=1}^n k(x - x_i)\Big)}{\sigma^2 \left(\sum_{i=1}^n k(z - x_i)\right)^2} \\
&= \frac{\Big(\sum_{i \notin I_c} k(x - x_i)\Big)\Big(\sum_{i \in I_c} k(x - x_i)(x - x_i)\Big)}{\sigma^2 \left(\sum_{i=1}^n k(z - x_i)\right)^2} \\
&- \frac{\Big(\sum_{i \notin I_c} k(x - x_i)(x - x_i)\Big)\Big(\sum_{i \in I_c} k(x - x_i)\Big)}{\sigma^2 \left(\sum_{i=1}^n k(z - x_i)\right)^2}
\end{aligned}
$$

and thus for the index set $I_{g(z)} = \{i \mid g(x_i) = g(z)\}$

$$
\begin{aligned}
\hat{\zeta}(z) &= \frac{\partial}{\partial x} \hat{p}(y \neq g(z) \mid x)\Big|_{x=z} \\
&= \frac{\Big(\sum_{i \notin I_{g(z)}} k(z - x_i)\Big)\Big(\sum_{i \in I_{g(z)}} k(z - x_i)(z - x_i)\Big)}{\sigma^2 \left(\sum_{i=1}^n k(z - x_i)\right)^2} \\
&- \frac{\Big(\sum_{i \notin I_{g(z)}} k(z - x_i)(z - x_i)\Big)\Big(\sum_{i \in I_{g(z)}} k(z - x_i)\Big)}{\sigma^2 \left(\sum_{i=1}^n k(z - x_i)\right)^2}
\end{aligned}
$$

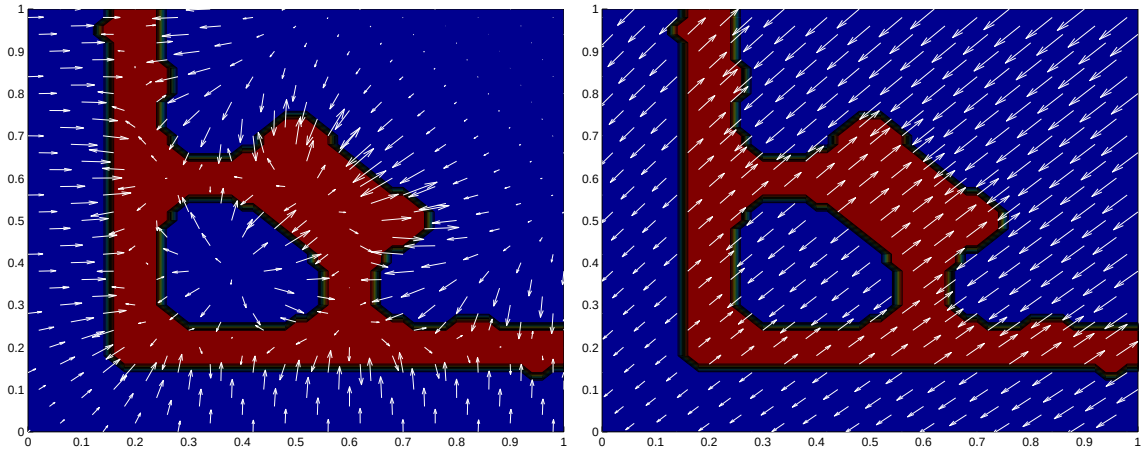
A.2.2 Goodness of fit by Parzen window

In our estimation framework the quality of the local gradients depends on the approximation of the classifier we want to explain by Parzen windows for which we can calculate the explanation vectors as given by Definition 3.



(a) SVM model

(b) estimated explanation with $\sigma = 0.00069$



(c) estimated explanation with $\sigma = 0.1$

(d) estimated explanation with $\sigma = 1.0$

Figure 14: Good fit of Parzen window approximation affects the quality of the estimated explanation vectors

Figure 14(a) shows an SVM model trained on the classification data from Figure 13(a). The local gradients estimated for this model by different Parzen window approximations are depicted in Subfigures 14(b), 14(c), and 14(d). We observe the following points:

- The SVM model was trained with $C = 10$ and using an RBF kernel of width $\sigma = 0.01$
- In Subfigure 14(b) a small window width has been chosen by minimizing the mean absolute error over the validation set of labels predicted by the SVM classifier. Thus we

obtain explaining local gradients on the class boundaries but zero vectors in the inner class regions. While this resembles the piecewise flat SVM model most accurately it may be more useful practically to choose a larger width to obtain non-zero gradients pointing to the borders in this regions as well. For a more detailed discussion of zero gradients see Section 8.

- A larger width practically useful in this example is shown in Subfigure 14(c). Here the local gradients in the inner class regions point to the other class as well.
- For a too large window width in Subfigure 14(d) the approximation fails to obtain local gradients which closely follow the model. Here only two directions are left and the gradients for the blue class on the left and on the bottom point in the wrong direction.

References

Bruce N. Ames, E. G. Gurney, James A. Miller, and H. Bartsch. Carcinogens as Frameshift Mutagens: Metabolites and Derivatives of 2-Acetylaminofluorene and Other Aromatic Amine Carcinogens. *Proceedings of the National Academy of Sciences of the United States of America*, 69(11):3128–3132, 1972. URL `http://www.pnas.org/content/69/11/3128.abstract`.

C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of Mathematics. Springer, New York, 1996.

R.A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.

Raphael Féraud and Fabrice Clérot. A methodology to explain neural network classification. *Neural Networks*, 15(2):237 – 246, 2002. ISSN 0893-6080. doi: DOI:10.1016/S0893-6080(01)00127-7. URL `http://www.sciencedirect.com/science/article/B6T08-4441WFN-5/2/d097075076605aa08026f96410`

Hansruedi Glatt, Reinhard Jung, and Franz Oesch. Bacterial mutagenicity investigation of epoxides: drugs, drug metabolites, steroids and pesticides. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 111(2):99–118, 1983. ISSN 0027-5107. doi: DOI:10.1016/0027-5107(83)90056-8. URL `http://dx.doi.org/10.1016/0027-5107(83)90056-8`.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York, 1986.

Katja Hansen, Sebastian Mika, Timon Schroeter, Andreas Sutter, Antonius Ter Laak, Thomas Steger-Hartmann, Nikolaus Heinrich, and Klaus-Robert Müller. A benchmark

data set for in silico prediction of ames mutagenicity. *Journal of Chemical Information and Modelling*, 49(9):2077–2081, 2009. URL http://dx.doi.org/10.1021/ci900161g.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

E. J. Horvitz, J. S. Breese, and M. Henrion. Decision theory in expert systems and artificial interigence. *Journal of Approximation Reasoning*, 2:247–302, 1988. Special Issue on Uncertainty in Artificial Intelligence.

Don H. Johnson and Sinan Sinanovic. Symmetrizing the Kullback-Leibler distance. Technical report, IEEE Transactions on Information Theory, 2000.

Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *J. Med. Chem.*, 48:312–320, 2005.

W. Kienzle, M. O. Franz, B. Schölkopf, and F. A. Wichmann. Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of Vision*, 9(5):1–15, 2009.

M. Kuss and C. E. Ramussen. Assesing approximate inference for bianry gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

Y. LeCun, L. Bottou, G.B. Orr, and K.-R. Müller. Efficient backprop. In G.B. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the trade*, pages 9–53. Springer, 1998.

Vincent Lemaire and Raphaël Feraud. Une méthode d'interprétation de scores. In Monique Noirhomme-Fraiture and Gilles Venturini, editors, *EGC*, volume RNTI-E-9 of *Revue des Nouvelles Technologies de l'Information*, pages 191–192. Cépaduès-Éditions, 2007. ISBN 978-2-85428-763-9.

K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *Neural Networks, IEEE Transactions on*, 12(2):181–201, 2001.

Olga Obrezanova, Gábor Csányi, Joelle M.R. Gola, and Matthew D. Segall. Gaussian processes: A method for automatic QSAR modelling of adme properties. *J. Chem. Inf. Model.*, 47(5):1847–1857, 2007. URL http://dx.doi.org/10.1021/ci7000633.

Olga Obrezanova, Joelle M. R. Gola, Edmund J. Champness, and Matthew D. Segall. Automatic QSAR modeling of adme properties: blood-brain barrier penetration and aqueous solubility. *J. Comput.-Aided Mol. Des.*, 22:431–440, 2008. URL http://dx.doi.org/10.1007/s10822-008-9193-8.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051, 9780262170055.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Springer, 2006.

Marko Robnik-Šikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE TKDE*, 20(5):589–600, 2008.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT, 2002.

Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Suelzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Estimating the domain of applicability for machine learning QSAR models: A study on aqueous solubility of drug discovery molecules. *Journal of Computer Aided Molecular Design - special issue on "ADME and Physical Properties"*, 21(9):485–498, 2007a. URL http://dx.doi.org/10.1007/s10822-007-9125-z.

Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Suelzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Machine learning models for lipophilicity and their domain of applicability. *Mol. Pharm.*, 4(4):524–538, 2007b. URL http://dx.doi.org/10.1021/mp0700413.

Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius ter Laak, Detlev Sülzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Predicting lipophilicity of drug discovery molecules using gaussian process models. *ChemMedChem*, 2(9):1265–1267, 2007c. URL http://dx.doi.org/10.1002/cmdc.200700041.

A. Schwaighofer. SVM Toolbox for Matlab, Jan 2002. URL http://ida.first.fraunhofer.de/~anton/software.html.

Anton Schwaighofer, Timon Schroeter, Sebastian Mika, Julian Laub, Antonius ter Laak, Detlev Sülzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Accurate solubility prediction with error bars for electrolytes: A machine learning approach. *Journal of Chemical Information and Modelling*, 47(2):407–424, 2007. URL http://dx.doi.org/10.1021/ci600205g.

Anton Schwaighofer, Timon Schroeter, Sebastian Mika, Katja Hansen, Antonius ter Laak, Philip Lienau, Andreas Reichel, Nikolaus Heinrich, and Klaus-Robert Müller. A probabilistic approach to classifying metabolic stability. *Journal of Chemical Information and Modelling*, 48(4):785–796, 2008. URL http://dx.doi.org/10.1021/ci700142c.

Sören Sonnenburg, Alexander Zien, Petra Philips, and Gunnar Rätsch. POIMs: positional oligomer importance matrices — understanding support vector machine based signal detectors. *Bioinformatics*, 2008. (received the Best Student Paper Award at ISMBÓ8).

H. Suermondt. *Explanation in Bayesian Belief Networks*. PhD thesis, Department of Computer Science and Medicine, Stanford University, Stanford, CA, 1992.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8: 985–1005, May 2007a.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems 20*. MIT Press, 2007b.

R. Todeschini, V. Consonni, A. Mauri, and M. Pavan. Dragon for windows and linux 2006. `http://www.talete.mi.it/help/dragon_help/` (accessed 27 March 2009), 2006.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

Paul von Bünau, Frank C Meinecke, Franz J Király, and Klaus-Robert Müller. Finding stationary subspaces in multivariate time series. *Physical review letters*, 103(21):214101, 2009. doi: 10.1103/PhysRevLett.103.214101. URL `http://link.aps.org/abstract/PRL/v103/e214101`.

Erik Štrumbelj and Igor Kononenko. Towards a model independent method for explaining classification for individual instances. In I.-Y. Song, J. Eder, and T.M. Nguyen, editors, *Data Warehousing and Knowledge Discovery*, volume 5182 of *Lecture Notes in Computer Science*, pages 273–282. Springer, 2008.