Chapter 2

Basics of Machine Learning

The conventional approach to solve problems with the help of computers is to write programs which solve the problem. In this approach the programmer must understand the problem, find a solution appropriate for the computer, and implement this solution on the computer. We call this approach *deductive* because the human deduces the solution from the problem formulation. However in biology, chemistry, biophysics, medicine, and other life science fields a huge amount of data is produced which is hard to understand and to interpret by humans. A solution to a problem may also be found by a machine which learns. Such a machine processes the data and automatically finds structures in the data, i.e. learns. The knowledge about the extracted structure can be used to solve the problem at hand. We call this approach *inductive*, Machine learning is about inductively solving problems by machines, i.e. computers.

Researchers in machine learning construct algorithms that automatically improve a solution to a problem with more data. In general the quality of the solution increases with the amount of problem-relevant data which is available.

Problems solved by machine learning methods range from classifying observations, predicting values, structuring data (e.g. clustering), compressing data, visualizing data, filtering data, selecting relevant components from data, extracting dependencies between data components, modeling the data generating systems, constructing noise models for the observed data, integrating data from different sensors,

Using classification a diagnosis based on the medical measurements can be made or proteins can be categorized according to their structure or function. Predictions support the current action through the knowledge of the future. A prominent example is stock market prediction but also prediction of the outcome of therapy helps to choose the right therapy or to adjust the doses of the drugs. In genomics identifying the relevant genes for a certain investigation (gene selection) is important for understanding the molecular-biological dynamics in the cell. Especially in medicine the identification of genes related to cancer draw the attention of the researchers.

2.1 Machine Learning in Bioinformatics

Many problems in bioinformatics are solved using machine learning techniques.

Machine learning approaches to bioinformatics include:

- Protein secondary structure prediction (neural networks, support vector machines)

- Gene recognition (hidden Markov models)
- Multiple alignment (hidden Markov models, clustering)
- Splice site recognition (neural networks)
- Microarray data: normalization (factor analysis)
- Microarray data: gene selection (feature selection)
- Microarray data: prediction of therapy outcome (neural networks, support vector machines)
- Microarray data: dependencies between genes (independent component analysis, clustering)
- Protein structure and function classification (support vector machines, recurrent networks)
- Alternative splice site recognition (SVMs, recurrent nets)
- Prediction of nucleosome positions
- Single nucleotide polymorphism (SNP) detection
- Peptide and protein array analysis
- Systems biology and modeling

For the last tasks like SNP data analysis, peptide or protein arrays and systems biology new approaches are developed currently.

For protein 3D structure prediction machine learning methods outperformed "threading" methods in template identification (Cheng and Baldi, 2006).

Threading was the golden standard for protein 3D structure recognition if the structure is known (almost all structures are known).

Also for alternative splice site recognition machine learning methods are superior to other methods (Gunnar Rätsch).

2.2 Introductory Example

In the following we will consider a classification problem taken from "Pattern Classification", Duda, Hart, and Stork, 2001, John Wiley & Sons, Inc. In this classification problem salmons must be distinguished from sea bass given pictures of the fishes. Goal is that an automated system is able to separate the fishes in a fish-packing company, where salmons and sea bass are sold. We are given a set of pictures where experts told whether the fish on the picture is salmon or sea bass. This set, called *training set*, can be used to construct the automated system. The objective is that future pictures of fishes can be used to automatically separate salmon from sea bass, i.e. to classify the fishes. Therefore, the goal is to correctly classify the fishes in the future on unseen data. The performance on future novel data is called *generalization*. Thus, our goal is to maximize the generalization performance.

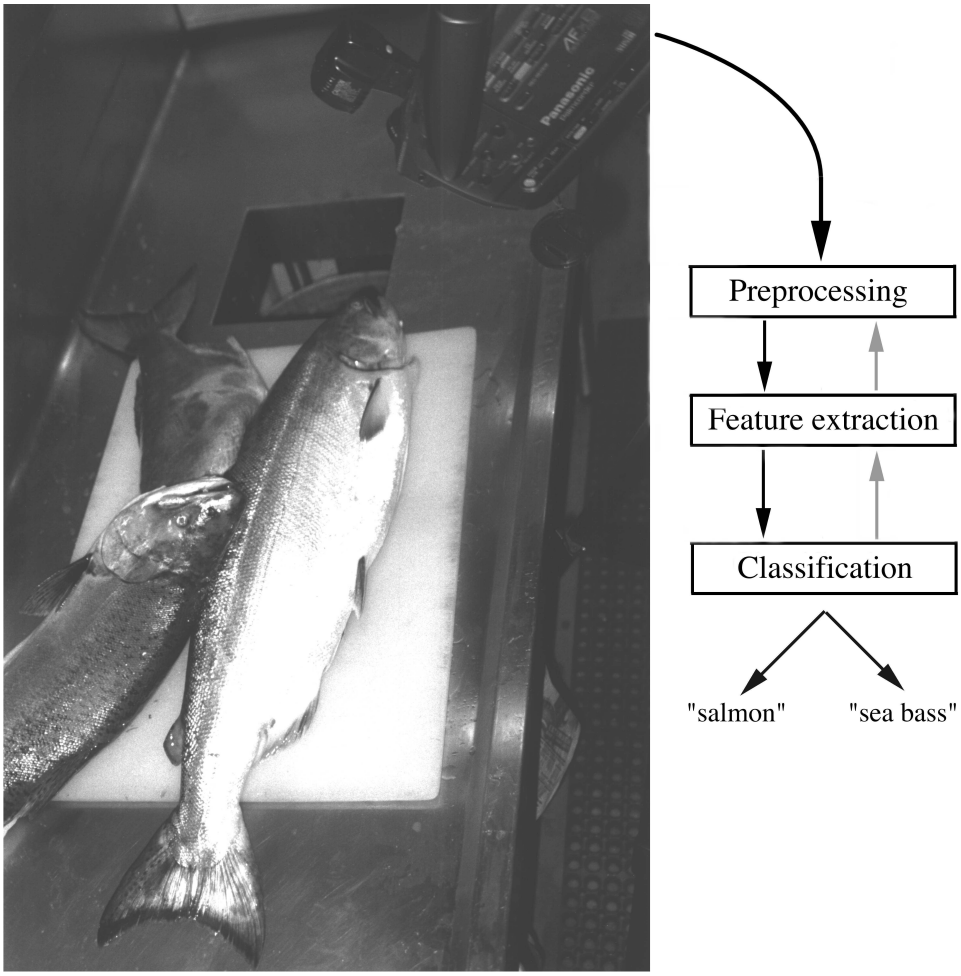

Figure 2.1: Salmons must be distinguished from sea bass. A camera takes pictures of the fishes and these pictures have to be classified as showing either a salmon or a sea bass. The pictures must be preprocessed and features extracted whereafter classification can be performed.

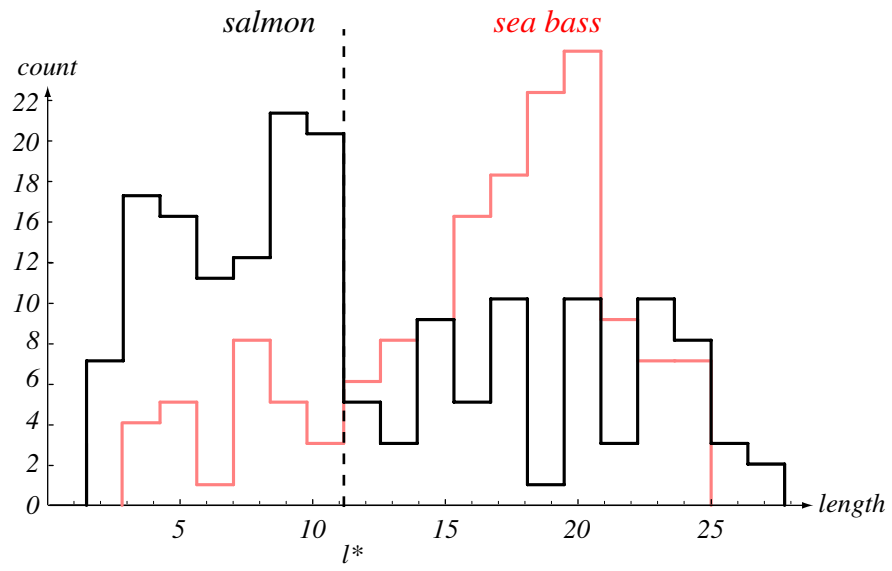

Figure 2.2: Salmon and sea bass are separated by their length. Each vertical line gives a decision boundary l, where fish with length smaller than l are assumed to be salmon and others as sea bass. l^* gives the vertical line which will lead to the minimal number of misclassifications. Copyright © 2001 John Wiley & Sons, Inc.

Before the classification can be done the pictures must be preprocessed and features extracted. Classification is performed with the extracted features. See Fig. 2.1.

The preprocessing might involve contrast and brightness adjustment, correction of a brightness gradient in the picture, and segmentation to separate the fish from other fishes and from the background. Thereafter the single fish is aligned, i.e. brought in a predefined position. Now features of the single fish can be extracted. Features may be the length of the fish and its lightness.

First we consider the length in Fig. 2.2. We chose a decision boundary l, where fish with length smaller than l are assumed to be salmon and others as sea bass. The optimal decision boundary l^* is the one which will lead to the minimal number of misclassifications.

The second feature is the lightness of the fish. A histogram if using only this feature to decide about the kind of fish is given in Fig. 2.3.

For the optimal boundary we assumed that each misclassification is equally serious. However it might be that selling sea bass as salmon by accident is more serious than selling salmon as sea bass. Taking this into account we would chose a decision boundary which is on the left hand side of x^* in Fig. 2.3. Thus the cost function governs the optimal decision boundary.

As third feature we use the width of the fishes. This feature alone may not be a good choice to separate the kind of fishes, however we may have observed that the optimal separating lightness value depends on the width of the fishes. Perhaps the width is correlated with the age of the fish and the lightness of the fishes change with age. It might be a good idea to combine both features. The result is depicted in Fig. 2.4, where for each width an optimal lightness value is given. The optimal lightness value is a linear function of the width.

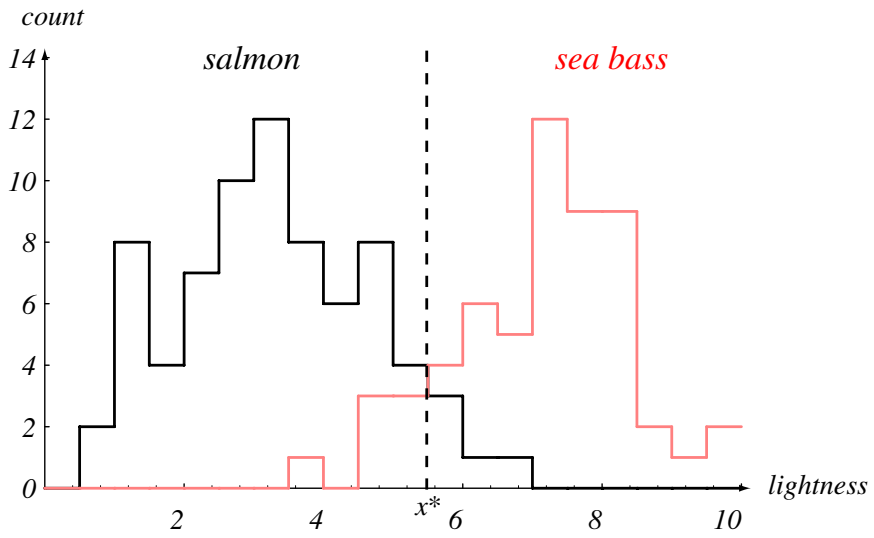

Figure 2.3: Salmon and sea bass are separated by their lightness. x^* gives the vertical line which will lead to the minimal number of misclassifications. Copyright © 2001 John Wiley & Sons, Inc.

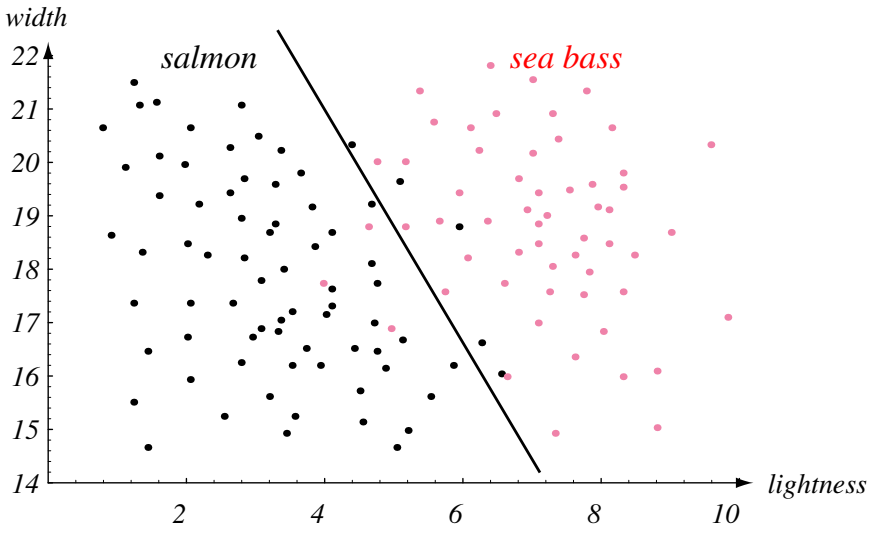

Figure 2.4: Salmon and sea bass are separated by their lightness and their width. For each width there is an optimal separating lightness value given by the line. Here the optimal lightness is a linear function of the width. Copyright © 2001 John Wiley & Sons, Inc.

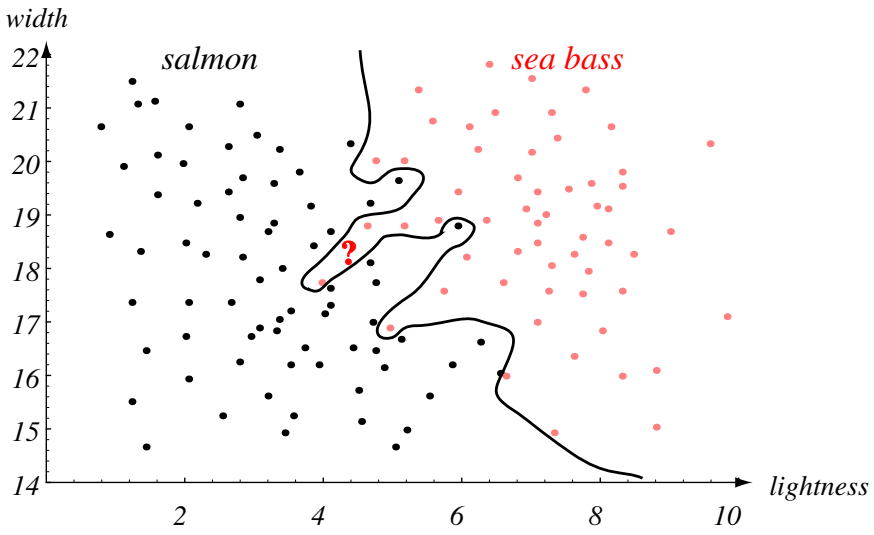

Figure 2.5: Salmon and sea bass are separated by a nonlinear curve in the two-dimensional space spanned by the lightness and the width of the fishes. The training set is separated perfectly. A new fish with lightness and width given at the position of the question mark "?" would be assumed to be sea bass even if most fishes with similar lightness and width were previously salmon. Copyright © 2001 John Wiley & Sons, Inc.

Can we do better? The optimal lightness value may be a nonlinear function of the width or the optimal boundary may be a nonlinear curve in the two-dimensional space spanned by the lightness and the width of the fishes. The later is depicted in Fig. 2.5, where the boundary is chosen that every fish is classified correctly on the training set. A new fish with lightness and width given at the position of the question mark "?" would be assumed to be sea bass. However most fishes with similar lightness and width were previously classified as salmon by the human expert. At this position we assume that the generalization performance is low. One sea bass, an outlier, has lightness and width which are typically for salmon. The complex boundary curve also catches this outlier however must assign space without fish examples in the region of salmons to sea bass. We assume that future examples in this region will be wrongly classified as sea bass. This case will later be treated under the terms *overfitting*, *high variance*, *high model complexity*, and *high structural risk*.

A decision boundary, which may represent the boundary with highest generalization, is shown in Fig. 2.6.

In this classification task we selected the features which are best suited for the classification. However in many bioinformatics applications the number of features is large and selecting the best feature by visual inspections is impossible. For example if the most indicative genes for a certain cancer type must be chosen from 30,000 human genes. In such cases with many features describing an object *feature selection* is important. Here a machine and not a human selects the features used for the final classification.

Another issue is to construct new features from given features, i.e. *feature construction*. In above example we used the width in combination with the lightness, where we assumed that

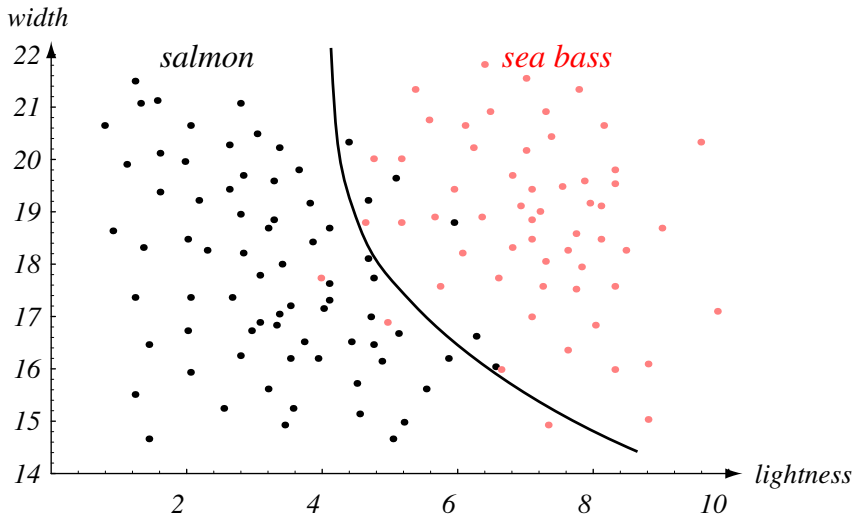

Figure 2.6: Salmon and sea bass are separated by a nonlinear curve in the two-dimensional space spanned by the lightness and the width of the fishes. The curve may represent the decision boundary leading to the best generalization. Copyright © 2001 John Wiley & Sons, Inc.

the width indicates the age. However, first combining the width with the length may give a better estimate of the age which thereafter can be combined with the lightness. In this approach averaging over width and length may be more robust to certain outliers or to errors in processing the original picture. In general redundant features can be used in order to reduce the noise from single features.

Both feature construction and feature selection can be combined by randomly generating new features and thereafter selecting appropriate features from this set of generated features.

We already addressed the question of cost. That is how expensive is a certain error. A related issue is the kind of noise on the measurements and on the class labels produced in our example by humans. Perhaps the fishes on the wrong side of the boundary in Fig. 2.6 are just error of the human experts. Another possibility is that the picture did not allow to extract the correct lightness value. Finally, outliers in lightness or width as in Fig. 2.6 may be typically for salmons and sea bass.

2.3 Supervised and Unsupervised Learning

In the previous example a human expert characterized the data, i.e. supplied the label (the class). Tasks, where the desired output for each object is given, are called *supervised* and the desired outputs are called *targets*. This term stems from the fact that during learning a model can obtain the correct value from the teacher, the supervisor.

If data has to be processed by machine learning methods, where the desired output is not given, then the learning task is called *unsupervised*. In a supervised task one can immediately measure how good the model performs on the training data, because the optimal outputs, the tar-

gets, are given. Further the measurement is done for each single object. This means that the model supplies an error value on each object. In contrast to supervised problems, the quality of models on unsupervised problems is mostly measured on the cumulative output on all objects. Typically measurements for unsupervised methods include the information contents, the orthogonality of the constructed components, the statistical independence, the variation explained by the model, the probability that the observed data can be produced by the model (later introduced as *likelihood*), distances between and within clusters, etc.

Typical fields of supervised learning are classification, regression (assigning a real value to the data), or time series analysis (predicting the future). An examples for regression is to predict the age of the fish from above examples based on length, width and lightness. In contrast to classification the age is a continuous value. In a time series prediction task future values have to be predicted based on present and past values. For example a prediction task would be if we monitor the length, width and lightness of the fish every day (or every week) from its birth and want to predict its size, its weight or its health status as a grown out fish. If such predictions are successful appropriate fish can be selected early.

Typical fields of unsupervised learning are projection methods ("principal component analysis", "independent component analysis", "factor analysis", "projection pursuit"), clustering methods ("k-means", "hierarchical clustering", "mixture models", "self-organizing maps"), density estimation ("kernel density estimation", "orthonormal polynomials", "Gaussian mixtures") or generative models ("hidden Markov models", "belief networks"). Unsupervised methods try to extract structure in the data, represent the data in a more compact or more useful way, or build a model of the data generating process or parts thereof.

2.4 Feature Extraction, Selection, and Construction

As already mentioned in our example with the salmon and sea bass, features must be extracted from the original data. To generate features from the raw data is called *feature extraction* Hochreiter and Schmidhuber [1997a,d, 1999c,a,b], Hochreiter and Mozer [2000, 2001a,d].

In our example features were extracted from images. Another example is given in Fig. 2.7 and Fig. 2.8 where brain patterns have to be extracted from fMRI brain images. In these figures also temporal patterns are given as EEG measurements from which features can be extracted. Features from EEG patterns would be certain frequencies with their amplitudes whereas features from the fMRI data may be the activation level of certain brain areas which must be selected.

In many applications features are directly measured, such features are length, weight, etc. In our fish example the length may not be extracted from images but is measured directly.

However there are tasks for which a huge number of features is available. In the bioinformatics context examples are the microarray technique where 30,000 genes are measured simultaneously with cDNA arrays, peptide arrays, protein arrays, data from mass spectrometry, "single nucleotide" (SNP) data, etc. In such cases many measurements are not related to the task to be solved. For example only a few genes are important for the task (e.g. detecting cancer or predicting the outcome of a therapy) and all other genes are not. An example is given in Fig. 2.9, where one variable is related to the classification task and the other is not.

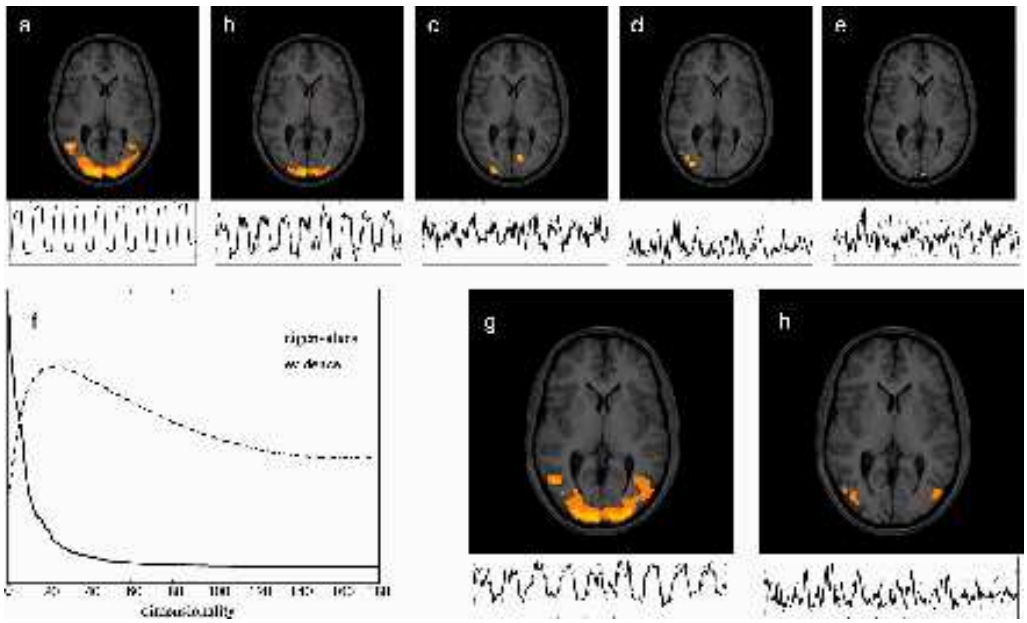

Figure 2.7: Images of fMRI brain data together with EEG data. Certain active brain regions are marked.

The first step of a machine learning approach would be to select the relevant features or chose a model which can deal with features not related to the task. Fig. 2.10 shows the design cycle for generating a model with machine learning methods. After collecting the data (or extracting the features) the features which are used must be chosen.

The problem of selecting the right variables can be difficult. Fig. 2.11 shows an example where single features cannot improve the classification performance but both features simultaneously help to classify correctly. Fig. 2.12 shows an example where in the left and right subfigure the features mean values and variances are equal for each class. However, the direction of the variance differs in the subfigures leading to different performance in classification.

There exist cases where the features which have no correlation with the target should be selected and cases where the feature with the largest correlation with the target should not be selected. For example, given the values of the left hand side in Tab. 2.1, the target t is computed from two features f_1 and f_2 as $t = f_1 + f_2$. All values have mean zero and the correlation coefficient between t and f_1 is zero. In this case f_1 should be selected because it has negative correlation with f_2. The top ranked feature may not be correlated to the target, e.g. if it contains target-independent information which can be removed from other features. The right hand side of Tab. 2.1 depicts another situation, where $t = f_2 + f_3$. f_1, the feature which has highest correlation coefficient with the target (0.9 compared to 0.71 of the other features) should not be selected because it is correlated to all other features.

In some tasks it is helpful to combine some features to a new feature, that is to construct features. In gene expression examples sometimes combining gene expression values to a meta-gene value gives more robust results because the noise is "averaged out". The standard way to combine linearly dependent feature components is to perform PCA or ICA as a first step. Thereafter the relevant PCA or ICA components are used for the machine learning task. Disadvantage is that

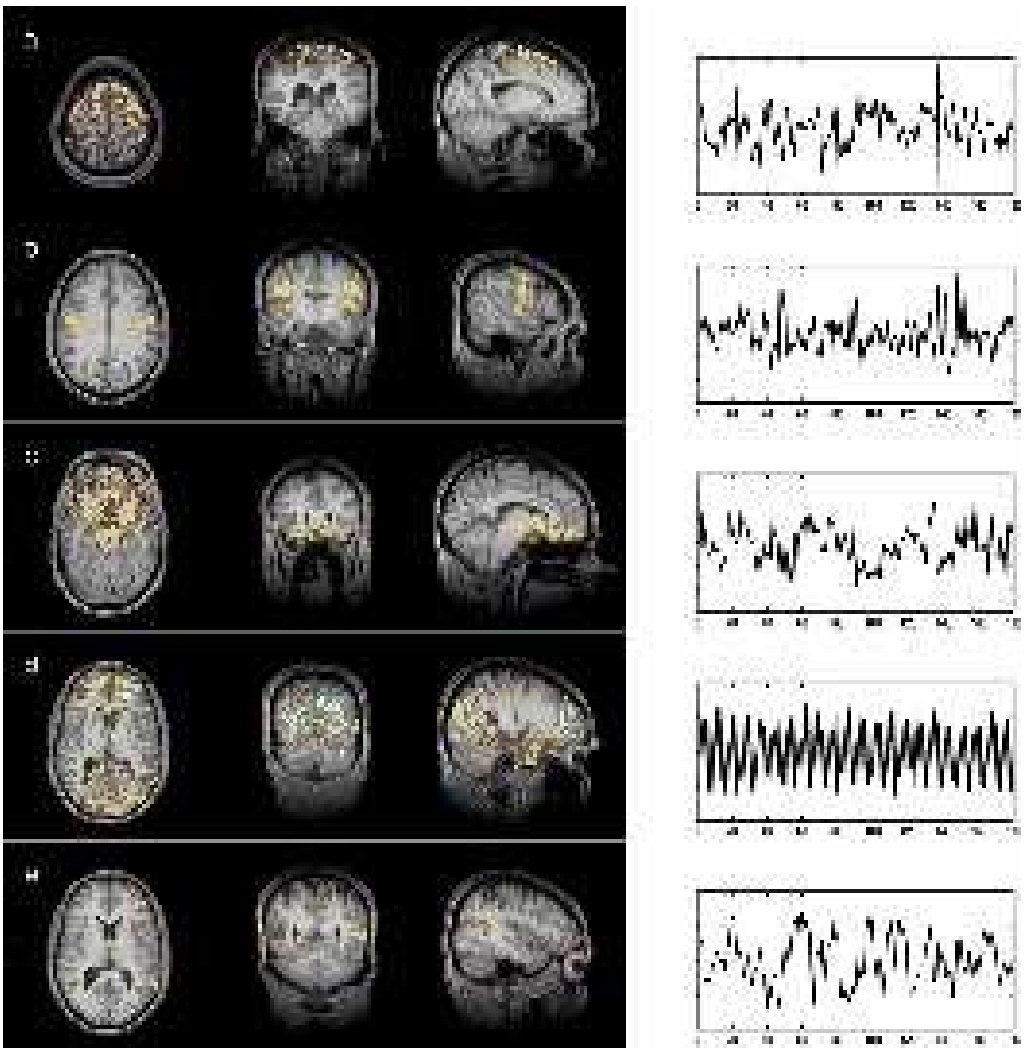

Figure 2.8: Another image of fMRI brain data together with EEG data. Again, active brain regions are marked.

f_1	f_2	t	f_1	f_2	f_3	t
-2	3	1	0	-1	0	-1
2	-3	-1	1	1	0	1
-2	1	-1	-1	0	-1	-1
2	-1	1	1	0	1	1

Table 2.1: Left hand side: the target t is computed from two features f_1 and f_2 as $t = f_1 + f_2$. No correlation between t and f_1.

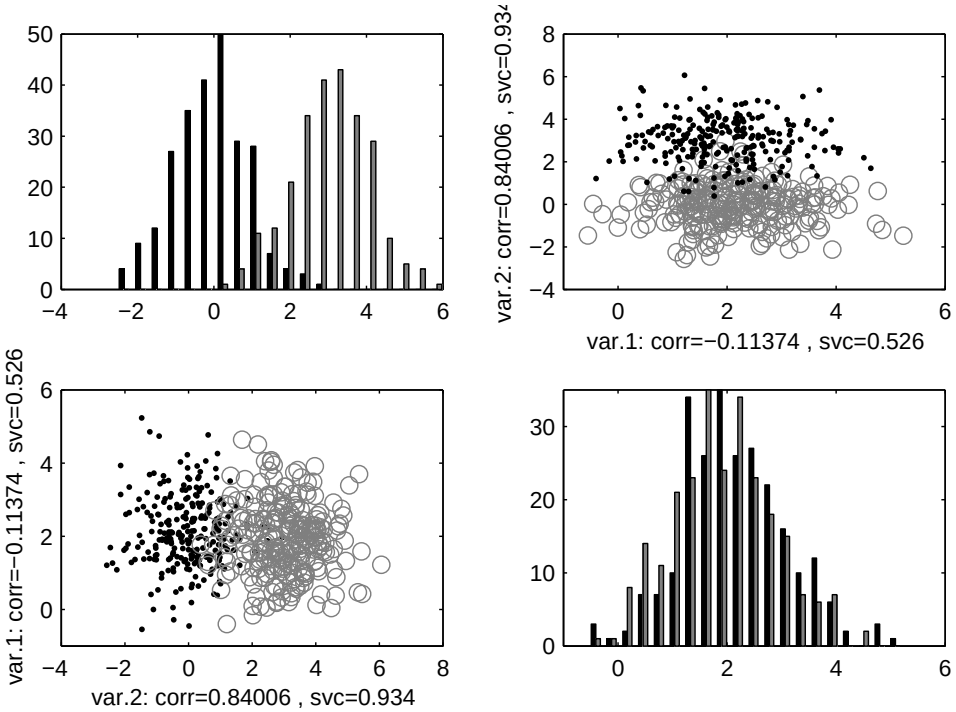

Figure 2.9: Simple two feature classification problem, where feature 1 (var. 1) is noise and feature 2 (var. 2) is correlated to the classes. In the upper right figure and lower left figure only the axis are exchanged. The upper left figure gives the class histogram along feature 2 whereas the lower right figure gives the histogram along feature 1. The correlation to the class (corr) and the performance of the single variable classifier (svc) is given.

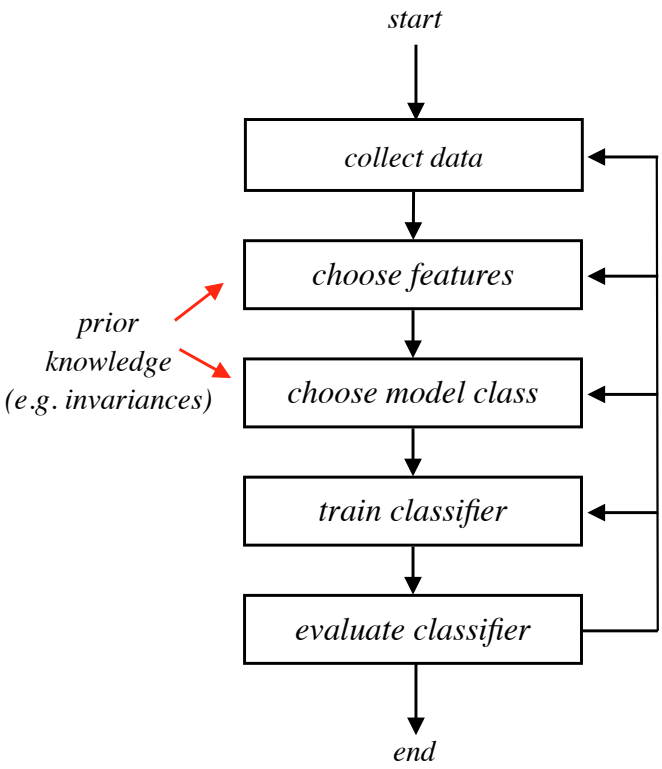

Figure 2.10: The design cycle for machine learning in order to solve a certain task. Copyright © 2001 John Wiley & Sons, Inc.

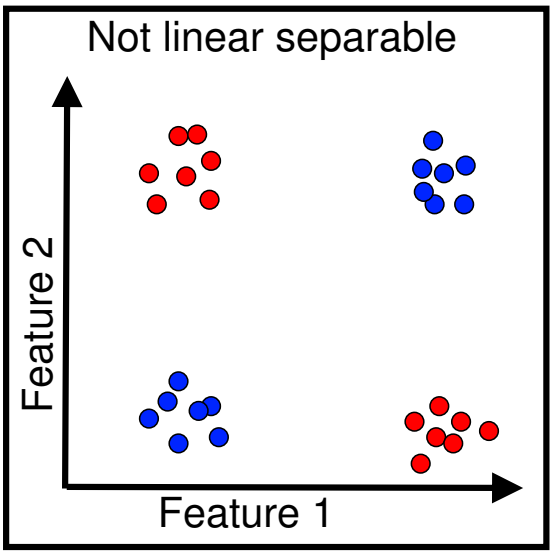

Figure 2.11: An XOR problem of two features, where each single feature is neither correlated to the problem nor helpful for classification. Only both features together help.

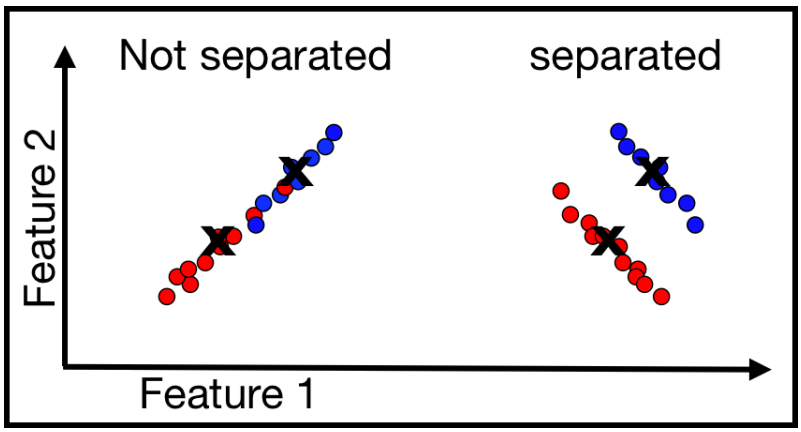

Figure 2.12: The left and right subfigure shows each two classes where the features mean value and variance for each class is equal. However, the direction of the variance differs in the subfigures leading to different performance in classification.

often PCA or ICA components are no longer interpretable.

Using kernel methods the original features can be mapped into another space where implicitly new features are used. In this new space PCA can be performed (kernel-PCA). For constructing non-linear features out of the original one, prior knowledge on the problem to solve is very helpful. For example a sequence of nucleotides or amino acids may be presented by the occurrence vector of certain motifs or through their similarity to other sequences. For a sequence the vector of similarities to other sequences will be its feature vector. In this case features are constructed through alignment with other features.

Issues like missing values for some features or varying noise or non-stationary measurements have to be considered in selecting the features. Here features can be completed or modified.

2.5 Parametric vs. Non-Parametric Models

An important step in machine learning is to select the methods which will be used. This addresses the third step in Fig. 2.10. To "choose a model" is not correct as a model class must be chosen. Training and evaluation then selects an appropriate model from the model class. Model selection is based on the data which is available and on prior or domain knowledge.

A very common model class are *parametric models*, where each parameter vector represents a certain model. Parametric models are neural networks, where the parameter are the synaptic weights between the neurons, or support vector machines, where the parameters are the support vector weights. For parametric models in many cases it is possible to compute derivatives of the models with respect to the parameters. Here gradient directions can be used to change the parameter vector and, therefore, the model. If the gradient gives the direction of improvement then learning can be realized by paths through the parameter space.

Disadvantages of parametric models are: (1) one model may have two different parameterizations and (2) defining the model complexity and therefore choosing a model class must be done via the parameters. Case (1) can easily be seen at neural networks where the dynamics of one neuron