Optimal Mean-Precision Classifier

David M.J. Tax, Marco Loog, and Robert P.W. Duin

Information and Communication Theory Group
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
D.M.J.Tax@tudelft.nl

Abstract. For pattern recognition problems where a small set of relevant objects should be retrieved from a (very) large set of irrelevant objects, standard evaluation criteria are often insufficient. For these situations often the precision-recall curve is used. An often-employed scalar measure derived from this curve is the mean precision, that estimates the average precision over all values of the recall. This performance measure, however, is designed to be non-symmetric in the two classes and it appears not very simple to optimize. This paper presents a classifier that approximately maximizes the mean precision by a collection of simple linear classifiers.

Keywords: Pattern recognition, performance evaluation, information retrieval, precision-recall graph.

1 Introduction

The standard performance measure in pattern recognition is the classification performance. In most real world applications the classification error is not well suited. For classification problems where classes are very imbalanced, or where the misclassification costs for different classes vary widely, the classification error can give a very unfair impression of the true performance. For the situations where the true misclassification costs are unknown, often the Receiver Operating Characteristic curve (ROC curve) is used. The ROC curve insensitive to class priors [Fla03], and the Area under the ROC curve (AUC) is often used as a scalar performance measure to compare classifiers [Bra97]. Classifiers are especially developed to directly optimize the AUC [BS05, FFHO02].

In the field of Information Retrieval (IR) one often does not use the ROC curve, but the Precision-Recall-curve. There the user queries a database of positive and negative documents, and retrieves the M most promising documents [SM83]. The performance measures therefore should incorporate the fact that just a limited number of objects is presented to the user. Two classical measures for Information Retrieval systems are the Precision and the Recall. Roughly speaking, the Precision measures how 'pure' the retrieved M documents are (so that just a few irrelevant documents are retrieved), while Recall measures how many of the total relevant documents are retrieved.

J.A. Benediktsson, J. Kittler, and F. Roli (Eds.): MCS 2009, LNCS 5519, pp. 72–81, 2009.

To evaluate a classifier, the number of retrieved documents M is often fixed, and the Precision and Recall are measured. Comparing two classifiers now involves comparing the two pairs of performances. To combine these two performances often the F-measure, which is the harmonic mean between the precision and recall, is used [vR79]. The comparison between two systems becomes harder when the operating point for the systems is not known beforehand, and the systems should work with varying operating points. In these situations it may be preferable to consider the mean precision, which is the precision averaged over a range of recall values.

Unfortunately, we don't know classifiers that directly optimize this mean precision. Often classifiers are constructed that minimize a classification error, or are tuned for a specific operating point in the precision-recall graph. This paper presents an algorithm that approximates the optimal mean precision by a combination of linear classifiers. In section 2 the mean precision is defined and rewritten such that it can estimated on a training dataset of objects. The estimator is decomposed in a sum of terms, where each term combines objects with equal precision. In section 3 a classifier is constructed that optimizes each term in the sum. In section 4 experiments are performed and in section 5 conclusions and further research directions are given.

2 Mean Precision

We have two classes, the positive target class, and the negative outlier or background class. It is assumed that the positive class has to be retrieved and that this class is (much) smaller than the negative class. Assume we have N^+ positive and N^- negative objects in our dataset: $x_j^+, j = 1, ..., N^+$ and $x_j^-, j = 1, ..., N^-$.

A classifier should rank all the objects in a dataset. Objects that are ranked below a threshold θ are 'accepted', otherwise they are 'rejected'. A graphical representation is given in Figure 1. The number of positive objects that is

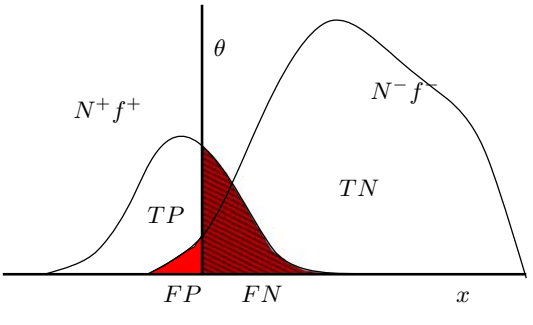

Fig. 1. Graphical representation of the distribution of the positive and negative class on the classifier output. Objects below the threshold θ are classified as positive. The different sets of data are shown: True Positive, True Negative, False Positive and False Negative.

accepted, is called the True Positives TP. The number of positive objects that is rejected, is called the False Positives FP.

The two performance measures, precision and recall, are defined as:

$$\text{precision}(\theta) = \frac{TP(\theta)}{TP(\theta) + FP(\theta)} \tag{1}$$

$$\text{recall}(\theta) = \frac{TP(\theta)}{TP(\theta) + FN(\theta)} = \frac{TP(\theta)}{N^+} \tag{2}$$

2.1 Mean Precision for Given 1D Distributions

Consider a one dimensional feature x for which the true distributions of the positive and negative classes f^+ and f^- are known. Then the cumulative distributions F^+ and F^- are defined, and the total cumulative sum becomes:

$$F(x) = N^+ F^+(x) + N^- F^-(x). \tag{3}$$

The True Positives and False Positives can be written as:

$$TP(\theta) = N^+ F^+(\theta), \quad FP(\theta) = N^- F^-(\theta). \tag{4}$$

The recall for a given threshold θ becomes:

$$\text{recall}(\theta) = \frac{N^+ F^+(\theta)}{N^+} = \int_{-\infty}^{\theta} f^+(u)du \tag{5}$$

and the precision:

$$\text{precision}(\theta) = \frac{N^+ F^+(\theta)}{F(\theta)} = \frac{N^+ \int_{-\infty}^{\theta} f^+(u)du}{N^+ \int_{-\infty}^{\theta} f^+(u)du + N^- \int_{-\infty}^{\theta} f^-(u)du}. \tag{6}$$

The mean precision is now defined as the precision averaged over all values of the recall. This can be written as an average over θ when a coordinate transform is applied. Note that

$$\frac{d\text{recall}(\theta)}{d\theta} = \frac{dF^+(\theta)}{d\theta} = f^+(\theta). \tag{7}$$

Therefore the integration variable $d\text{recall}(\theta)$ can be replaced by $f^+(\theta)d\theta$:

$$\overline{\text{prec}} = \int_{-\infty}^{\infty} \frac{N^+ F^+(\theta)}{F(\theta)} f^+(\theta)d\theta = \int_{-\infty}^{\infty} \frac{N^+ \int_{-\infty}^{\theta} f^+(u)du}{N^+ \int_{-\infty}^{\theta} f^+(u)du + N^- \int_{-\infty}^{\theta} f^-(u)du} f^+(\theta)d\theta. \tag{8}$$

2.2 Mean Precision Using Sampled Distributions

When the exact distributions f^+ and f^- are not available, the distributions can be approximated using sampled versions. Assume that the objects $x_j^+, j = 1, ..., N^+$ are samples from the positive class, and $x_j^-, j = 1, ..., N^-$ are from the negative class. For simplicity later, we assume that the positive and negative objects are ordered: $x_1^+ < x_2^+ < ..., < x_{N^+}^+$. When no superscript is given, the data can be of any of the two classes $x_j, j = 1, ..., N$, where $N = N^+ + N^-$. Then:

$$f^+(x) = \sum_{j=1}^{N^+} \delta(x - x_j^+), \quad F^+(x) = \sum_{j=1}^{N^+} \mathcal{I}(x \geq x_j^+), \tag{9}$$

where $\delta(x)$ is a Dirac-Delta function and $\mathcal{I}(.)$ is the indicator function $\mathcal{I}(A) = 1$ if the statement A is true and $\mathcal{I}(A) = 0$ otherwise).

Substituting this in (5) and (6) gives for a single value of θ:

$$\text{precision}(\theta) = \frac{\sum_{j=1}^{N^+} \mathcal{I}(\theta \geq x_j^+)}{\sum_j^N \mathcal{I}(\theta \geq x_j)}, \qquad \text{recall}(\theta) = \sum_{j=1}^{N^+} \mathcal{I}(\theta \geq x_j^+). \tag{10}$$

To find the mean precision, we have to average over all values of recall. The average over all positive objects x_i is computed by:

$$\overline{\text{prec}} = \frac{1}{N^+} \sum_{i=1}^{N^+} \text{prec}(x_i^+) = \frac{1}{N^+} \sum_{i=1}^{N^+} \frac{\sum_{j=1}^{N^+} \mathcal{I}(x_i^+ \geq x_j^+)}{\sum_{j=1}^{N} \mathcal{I}(x_i^+ \geq x_j)} = \frac{1}{N^+} \sum_{i=1}^{N^+} \frac{i}{\sum_{j=1}^{N} \mathcal{I}(x_i^+ \geq x_j)}. \tag{11}$$

Note that in the last step we assumed that the objects $x_1^+, x_2^+, ..., x_{N^+}^+$ are ordered ($x_1^+ < x_2^+ < ..., < x_{N^+}^+$), such that the sum in the numerator can be reduced to: $\sum_{j=1}^{N^+} \mathcal{I}(x_i^+ \geq x_j^+) = i$.

2.3 Decomposition of the Mean Precision

The sum given in the denominator of (11) can be decomposed into two parts; one sum over all positive objects and one over all negative objects. This results in:

$$\sum_{j=1}^{N} \mathcal{I}(x_i^+ \geq x_j) = \sum_{j=1}^{N^+} \mathcal{I}(x_i^+ \geq x_j^+) + \sum_{j=1}^{N^-} \mathcal{I}(x_i^+ \geq x_j^-) = i + c_i^-. \tag{12}$$

We define S_0 as the subset of the positive objects x_i^+, $i = 1, ..., m_0$ for which $c_i^- = \sum_{j=1}^{N^-} \mathcal{I}(x_i^+ \geq x_j^-) = 0$. These are the objects that do not have any negative objects with a lower feature value. In figure 2 these are the three positive objects that are located left of 0.1. For these objects the terms in the sum in (11) becomes one. Next we can define the sets S_k as the subsets of positive objects for which $c_i^- = k$, and define the cardinalities of the sets $m_k = |S_k|$:

$$S_k = \left\{ x_i^+ \Big| \sum_{j=1}^{N^-} \mathcal{I}(x_i^+ \geq x_j^-) = k \right\}, \quad k = 0, .., N^-. \tag{13}$$

For instance, the set S_1 in figure 2 contains the positive objects that have just a single negative object left of them, so that are all positive objects between 0.1 and 1.0. For this example there is two objects (around $x = 0.5$), and therefore $m_1 = 2$. Note also that $m_6 = m_7 = ... = m_{12} = 0$.

The total mean precision can now be written as (combining (11) and (12)):

$$\begin{aligned} \overline{\text{prec}} &= \frac{1}{N^+} \sum_{i=1}^{N^+} \frac{i}{i + \sum_{j=1}^{N^-} \mathcal{I}(x_i^+ \geq x_j^-)} \\ &= \frac{1}{N^+} \sum_{i=1}^{N^+} \frac{i}{i + c_i^-} = \frac{1}{N^+} \left[\sum_{i=1}^{m_0} \frac{i}{i} + \sum_{i=m_0+1}^{m_0+m_1} \frac{i}{i+1} + \sum_{i=m_0+m_1+1}^{m_0+m_1+m_2} \frac{i}{i+2} + ... \right] \end{aligned} \tag{14}$$

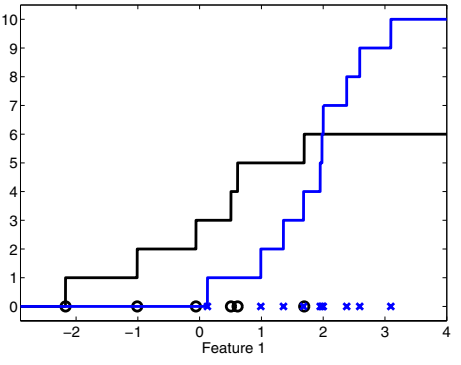

Fig. 2. The (unnormalized) cumulative distributions $N^+F^+(x)$ and $N^-F^-(x)$ for the positive objects (small circles) and the negative objects (crosses)

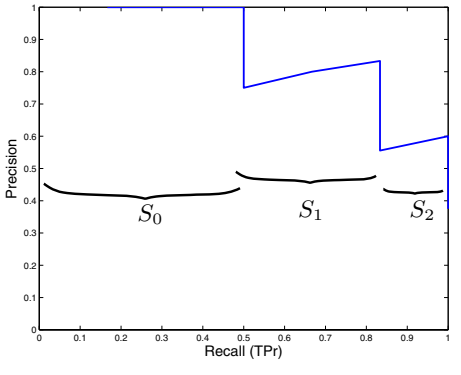

Fig. 3. The resulting precision-recall graph for the data that is shown in Figure 2

In figure 3 the Precision-Recall curve for the data shown in figure 2 is shown. The different subsets S_k of positive objects define different ranges in the recall. For this example only three subsets, S_0, S_1 and S_2 are non-empty, and they are indicated in the graph. The first three positive objects on the left are element of S_0 and have a precision of 1. The two objects in S_1 have two different precision values of 4/5 and 5/6.

3 Optimizing Mean Precision

For a given feature x the mean precision (11) or (14) can directly be computed. Assume we would like to define a new feature as a linear combination of original features:

$$z = \mathbf{w}^T \mathbf{x} \tag{15}$$

in such a way that it optimizes the mean precision:

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}} \frac{1}{N^+} \sum_{i=1}^{N^+} \frac{i}{i + \sum_{j=1}^{N^-} \mathcal{I}(\mathbf{w}^T \mathbf{x}_i^+ \geq \mathbf{w}^T \mathbf{x}_j^-)} \tag{16}$$

$$= \operatorname*{argmax}_{\mathbf{w}} \frac{1}{N^+} \left[\sum_{i=1}^{m_0} \frac{i}{i} + \sum_{i=m_0+1}^{m_0+m_1} \frac{i}{i+1} + \sum_{i=m_0+m_1+1}^{m_0+m_1+m_2} \frac{i}{i+2} + ... \right]. \tag{17}$$

Unfortunately, this is a complicated nonlinear optimization problem. When the weights $\mathbf{w}$ are changed a bit, the ordering of the objects may change drastically, and objects move between the different sets S_k. This causes the numbers m_k to change and therefore also the sums in (17). To optimize all terms simultaneously appears to be very hard (we could not reduce it to a easily-solvable optimization problem). Therefore we decide to optimize (17) term by term in a greedy optimization.

3.1 Optimizing the First Term in Mean Precision

To optimizing the first term in (17) we try to find that direction $\mathbf{w}$ in feature space such that the maximum number of positive objects do not have a single negative object below them. That means that we try to find a classifier $f(\mathbf{x}) = \mathcal{I}(\mathbf{w}^T \mathbf{x} < \theta)$ that maximizes m_0 first. This can be done by the following (linear programming) optimization procedure:

$$\min_{\mathbf{w}} |\mathbf{w}| + C \sum_{i \in +} \xi_i \tag{18}$$

$$\text{s.t.} \quad \mathbf{w}^T \mathbf{x}_i^- \geq \theta, \qquad \text{for negative objects} \tag{19}$$

$$\mathbf{w}^T \mathbf{x}_i^+ \leq \theta + \xi_i, \quad \xi_i \geq 0 \qquad \text{for positive objects.} \tag{20}$$

In the constraints (19) all the negative objects x_i^- are forced to be above the threshold θ, so the negative objects are classified without error and the term $\sum_{j=1}^{N^-} \mathcal{I}(\mathbf{w}^T \mathbf{x}_i^+ \geq \mathbf{w}^T \mathbf{x}_j^-)$ in (16) becomes 0. Positive objects x_i^+ that are not

on the correct side (i.e. below) of the decision boundary are punished by a so-called slack ξ_i. This is defined in the first constraint (20). The sum of the slacks is minimized in the function (18), together with a regularization term on the L_1-norm of the weight vector $\mathbf{w}$.

This optimization procedure finds a linear combination of original features, or a linear classifier, such that the number of correctly classified positive objects is as large as possible, and such that none of the negative objects is misclassified.

3.2 Optimizing the Next Terms in Mean Precision

The previous formulation only considers the first term in (17). To maximize the second term in (17), or to find the largest set S_1, we have to find the largest set of positive objects that have *one* negative object below them. This optimization becomes very complicated. Therefore we decided to remove the negative object that is the closest to the set S_0. That is the object for which the constraint (20) is violated the first[1]. After this object is removed from the training set, the optimization (18) is run again, selecting the set S_1.

This process is repeated until all positive objects have been classified to the positive class, or until a pre-specified value for the recall is obtained. Assume that a certain recall r is required. The number of classifiers n that has to be trained to obtain at least this recall, becomes $n : \sum_{i=1}^{n} m_i \geq N^+ r$. This depends on the number of positive objects m_k in each of the sets S_k, which again depends on the data. It is therefore not clear beforehand how many classifiers have to be trained to obtain a certain recall.

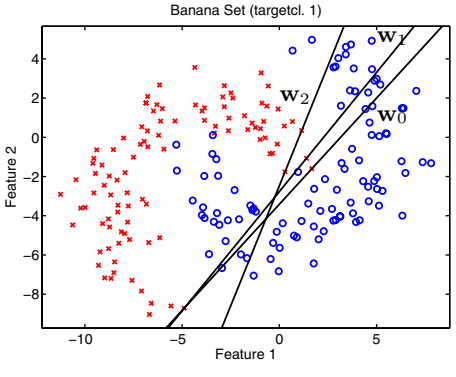

Fig. 4. Scatterplot with a positive and negative class (circles and crosses respectively), containing the sequence of classifiers $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$

[1] In linear programming optimization algorithms each of the constraints obtains a dual variable that indicates how heavily this constraint has to be enforced. The object corresponding to the constraint with the highest dual variable is selected to be removed.

This process is also shown in figure 4. The first classifier $\mathbf{w}_0$ separates the positive objects (indicated by the circles) from the negative objects (crosses). None of the negative objects are misclassified, and around 50% of the positive objects are correct. In the second iteration the rightmost negative object (at position $(1.6, -1.6)$) is removed, and the new classifier $\mathbf{w}_2$ is trained.

3.3 Combining the Classifiers

Unfortunately, the procedure does not result in a single unique $\mathbf{w}$, but in a collection of classifiers $\mathbf{w}_k$. When a new object has to be classified, the classifiers have to be combined into a single output, depending on the actual operating point that is chosen.

Following the philosophy of optimizing (17) term by term, an iterative scheme has to be used. First classify the new object $\mathbf{w}$ by $\mathbf{w}_0$. When $\mathbf{z}$ is classified positive by $\mathbf{w}_0$, the classification is finished and the object $\mathbf{z}$ is labeled positive. When $\mathbf{w}_0$ classifies $\mathbf{z}$ as negative, the object is presented to the next classifier $\mathbf{w}_1$, and the process repeats itself. The classifier labels the object as positive, or it moves it to the next classifier, until the last n-th classifier is reached, and the object is classified negative.

Input: Datasets $\{x_j^+, j = 1, ..., N^+\}$ and $\{x_j^-, j = 1, ..., N^-\}$, M
Output: Classifier $\mathbf{w}$
for $k \leftarrow 0$ **to** M **do**
  optimize $\mathbf{w}_k$ using (17);
  find object $\mathbf{x}^* \in \{x_j^-\}$ with the largest dual variable;
  remove object $\mathbf{x}^*$ from the negative examples;
end
combine classifiers $\mathbf{w}_k$ to classifier $\mathbf{w}$

Algorithm 1. The optimal mean-precision algorithm

In algorithm 1 shows a high-level overview of the steps that are taken in the (approximate) optimization of the mean precision. It is not clear if this is the optimal approach. It is expected that the final classifier becomes more stable and robust when the individual classifiers $\mathbf{w}_k$ are averaged, but at the expense of the flexibility of the classifier. This is subject for further research.

4 Experiments

In this section we perform some experiments on some real world datasets to show the feasibility of the approach. The number of classifiers that is iteratively trained is also limited. Three versions are tested, using a single linear classifier (just $\mathbf{w}_0$, $M = 1$), using $M = 10$ classifiers and $M = 25$ classifiers. It was observed that for more than $M = 25$ classifiers the performance rarely improved.

We compare the Optimal-Mean-Precision formulation (OptPrec) with a collection of simple classifiers: the Linear Discriminant Analysis (LDA) and

the Quadratic Discriminant (QD) [DHS01], the Logistic classifier [And82], the Parzen density classifier [Par62] and the Support Vector Classifier [Vap98]. For the Parzen classifier, the width parameter is optimized by maximizing the likelihood using leave-one-out on the training set. For the support vector classifier a linear kernel is used, where the C parameter is optimized using 10-fold cross-validation. In the experiments can show some results on the Imports85 dataset (159 objects in 25D), the Glass dataset (214 objects in 9D, where the classification task is to distinguish class 1 from the rest), and the Sonar dataset (208 objects in 60D). These datasets are taken from the UCI repository [NHBM98].

Table 1. The mean-precision (×100) for different classifiers and datasets. The best performances are indicated in bold. Results averaged over 10-fold stratified cross-validation. Values between the brackets indicate the standard deviation.

classifier	Imports85	Glass	Sonar
LDA	**85.2 (15.2)**	69.8 (11.7)	74.1 (15.6)
QD	**77.2 (18.0)**	**81.5 (19.0)**	77.1 (13.9)
Logistic	72.9 (23.2)	74.3 (15.1)	79.6 (14.9)
Parzen	**85.4 (17.3)**	75.4 (16.7)	80.9 (12.7)
SVM	**88.9 (14.4)**	72.1 (15.6)	77.9 (15.3)
OptPrec M=1	**90.8 (12.0)**	**79.1 (28.8)**	77.5 (17.2)
OptPrec M=10	**90.6 (12.1)**	**93.1 (9.4)**	76.6 (18.0)
OptPrec M=25	**89.4 (12.0)**	**93.1 (9.4)**	76.4 (17.8)
Kernel OptPrec M=1	74.6 (15.8)	80.7 (14.1)	**99.0 (1.0)**

The experimental results are shown in Table 1. All experiments are performed using 10-fold stratified cross-validation, and the performance measure is mean-precision. For each dataset the best average performance is written in bold. Furthermore, all performances that are not significantly worse (in terms of a one-sided t-test with a 5% significance level) are also written in bold.

The results on the Imports85 dataset show a common outcome: many classifiers have a similar performance, but the OptPrec slightly outperforms the other classifiers (although not significantly). Note also that the outcomes of the OptPrec are a bit more robust than of the other classifiers. The Glass dataset results show a situation where the OptPrec significantly outperforms the other approaches. Only the classifier that only maximizes the size of the first set m_0, is too unstable and gives poor results. The outcomes on the Sonar dataset shows that the OptPrec classifier is not always optimal in its linear implementation, but that for some datasets nonlinear decision boundaries are needed. This is implemented by kernelizing the linear classifier; the original data is mapped into a new feature representation using (in this case) the RBF kernel with an optimized width parameter σ. For the Sonar dataset it resulted in an almost perfect mean precision.

For many datasets the OptPrec classifier is not superior: in the cases where data is (almost) separable, or when the decision boundary is very nonlinear other classifiers may perform equally well, or better. Even more importantly, in

small sample size classification problems it is often advantageous to use *all* the available training data for estimating the class conditional probabilities. It is hard to estimate the subset of positive objects in S_0 from a small training set. In these cases it might be advantageous to make the individual estimates $\mathbf{w}_k$ more robust and combine them. This is still an issue for further research.

5 Conclusions

This paper presents the derivation of a classifier that (approximately) optimizes the mean precision for a two-class classification problem. The classifier iteratively separates a part of the positive class from the negative class, such that the positive part is as 'pure' as possible (i.e. it does not contain any negative objects) and as large as possible. For each separation of a pure part, a classifier is obtained. When these classifiers are combined into a final classifier, the mean precision is optimized. Experiments show that for some datasets very good performances can be obtained. Further research is needed to investigate the possibility to optimize the mean precision in one step, how the classifiers have to be combined (in particular in the low sample size situation) and how it will perform on real world retrieval problems.

References

[And82] Anderson, J.A.: Logistic discrimination. In: Kirshnaiah, P.R., Kanal, L.N. (eds.) Classification, Pattern Recognition and Reduction of Dimensionality. Handbook of Statistics, vol. 2, pp. 169–191. North Holland, Amsterdam (1982)

[Bra97] Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30(7), 1145–1159 (1997)

[BS05] Brefeld, U., Scheffer, T.: AUC miximizing support vector learning. In: Proceedings of ICML 2005 workshop on ROC analysis in Machine Learning (2005)

[DHS01] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, Chichester (2001)

[FFHO02] Ferri, C., Flach, P., Hernandez-Orallo, J.: Learning decision trees using the area under the ROC curve. In: Proceedings of the ICML (2002)

[Fla03] Flach, P.: The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In: Proceedings of the international conference on Machine learning 2003, pp. 194–201 (2003)

[NHBM98] Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998)

[Par62] Parzen, E.: On estimation of a probability density function and mode. Annals of Mathenatical Statistics 33, 1065–1076 (1962)

[SM83] Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)

[Vap98] Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)

[vR79] van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterwort (1979)