Optimizing F-Measure with Support Vector Machines

David R. Musicant
Department of Mathematics and Computer Science
Carleton College
Northfield, MN 55057
dmusican@carleton.edu

Vipin Kumar and **Aysel Ozgur**
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455
kumar,ozgur@cs.umn.edu

Abstract

Support vector machines (SVMs) are regularly used for classification of unbalanced data by weighting more heavily the error contribution from the rare class. This heuristic technique is often used to learn classifiers with high F-measure, although this particular application of SVMs has not been rigorously examined. We provide significant and new theoretical results that support this popular heuristic. Specifically, we demonstrate that with the right parameter settings SVMs approximately optimize F-measure in the same way that SVMs have already been known to approximately optimize accuracy. This finding has a number of theoretical and practical implications for using SVMs in F-measure optimization.

Introduction

Support vector machines (SVMs) (Vapnik 1995; Cristianini & Shawe-Taylor 2000) have been shown throughout the last decade to be a popular and successful methodology for classification problems. SVMs for classification have most often been used to optimize accuracy on a given dataset. When training data is unbalanced, however, accuracy is often a poor metric to use. For example, if a dataset has 99% of its points in class "A" and only 1% of its points are in class "B," then an accuracy maximizing classifier may draw the conclusion that "all points are in class A." When dealing with this situation practitioners often prefer to measure *precision* and *recall* (Hand, Mannila, & Smyth 2001) instead of accuracy. Precision and recall are typically each maximized at the expense of the other, and thus practitioners must choose a compromise. One popular balance is F-measure (van Rijsbergen 1979), which is a particular kind of average between precision and recall. F-measure is therefore a relevant goal in any machine learning scenario where data from one class are present in much greater quantities than data from the other class.

In this situation, i.e. when one class is "rare" relative to the other, a traditional SVM can be augmented with a weighting parameter to provide extra emphasis on the rare class. This weighting parameter can be set directly via a heuristic (Morik, Brockhausen, & Joachims 1999), or a tuning procedure can be used to determine what the optimal value for this parameter should be. These methods work well, and have been successful in practice.

It seems reasonable to conclude that a variation on the standard SVM, designed to optimize F-measure, should do a better job than a standard SVM. An SVM variant that optimizes F-measure directly would be highly desirable, but would be difficult to solve due to the nonlinearities inherent in the formulation. We do not attempt to solve such an SVM variant in this work. All approaches to date that we know of using SVMs to maximize F-measure do so by varying parameters in standard SVMs in an attempt to maximize F-measure as much as possible. While this may result in a "best possible" F-measure for a standard SVM, there is no evidence that this technique should produce an F-measure comparable with one from a classifier designed to specifically optimize F-measure. Our discovery, which is the main thrust of this paper, is that for the right parameter settings the standard SVM does in fact optimize an approximation to F-measure. This provides significant new evidence that the ad-hoc techniques that researchers have been using for years are in fact "the right thing to do" in trying to optimize F-measure.

We now describe our notation and give some background material. All vectors will be column vectors unless transposed to a row vector by a prime $'$. For a vector $x \in R^n$, x_* denotes the vector in R^n with components $(x_*)_i = 1$ if $x_i > 0$ and 0 otherwise (i.e. x_* is the result of applying the step function component-wise to x). The notation $A \in R^{m \times n}$ will signify a real $m \times n$ matrix. For such a matrix A' will denote the transpose of A and A_i will denote the i-th row of A. A vector of ones or zeroes in a real space of arbitrary dimension will be denoted by e or 0, respectively. For two vectors x and y in R^n, $x \perp y$ denotes orthogonality, that is $x'y = 0$.

This paper is organized as follows. We first present the standard SVM and the formulation it approximates, the misclassification counting SVM. We next present our new F-measure maximizing SVM, and show its equivalence to the misclassification counting SVM. Finally, we consider implications of this equivalence and conclude.

The Misclassification Counting SVM

We consider the problem of classifying m points in the n-dimensional real space R^n, represented by the $m \times n$ matrix

A, according to membership of each point A_i in the class A_+ or A_- as specified by a given $m \times m$ diagonal matrix D with +1's or -1's along its diagonal. For this problem the standard SVM with a linear kernel (Vapnik 1995) is given by the following quadratic program with parameter $C > 0$:

Formulation 1 (Standard Linear SVM)

$$\begin{array}{lrcl} \min\limits_{(w,b,\xi)\in R^{n+1+m},\ \xi\geq 0} & \frac{1}{2}w'w + Ce'\xi & & \\ \textit{s.t.} & D(Aw - eb) + \xi & \geq & e \end{array} \tag{1}$$

Here w is the normal to the bounding planes:

$$\begin{array}{rcl} x'w & = & b+1 \\ x'w & = & b-1, \end{array} \tag{2}$$

and b determines their location relative to the origin. The plane $x'w = b+1$ bounds the class A_+ points, possibly with some error, and the plane $x'w = b-1$ bounds the class A_- points, also possibly with some error. The linear separating surface is the plane:

$$x'w = b, \tag{3}$$

midway between the bounding planes (2). The quadratic term in (1) is twice the reciprocal of the square of the 2-norm distance $\frac{2}{\|w\|_2}$ between the two bounding planes of (2). This term maximizes this distance which is often called the "margin." If the classes are linearly inseparable, then the two planes bound the two classes with a "soft margin." That is, they bound each set approximately with some error determined by the nonnegative error variable ξ:

$$\begin{array}{lcccll} A_i w & + & \xi_i & \geq & b+1, & \text{for } D_{ii} = 1, \\ A_i w & - & \xi_i & \leq & b-1, & \text{for } D_{ii} = -1. \end{array} \tag{4}$$

Traditionally the 1-norm of the error variable ξ is minimized parametrically with weight C in (1) resulting in an approximate separation.

The standard SVM as presented in (1) measures misclassification errors by a 1-norm distance metric. The 2-norm distance is a common variation (Mangasarian & Musicant 2001; Cristianini & Shawe-Taylor 2000). Both these metrics are typically chosen out of a need to easily solve the support vector machine optimization problem. It is important to point out, however, that the error metric truly desired is a *count* of the number of misclassified points (Cortes & Vapnik 1995). This can be formulated similarly to (1), using the step function $(\cdot)_*$ as follows:

Formulation 2 (Misclassification Counting Linear SVM)

$$\begin{array}{lrcl} \min\limits_{(w,b,\xi)\in R^{n+1+m},\ \xi\geq 0} & \frac{1}{2}w'w + Ce'(\xi)_* & & \\ \textit{s.t.} & D(Aw - eb) + \xi & \geq & e \end{array} \tag{5}$$

Though this approach has been studied (Mangasarian 1994a; Chen & Mangasarian 1996), it is generally avoided because the problem of finding an exact solution is NP-complete. Furthermore, the non-differentiability of the objective function renders analysis more difficult. We therefore approximate it as (Mangasarian 1996):

Formulation 3 (Approximate Misclassification Counting Linear SVM)

$$\begin{array}{lrcl} \min\limits_{(w,b,\xi)\in R^{n+1+m},\ \xi\geq 0} & \frac{1}{2}w'w + Ce's(\xi) & & \\ \textit{s.t.} & D(Aw - eb) + \xi & \geq & e \end{array} \tag{6}$$

where $s(\xi)$ is a differentiable function that approximates the step function. One such choice is the following, where α is an arbitrary positive fixed constant that determines the closeness of the approximation:

$$s(\xi_i) = \begin{cases} 1 - \exp(\alpha\xi_i), & \xi_i \geq 0 \\ 0 & \xi_i < 0 \end{cases} \tag{7}$$

All of the above formulations are motivated by trying to equally minimize, approximately or exactly, the number of classification errors across all points. This is not appropriate when one is concerned with emphasizing correctness on a rare class. To that end, the standard approach is to weight the rare class more heavily. Therefore, let C_+ be the weight assigned to the class A_+, and C_- be the weight assigned to the class A_-. Define the vector $c \in R^m$ as

$$c_i = \begin{cases} C_+ & \textit{if}\, A_i \in A_+ \\ C_- & \textit{if}\, A_i \in A_- \end{cases} \tag{8}$$

The standard SVM (1) is then reformulated as

Formulation 4 (Weighted Standard Linear SVM)

$$\begin{array}{lrcl} \min\limits_{(w,b,\xi)\in R^{n+1+m},\ \xi\geq 0} & \frac{1}{2}w'w + c'\xi & & \\ \textit{s.t.} & D(Aw - eb) + \xi & \geq & e \end{array} \tag{9}$$

Likewise, the misclassification counting SVM (6) can be reformulated as

Formulation 5 (Weighted Approximate Misclassification Counting Linear SVM)

$$\begin{array}{lrcl} \min\limits_{(w,b,\xi)\in R^{n+1+m},\ \xi\geq 0} & \frac{1}{2}w'w + c's(\xi) & & \\ \textit{s.t.} & D(Aw - eb) + \xi & \geq & e \end{array} \tag{10}$$

We now proceed to define the nonlinear kernelized variants of these two approaches. Once this is done, we can consider generating SVMs designed to optimize F-measure.

The KKT optimality conditions (Mangasarian 1994b) for the standard linear SVM, which indicate the conditions under which a potential solution is optimal, are given in the following *dual problem*:

Formulation 6 (KKT Conditions for Weighted Standard Linear SVM)

$$\begin{array}{l} e'Du = 0 \\ u_i = c_i - v_i,\ i = 1, ..., m \\ D(AA'Du - eb) + \xi \geq e \\ u \perp (D(AA'Du - eb) + \xi - e) \\ v \perp \xi = 0 \\ \xi,\ u,\ v \geq 0 \end{array} \tag{11}$$

The variables (w, b) of the standard linear SVM which determine the separating surface (3) can be obtained from the solution of the dual problem above (Mangasarian & Musicant 1999, Eqns. 5 and 7):

$$w = A'Du, \quad b \in \underset{\alpha \in R}{\operatorname{argmin}}\, e'(e - D(AA'Du - e\alpha))_+ \quad (12)$$

In order to introduce a nonlinear kernel into (11) in the normal fashion, we will use the well known "kernel-trick" (Schölkopf 2000) incorporating the following notation. For $A \in R^{m \times n}$ and $B \in R^{n \times \ell}$, the **kernel** $K(A, B)$ maps $R^{m \times n} \times R^{n \times \ell}$ into $R^{m \times \ell}$. A typical kernel is the Gaussian kernel $K(A, B) = \exp(-\mu \|A'_i - B_{\cdot j}\|^2)$, $i, j = 1, \ldots, m$, $\ell = m$, while a linear kernel is $K(A, B) = AB$. We therefore substitute a kernel in for the matrix product AA' to obtain the kernelized formulation:

Formulation 7 (KKT Conditions for Weighted Standard Nonlinear SVM)

$$\begin{array}{l} e'Du = 0 \\ u_i = c_i - v_i,\ i = 1, ..., m \\ D(K(A, A')Du - eb) + \xi \geq e \\ u \perp (D(K(A, A')Du - eb) + \xi - e) \\ v \perp \xi = 0 \\ \xi,\ u,\ v \geq 0 \end{array} \quad (13)$$

The separating surface in this case is given by:

$$K(x', A')Du = b \quad (14)$$

Finally, we now present KKT conditions for the misclassification counting SVM (6). Using the same kernel trick, we obtain:

Formulation 8 (KKT Conditions for Weighted Approximate Misclassification Counting Nonlinear SVM)

$$\begin{array}{l} e'Du = 0 \\ u_i = c_i \frac{\partial s(\xi_i)}{\partial \xi_i} - v_i,\ i = 1, ..., m \\ D(K(A, A')Du - eb) + \xi \geq e \\ u \perp (D(K(A, A')Du - eb) + \xi - e) \\ v \perp \xi = 0 \\ \xi,\ u,\ v \geq 0 \end{array} \quad (15)$$

The above two formulations describe precisely the conditions under which the standard SVM and the misclassification counting SVM have a solution. We now move on to developing a variation on the SVM that can be used for maximizing F-measure. Once this has been done, we will use these KKT conditions to show the equivalence of this new variation to the nonlinear misclassification counting SVM (15) (for which the standard SVM is typically used as a proxy).

The F-Measure Maximizing SVM

The basic form of the support vector machine is designed to optimize training set accuracy. The goal is to minimize the number of misclassified points in the training set. In the presence of rare classes, a more common approach is to maximize F-measure (van Rijsbergen 1979). To define F-measure, we first focus on the following confusion matrix where we label the rare class as A_+, and the nonrare class as A_-:

		Actual class	
		A_+	A_-
Predicted class	A_+	True Pos (TP)	False Pos (FP)
	A_-	False Neg (FN)	True Neg (TN)
	Total	# of pos (m_+)	# of neg (m_-)

We next define *precision* (P) and *recall* (R) as:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \quad (16)$$

Recall measures how many of the rare points are predicted to be rare, whereas precision measures how many of the points predicted as rare are in fact rare. Both precision and recall are desirable, but they typically trade off against each other. The following weighted average F-measure is thus often used:

$$F = \frac{2(P)(R)}{P + R} \quad (17)$$

We wish to formulate a modified SVM that actually optimizes F-measure in addition to performing structural risk minimization (Vapnik 1995; Cristianini & Shawe-Taylor 2000) as an SVM normally does. To that end, we observe that can express F-measure as:

$$F = \frac{2(TP)^2}{2(TP)^2 + (TP)(FP) + (TP)(FN)} \quad (18)$$

We can simplify this expression and also replace TP by its equal value $m_+ - FN$ to obtain:

$$F = \frac{1}{1 + \frac{1}{2}\, \frac{FP+FN}{m_+ - FN}} \quad (19)$$

We can thus conclude that in order to maximize F-measure, we can instead minimize the quantity $\frac{FP+FN}{m_+ - FN}$. This can be easily formulated into a variation on the standard linear SVM, if we define an $m \times 1$ vector n such that $n_i = 1$ if point i is in class A_+ and $n_i = 0$ otherwise. Our new F-measure maximizing SVM formulation is:

Formulation 9 (F-measure Maximizing Linear SVM)

$$\begin{array}{rl} \min\limits_{(w,b,\xi) \in R^{n+1+m},\ \xi \geq 0} & \frac{1}{2} w'w + C \frac{e'(\xi)_*}{m_+ - n'(\xi)_*} \\ s.t. & D(Aw - eb) + \xi \geq e \end{array} \quad (20)$$

As earlier, we approximate the step function with a differentiable approximation in order to yield

Formulation 10 (Approximate F-measure Maximizing Linear SVM)

$$\begin{array}{rl} \min\limits_{(w,b,\xi) \in R^{n+1+m},\ \xi \geq 0} & \frac{1}{2} w'w + C \frac{e's(\xi)}{m_+ - n's(\xi)} \\ s.t. & D(Aw - eb) + \xi \geq e \end{array} \quad (21)$$

This optimization problem (21) is precisely the desired formulation in order to optimize F-measure on a training set. The objective directly balances maximizing F-measure while at the same time minimizing the norm of w so as to avoid overfitting. The parameter C controls the balance of these two goals in the same fashion as it does in the standard SVM. This formulation is highly nonlinear and quite difficult to solve. Nonetheless, we can now assert the following:

Proposition 1 *Given a parameter C, an optimal separating surface found by the F-measure optimizing SVM* (21) *is also an optimal separating surface for the misclassification counting SVM* (10) *for an appropriate choice of parameters C_+ and C_- (contained in c) in* (10). *The result also holds for the nonlinear generalizations of these two formulations.*

Proof Proposition 1 can be seen to be true by looking at the KKT optimality conditions for the F-measure optimizing SVM (21):

$$\begin{array}{l} e'Du = 0 \\ u_i = C\dfrac{(m_+ - n's(\xi) + e's(\xi)n_i)}{(m_+ - n's(\xi))^2}\dfrac{\partial s(\xi_i)}{\partial \xi_i} - v_i,\ i = 1...m \\ D(AA'Du - eb) + \xi \geq e \\ u \perp (D(AA'Du - eb) + \xi - e) \\ v \perp \xi = 0 \\ \xi,\ u,\ v \geq 0 \end{array} \tag{22}$$

Using the same kernel substitution that we used in (13) and (15), we obtain the following KKT conditions for the nonlinear F-measure optimizing SVM:

$$\begin{array}{l} e'Du = 0 \\ u_i = C\dfrac{(m_+ - n's(\xi) + e's(\xi)n_i)}{(m_+ - n's(\xi))^2}\dfrac{\partial s(\xi_i)}{\partial \xi_i} - v_i,\ i = 1...m \\ D(K(A, A')Du - eb) + \xi \geq e \\ u \perp (D(K(A, A')Du - eb) + \xi - e) \\ v \perp \xi = 0 \\ \xi,\ u,\ v \geq 0 \end{array} \tag{23}$$

Let (w, b, ξ, u, v) satisfy the KKT conditions (23). Recall that $n_i = 1$ if point i is in class A_+, and $n_i = 0$ otherwise. We can then observe that these conditions are identical to KKT conditions (15) with the following choices of C_+ and C_-:

$$C_+ = C\left(\frac{m_+ - n's(\xi) + e's(\xi)}{(m_+ - n's(\xi))^2}\right) \tag{24}$$

and

$$C_- = C\left(\frac{1}{m_+ - n's(\xi)}\right) \tag{25}$$

We may therefore conclude that for these particular values of C_+ and C_-, (23) and (15) are identical and thus optimization problems (21) and (10) (and their nonlinear extensions) have the same optimal solutions. ∎

Implications

There are a number of specific and important implications that we can draw from Proposition 1.

1. For practical reasons, the standard SVM (9) is used as a proxy for the misclassification counting SVM (10). It has become common practice to optimize F-measure using the standard SVM (9) by tweaking the C_+ and C_- parameters. To our knowledge, despite its common usage this parameter varying methodology has not been justified beyond the simple heuristic idea that weighting the classes differently will help balance precision and recall. Proposition 1 indicates that the misclassification counting SVM (10) is equivalent to the F-measure optimizing SVM (21) for the right set of parameters, and thus that the standard SVM (9) serves just as well as a proxy for the F-measure optimizing SVM (21). Therefore, tweaking C_+ and C_- parameters in the standard SVM (9) for purposes of optimizing F-measure is just as reasonable as the typical procedure of using the standard SVM (9) to approximate misclassification counting. These arguments hold for the nonlinear variations of these formulations as well.

2. A common heuristic to use in choosing C_+ and C_- is to balance them such that the errors for both classes contribute equally. For example (Morik, Brockhausen, & Joachims 1999), choose C_+ and C_- to satisfy the ratio

$$\frac{C_+}{C_-} = \frac{\text{number of points in class A}_-}{\text{number of points in class A}_+} \tag{26}$$

   Proposition 1 indicates that in trying to optimize F-measure, the optimal values for C_+ and C_- are not necessarily determined by this ratio. We note that though this ratio heuristic may provide a reasonable "first guess," optimizing F-measure requires significantly more experimentation.

3. Proposition 1 is also relevant if the linear error function $s(\xi) = \xi$ is used. In this case, we draw the conclusion that appropriate parameter choices in the standard SVM (9) provide the same solution as that of an approximation to the F-measure optimizing SVM where a linear error metric is used.

4. Our original goal was to produce a new SVM formulation to optimize F-measure. Such a problem is highly nonlinear, and would require new approximations and algorithms. On the other hand, existing algorithms and software such as SVM^{light} (Joachims 1999), SVMTorch (Collobert & Bengio 2001), and others have undergone extensive effort to render them fast and usable. The community has been using these tools to optimize F-measure, since it is reasonably easy to do. Our results, combined with the speed and usability of these tools, provide strong evidence that practitioners should feel secure in using these tools to optimize F-measure.

5. A number of non-SVM related techniques exist for dealing with classification in the presence of rare classes. A recent successful example is PNrule (Agarwal & Joshi 2001). SVMs sometimes perform better than PNrule, and sometimes worse. For example, Table 1 shows our SVM experimental performance on the king-rook-king problem from the UCI repository (Murphy & Aha 1992), when compared to previous experiments with PNrule (Joshi, Agarwal, & Kumar 2002). An SVM with a Gaussian kernel seems to perform better than PNrule on this problem. On the other hand,

Dataset	PNrule	Boosted PNrule	SVM
krkopt-sixteen	56.35	70.17	76.6
krkopt-five	63.48	65.84	63.4
krkopt-eight	52.74	61.84	66.3
krkopt-nine	43.35	59.11	65.8
krkopt-ten	42.08	54.63	56.2
krkopt-fifteen	66.07	72.09	73.3
krkopt-eleven	49.00	58.62	57.4
krkopt-thirteen	58.51	61.56	65.1
krkopt-fourteen	61.73	72.90	73.1

Table 1: **SVM test set F-measure on the king-rook-king dataset, compared with both PNrule and a boosted version of PNrule. SVM results are seen to be considerably better than PNrule, and comparable with boosted PNrule.**

SVMs performed poorly in our experiments under a variety of standard kernels when compared with synthetic datasets designed to highlight the strengths of PNrule (Joshi, Agarwal, & Kumar 2001). These synthetic datasets are particularly hard for a standard SVM to handle, as the features that characterize the rare class are different from the features that characterize the non-rare class. The theoretical results that we provide here yield evidence that the lack of success on these datasets by SVMs is not due to an inability to maximize F-measure, but due to some other intrinsic difficulty in using an SVM to fit the data. We therefore provide further evidence that the two-phase technique that PNrule uses is doing something new and different, and perhaps such a technique could be wrapped around an SVM.

Conclusions and Future Work

We have provided a framework and yielded new insights into what is accomplished when support vector machines are used to optimize F-measure on a dataset. We have also provided new theoretical evidence that heuristic techniques that are popular within the data mining community are a worthwhile endeavor.

Future research directions include integrating feature selection techniques in conjunction with SVMs to compete further with PNrule, as well as looking more directly at how to modify SVMs to optimize F-measure without the need for adjusting parameters.

Acknowledgments

This work was performed during the summer of 2002 via support from the University of Minnesota Army High Performance Computing Research Center.

References

Agarwal, R., and Joshi, M. V. 2001. PNrule: A new framework for learning classifier models in data mining. In *Proceedings of First SIAM International Conference on Data Mining*.

Chen, C., and Mangasarian, O. L. 1996. Hybrid misclassification minimization. *Advances in Computational Mathematics* 5(2):127–136.

Collobert, R., and Bengio, S. 2001. SVMTorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research* 1(1):143–160. `http://www.jmlr.org`.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine Learning* 20(3):273–297.

Cristianini, N., and Shawe-Taylor, J. 2000. *An Introduction to Support Vector Machines*. Cambridge: Cambridge University Press.

Hand, D.; Mannila, H.; and Smyth, P. 2001. *Principles of Data Mining*. Cambridge, MA: MIT Press.

Joachims, T. 1999. Making large-scale support vector machine learning practical. In Schölkopf, B.; Burges, C. J. C.; and Smola, A. J., eds., *Advances in Kernel Methods - Support Vector Learning*, 169–184. MIT Press.

Joshi, M. V.; Agarwal, R. C.; and Kumar, V. 2001. Mining needles in a haystack: Classifying rare classes via two-phase rule induction. In *Proceedings of the ACM SIGMOD 2001 Conference on Management of Data*.

Joshi, M. V.; Agarwal, R. G.; and Kumar, V. 2002. Predicting rare classes: Can boosting make any weak learner strong? In *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*.

Mangasarian, O. L., and Musicant, D. R. 1999. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks* 10:1032–1037.

Mangasarian, O. L., and Musicant, D. R. 2001. Lagrangian support vector machines. *Journal of Machine Learning Research* 1:161–177.

Mangasarian, O. L. 1994a. Misclassification minimization. *Journal of Global Optimization* 5:309–323.

Mangasarian, O. L. 1994b. *Nonlinear Programming*. Philadelphia, PA: SIAM.

Mangasarian, O. L. 1996. Machine learning via polyhedral concave minimization. In Fischer, H.; Riedmueller, B.; and Schaeffler, S., eds., *Applied Mathematics and Parallel Computing – Festschrift for Klaus Ritter*. Germany: Physica-Verlag. 175–188.

Morik, K.; Brockhausen, P.; and Joachims, T. 1999. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proceedings of the International Conference on Machine Learning*.

Murphy, P. M., and Aha, D. W. 1992. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Schölkopf, B. 2000. The kernel trick for distances. TR MSR 2000-51, Microsoft Research, Redmond, WA.

van Rijsbergen, C. 1979. *Information Retrieval*. London: Butterworths.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. New York: Springer.