Margin Based Feature Selection - Theory and Algorithms

Ran Gilad-Bachrach[†] RANB@CS.HUJI.AC.IL
Amir Navot[‡] ANAVOT@CS.HUJI.AC.IL
Naftali Tishby[†,‡] TISHBY@CS.HUJI.AC.IL
[†]School of Computer Science and Engineering, [‡]Interdisciplinary Center for Neural Computation.
The Hebrew University, Jerusalem, Israel

Abstract

Feature selection is the task of choosing a small set out of a given set of features that capture the relevant properties of the data. In the context of supervised classification problems the relevance is determined by the given labels on the training data. A good choice of features is a key for building compact and accurate classifiers. In this paper we introduce a margin based feature selection criterion and apply it to measure the quality of sets of features. Using margins we devise novel selection algorithms for multi-class classification problems and provide theoretical generalization bound. We also study the well known *Relief* algorithm and show that it resembles a gradient ascent over our margin criterion. We apply our new algorithm to various datasets and show that our new *Simba* algorithm, which directly optimizes the margin, outperforms *Relief*.

1. Introduction

In many supervised learning tasks the input data is represented by a very large number of features, but only few of them are relevant for predicting the label. Even state-of-art classification algorithms (e.g. SVM (Cortes & Vapnik, 1995)) cannot overcome the presence of large number of weakly relevant and redundant features. This is usually attributed to "the curse of dimensionality" (Bellman, 1961), or to the fact that irrelevant features decrease the signal-to-noise ratio. In addition, many algorithms become computationally intractable when the dimension is high. On the other hand once a good small set of features has been chosen even the most basic classifiers (e.g. 1-Nearest Neighbor (Fix & Hodges, 1951)) can achieve high performance levels. Therefore *feature selection*, i.e. the task of choosing a small subset of features which is sufficient to predict the target labels well, is crucial for efficient learning.

Feature selection is closely related to the more general problems of dimensionality reduction and efficient data representation. Many dimensionality reduction methods, like *Principal Component Analysis* (Jolliffee, 1986) or *Locally Linear Embedding* (Roweis & Saul, 2000), are in fact unsupervised *feature extraction* algorithms, where the obtained lower dimensions are not necessarily subsets of the original coordinates. Other methods, more related to supervised feature extraction, are the *Information Bottleneck* (Tishby et al., 1999) and *Sufficient Dimensionality Reduction* (Globerson & Tishby, 2003). However, on many cases, feature selection algorithms provide a much simpler approach as they do not require the evaluation of new complex functions of the irrelevant features.

Roughly speaking, supervised feature selection methods are applied in one of two conceptual frameworks: the *filter model* and the *wrapper model* (Kohavi & John, 1997). In the wrapper model the selection method tries to directly optimize the performance of a specific predictor (algorithm). This may be done by estimating the predictor generalization performance (e.g. by cross validation) for the selected feature set in each step. The main drawback of this method is its computational deficiency.

In the filter model the selection is done as a *preprocessing*, without trying to optimize the performance of any specific predictor directly. This is usually achieved through an (ad-hoc) *evaluation function* using a search method in order to select a set that maximizes this function. Performing an exhaustive search is usually intractable due to the large number of initial features. Different methods apply a variety of search heuristics, such as hill climbing and genetic algorithms. One commonly used evaluation function is the mutual informa-

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

tion between the feature set and the labels (Quinlan, 1990). See (Guyon & Elisseeff, 2003) for a comprehensive discussion of feature selection methodologies.

In this paper we introduce the idea of measuring the quality of a set of features by the margin it induces. A margin (Cortes & Vapnik, 1995; Schapire et al., 1998) is a geometric measure for evaluating the confidence of a classifier with respect to its decision. Margins already play a crucial role in current machine learning research. The novelty of this paper is the use of large margin principle for feature selection[1].

Throughout this paper we will use the 1-NN as the "study-case" predictor, but most of the results are relevant to other distance based classifiers (e.g. LVQ (Kohonen, 1995), SVM-RBF (Cortes & Vapnik, 1995)) as well. The margin for these kind of classifiers was previously defined in (Crammer et al., 2002). The use of margins allows us to devise new feature selection algorithms as well as prove a PAC style generalization bound. The bound is on the generalization accuracy of 1-NN on a selected set of features, and guarantees good performance for any feature selection scheme which selects small set of features while keeping the margin large. On the algorithmic side, we use a margin based criteria to measure the quality of sets of features. We present two new feature selection algorithms, a Greedy Feature Flip (*G-flip*) and an Iterative Search Margin Based Algorithm which we call *Simba*, based on this criteria. The merits of these algorithms is demonstrated on a synthetic data and a face classification task.

The paper is organized as follows: Section 2 discusses margins in machine learning and presents our new margin based criterion for feature selection. In section 3 we present two new feature selection algorithms *G-flip* and *Simba* and compare them to the *Relief* algorithm. A theoretical generalization analysis is presented in section 4. Empirical evidence on the performance of these algorithms is provided in section 5, followed by concluding discussion in section 6.

2. Margins

Margins play a crucial role in modern machine learning research. They measure the classifier confidence when making its decision. Margins are used both for theoretic generalization bounds and as guidelines for algorithm design.

[1](Weston et al., 2000) devised a wrapper feature selection algorithm for SVM, and thus used margin for feature selection indirectly

2.1. Two types of Margins

As described in (Crammer et al., 2002) there are two natural ways of defining the margin of an instance with respect to a classification rule. The more common type, *sample-margin*, measures the distance between the instance and the decision boundary induced by the classifier. *Support Vector Machines* (Cortes & Vapnik, 1995), for example, finds the separating hyper-plane with the largest sample-margin. Bartlett (1998), also discusses the distance between instances and the decision boundary. He uses the *sample-margin* to derive generalization bounds.

An alternative definition, the *hypothesis-margin*, requires the existence of a distance measure on the hypothesis class. The margin of an hypothesis with respect to an instance is the distance between the hypothesis and the closest hypothesis that assigns alternative label to the given instance. For example *AdaBoost* (Freund & Schapire, 1997) uses this type of margin with the L_1-norm as the distance measure among hypotheses.

Throughout this paper we will be interested in margins for 1-NN. For this special case, (Crammer et al., 2002) proved the following two results:

1. The hypothesis-margin lower bounds the sample-margin.
2. It is easy to compute the hypothesis-margin of an instance x with respect to a set of points P by the following formula:

$$\theta_P(x) = \frac{1}{2}\Big(\|x - \mathbf{nearmiss}(x)\| - \|x - \mathbf{nearhit}(x)\| \Big)$$

where $\mathbf{nearhit}(x)$ and $\mathbf{nearmiss}(x)$ denote the nearest point to x in P with the same and different label, respectively. Note that a chosen set of features affects the margin through the distance measure.

Therefore in the case of Nearest Neighbor large hypothesis-margin ensures large sample-margin, and hypothesis-margin is easy to compute.

2.2. Margin Based Evaluation Function

A good generalization can be guaranteed if many sample points have large margin (see section 4). We introduce an evaluation function which assigns a score to sets of features according to the margin they induce. First we formulate the margin as a function of the selected set of features.

Definition 1 *Let P be a set of points and x be an instance. Let w be a weight vector over the feature set, then the* margin *of x is*

$$\theta_P^w = \frac{1}{2}\left(\|x - \textbf{\textit{nearmiss}}(x)\|_w - \|x - \textbf{\textit{nearhit}}(w)\|_w\right) \quad (1)$$

where $\|z\|_w = \sqrt{\sum_i w_i^2 z_i^2}$.

Definition 1 extends beyond feature selection and allows weight over the features. When selecting a set of features F we can use the same definition by identifying F with its indicating vector. Therefore, we use the notation $\theta_P^F(x)$ for $\theta_P^{I_F}(x)$ where I_F is one for any feature in F and zero otherwise.

Since $\theta^{\lambda w}(x) = |\lambda|\theta^w(x)$ for any scalar λ, it is natural to introduce some normalization factor. The natural normalization is to require $\max w_i^2 = 1$, since it guaranties that $\|z\|_w \leq \|z\|$ where the right hand side is the Euclidean norm of z.

Now we turn to define the evaluation function. The building blocks of this function are the margins of all the sample points. The margin of each instance x is calculated with respect to the sample excluding x ("leave-one-out margin").

Definition 2 *Given a training set S and a weight vector w, the* evaluation function *is:*

$$e(w) = \sum_{x \in S} \theta_{S \setminus x}^w(x) \quad (2)$$

It is natural to look at the evaluation function only for weight vectors w such that $\max w_i^2 = 1$. However, formally, the evaluation function is well defined for any w and fulfills $e(\lambda w) = |\lambda| e(w)$, a fact which we make use of in the *Simba* algorithm. We also use the notation $e(F)$, where F is a set of features to denote $e(I_F)$.

3. Algorithms

In this section we present two algorithms which attempts to maximize the margin based evaluation function. Both algorithms can cope with multi-class problems[2].

3.1. Greedy Feature Flip Algorithm (G-flip)

G-flip (algorithm 1) is a greedy search algorithm for maximizing $e(F)$, where F is a set of features. The algorithm repeatedly iterates over the feature set and updates the set of chosen features. In each iteration it decides to remove or add the current feature to the selected set by evaluating the margin term (2) with and without this feature. This algorithm is similar to the zero-temperature Monte-Carlo (Metropolis) method. It converges to a local maximum of the evaluation function, as each step increases its value and the number of possible feature sets is finite. The computational complexity of one pass over all features of *G-flip* is $\Theta\left(N^2 m^2\right)$ where N is the number of features and m is the number of instances. Empirically *G-flip* converges in a few iterations. In all our experiments it converged after less than 20 epochs, in most of the cases in less than 10 epochs. A nice property of this algorithm is that it is *parameter free*. There is no need to tune the number of features or any type of threshold.

Algorithm 1 Greedy Feature Flip (G-flip)

1. Initialize the set of chosen features to the empty set: $F = \phi$
2. for $t = 1, 2, \ldots$
   (a) pick a random permutation s of $\{1 \ldots N\}$
   (b) for $i = 1$ to N,
      i. evaluate $e_1 = e\left(F \cup \{s(i)\}\right)$ and $e_2 = e\left(F \setminus \{s(i)\}\right)$
      ii. if $e_1 > e_2$, $F = F \cup \{s(i)\}$ else-if $e_2 > e_1$, $F = F \setminus \{s(i)\}$
   (c) if no change made in step (b) then break

3.2. Iterative Search Margin Based Algorithm (Simba)

The *G-flip* algorithm presented in section 3.1 tries to find the feature set that maximizes the margin directly. Here we take another approach. We first find the weight vector w that maximizes $e(w)$ as defined in (2) and then use a threshold in order to get a feature set. Of course, it is also possible to use the weights directly by using the induced distance measure instead. Since $e(w)$ is smooth almost everywhere, we use gradient ascent in order to maximize it. The gradient of $e(w)$ when evaluated on a sample S is:

$$\begin{aligned}(\bigtriangledown e(w))_i &= \frac{\partial e(w)}{\partial w_i} = \sum_{x \in S} \frac{\partial \theta(x)}{\partial w_i} \quad (3)\\ &= \frac{1}{2}\sum_{x \in S}\Big(\frac{\left(x_i - \textbf{nearmiss}(x)_i\right)^2}{\|x - \textbf{nearmiss}(x)\|_w} \\ &\qquad - \frac{\left(x_i - \textbf{nearhit}(x)_i\right)^2}{\|x - \textbf{nearhit}(x)\|_w}\Big) w_i\end{aligned}$$

[2] A Matlab code of these algorithms is available at: *www.cs.huji.ac.il/labs/learning/code/feature_selection*

Algorithm 2 Simba

1. initialize $w = (1, 1, \ldots, 1)$
2. for $t = 1 \ldots T$
   (a) pick randomly an instance x from S
   (b) calculate $\mathbf{nearmiss}(x)$ and $\mathbf{nearhit}(x)$ with respect to $S \setminus \{x\}$ and the weight vector w.
   (c) for $i = 1, \ldots, N$ calculate

$$\triangle_i = \frac{1}{2}\Big(\frac{(x_i - \mathbf{nearmiss}(x)_i)^2}{\|x - \mathbf{nearmiss}(x)\|_w} - \frac{(x_i - \mathbf{nearhit}(x)_i)^2}{\|x - \mathbf{nearhit}(x)\|_w}\Big) w_i$$

   (d) $w = w + \triangle$
3. $w \leftarrow w^2 / \left\|w^2\right\|_\infty$ where $(w^2)_i := (w_i)^2$.

In *Simba* (algorithm 2) we use a stochastic gradient ascent over $e(w)$ while ignoring the constraint $\|w^2\|_\infty = 1$, the projection on the constraint is done only at the end (step 3). This is sound since $e(\lambda w) = |\lambda| e(w)$. In each iteration we evaluate only one term in the sum in (3) and add it to the weight vector w. Note that the term Δ evaluated in step 2(c) is invariant to scalar scaling of w (i.e. $\Delta(w) = \Delta(\lambda w)$ for any $\lambda > 0$). Therefore, since $\|w\|$ increases, the relative effect of the correction term Δ decreases and the algorithm typically convergence.

The computational complexity of *Simba* is $\Theta(TNm)$ where T is the number of iterations, N is the number of features and m is the size of the sample S. Note that when iterating over all training instances, i.e. when $T = m$, the complexity is $\Theta\left(Nm^2\right)$ which is better than *G-flip* by a factor of N.

3.3. Comparison to Relief

Relief (Kira & Rendell, 1992) is a feature selection algorithm (see algorithm 3), which was shown to be very efficient for estimating features quality. The algorithm holds a weight vector over all features and updates this vector according to the sample points presented. Kira & Rendell (1992) proved that under some assumptions, the expected weight is large for relevant features and small for irrelevant ones. They also explain how to choose the relevance threshold τ in a way that ensures the probability that a given irrelevant feature will be chosen is small. *Relief* was extended to deal with multi-class problems, noise and missing data by Kononenko (1994).

Algorithm 3 RELIEF (Kira & Rendell, 1992)

1. initiate the weights vector to zero: $w = 0$
2. for $t = 1 \ldots T$,
   (a) pick randomly an instance x from S
   (b) for $i = 1 \ldots N$,
      i. $w_i = w_i + (x_i - \mathbf{nearmiss}(x)_i)^2 - (x_i - \mathbf{nearhit}(x)_i)^2$
3. the chosen feature set is $\{i | w_i > \tau\}$ where τ is a threshold

Note that the update rule in a single step of *Relief* is similar to the one performed by *Simba*. Indeed, empirical evidence shows that *Relief* does increase the margin (see section 5). However, there is a major difference: *Relief* does not re-evaluate the distances according to the weight vector w and thus it is inferior to *Simba*. In particular, *Relief* has no mechanism for eliminating redundant features. *Simba* may also choose correlated features, but only if this contributes to the overall performance. In terms of computational complexity, *Relief* and *Simba* are equivalent.

4. Theoretical Analysis

In this section we use feature selection and large margin principals to prove finite sample generalization bound for 1-*Nearest Neighbor*. (Cover & Hart, 1967), showed that asymptotically the generalization error of 1-NN can exceed by at most a factor of 2 the generalization error of the Bayes optimal classification rule. However, on finite samples nearest neighbor can overfit and exhibit poor performance. Indeed 1-NN will give zero training error, on almost any sample.

The training error is thus too rough to provide information on the generalization performance of 1-NN. We therefore need a more detailed measure in order to provide meaningful generalization bounds and this is where margins become useful. It turns out that in a sense, 1-NN is a maximum margin algorithm. Indeed once our proper definition of margin is used, i.e. sample-margin, it is easy to verify that 1-NN generates the classification rule with the largest possible margin.

The combination of a large margin and a small number of features provides enough evidence to obtain a useful bound on the generalization error. The bound we provide here is data-dependent (Shawe-Taylor et al., 1998; Bartlett, 1998). Therefore, the quality of the bound depends on our specific sample. It holds simultaneously for any possible method to select a set of

features. If an algorithm selects a small set of features with large margin, the bound guarantees it generalizes well. This is the motivation for *Simba* and *G-flip*.

We use the following notation:

Definition 3 *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{\pm 1\}$ and $h : \mathcal{X} \longrightarrow \{\pm 1\}$ a classification function. We denote by $er_{\mathcal{D}}(h)$ the generalization error of h with respect to $\mathcal{D}$:*

$$er_{\mathcal{D}}(h) \quad = \quad \Pr_{x,y\sim\mathcal{D}}[h(x) \neq y]$$

For a sample $S = \{(x_k, y_k)\}_{k=1}^{m} \in (\mathcal{X} \times \{\pm 1\})^m$ and a constant $\gamma > 0$ we define the γ-sensitive training error to be

$$\hat{er}_S^{\gamma}(h) \quad = \quad \frac{1}{m}\Big|\Big\{ (k : h(x_k) \neq y_k) \ \ or \ \ (x_k \ has \ sample\text{-}margin < \gamma) \Big\}\Big|$$

Our main result is the following theorem[3]:

Theorem 1 *Let $\mathcal{D}$ be a distribution over $\mathcal{R}^N \times \{\pm 1\}$ which is supported on a ball of radius R in $\mathcal{R}^N$. Let $\delta > 0$ and let S be a sample of size m such that $S \sim \mathcal{D}^m$. With probability $1 - \delta$ over the random choice of S, for any set of features F and any $\gamma \in (0, 1]$*

$$er_{\mathcal{D}}(h) \quad \leq \quad \hat{er}_S^{\gamma}(h) + \sqrt{\frac{2}{m}\left(d \ln\left(\frac{34em}{d}\right)\log_2(578m) + \ln\left(\frac{8}{\gamma\delta}\right) + (|F|+1)\ln N\right)} \tag{4}$$

Where h is the nearest neighbor classification rule when distance is measured only on the features in F and $d = (64R/\gamma)^{|F|}$.

The size of the feature space, N, appears only logarithmically in the bound. Hence, it has a minor effect on the generalization error of 1-NN. On the other hand, the number of selected features, F, appears in the exponent. This is another realization of the "curse of dimensionality" (Bellman, 1961). See appendix A for the proof of theorem 1.

5. Empirical Assessment

We first demonstrate the behavior of *Simba* on a small synthetic problem. Then we test it on a task of pixel (feature) selection for discriminating between male and female face images. For the *G-flip* algorithm, we report the results obtained on some of the datasets of the *NIPS-2003 feature selection challenge* (Guyon & Gunn, 2003).

[3]Note that the theorem holds when sample-margin is replaced by hypothesis-margin since the later lower bounds the former.

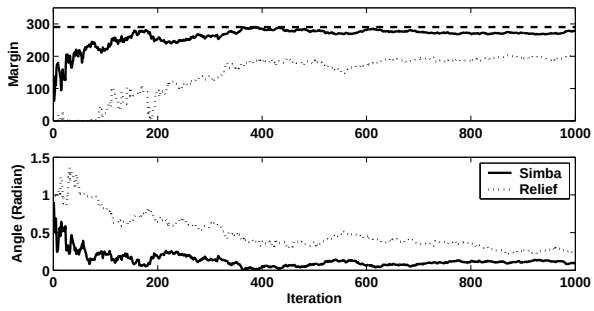

Figure 1. The results of applying *Simba* (solid) and *Relief* (dotted) on the *xor* synthetic problem. **Top:** The margin value, $e(w)$, at each iteration. The dashed line is the margin of the correct weight vector. **Bottom:** the angle between the weight vector and the correct feature vector at each iteration (in Radians).

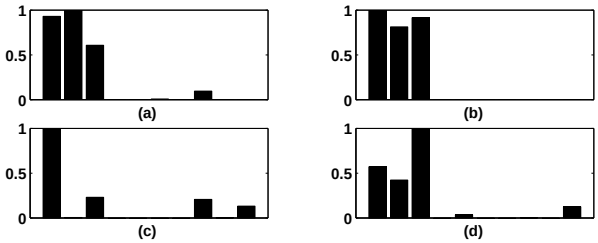

Figure 2. The weights *Simba* and *Relief* assign to the 10 features when applying on the *xor* problem. (a) and (b) are the weights obtained by *Simba* after 100 and 500 iterations respectively. (c) and (d) are the corresponding weights obtained by *Relief*. The correct weights are "1" for the first 3 features and "0" for all the others.

5.1. The Xor Problem

To demonstrate the quality of the margin based evaluation function and the ability of *Simba* algorithm to deal with dependent features we use a synthetic problem. The problem consisted of 1000 sample points with 10 real valued features. The target concept is a *xor* function over the first 3 features. Hence, the first 3 features are relevant while the other features are irrelevant. Notice that this task is a special case of parity function learning and is considered hard for many feature selection algorithms (Guyon & Elisseeff, 2003). Thus for example, any algorithm which does not consider functional dependencies between features fails on this task. Figures 1 and 2 present the results we obtained on this problem.

A few phenomena are apparent in these results. The value of the margin evaluation function is highly correlated with the angle between the weight vector and the correct feature vector (see figures 1 and 3). This correlation demonstrates that the margins character-

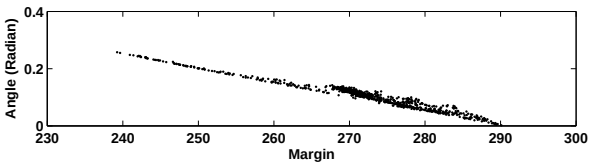

Figure 3. The scatter plot shows the angle to the correct feature vector as function of the value of the margin evaluation function. The values were calculated for the *xor* problem using *Simba* during iterations 150 to 1000. Notice the linear relation between the two quantities.

ize correctly the quality of the weight vector. This is quite remarkable since our margin evaluation function can be measured empirically on the training data whereas the angle to the correct feature vector is unknown during learning.

As suggested in section 3.3 *Relief* does increase the margin as well. However, *Simba* outperforms *Relief* significantly, as shown in figure 2.

5.2. Face Images

We applied the algorithms to the AR face database (Martinez & Benavente, 1998) which is a collection of digital images of males and females with various facial expressions, illumination conditions, and occlusions. We selected 1456 images and converted them to gray-scale images of 85×60 pixels, which are taken as our initial 5100 features. Examples of the images are shown in figure 4. The task we tested is classifying the male vs. the female faces.

In order to improve the statistical significance of the results, the dataset was partitioned independently 20 times into training data of 1000 images and test data of 456 images. For each such partitioning (split) *Simba*, *Relief* and *Infogain*[4] were applied to select optimal features and the 1-NN algorithm was used to classify the test data points. We used 10 random starting points for *Simba* (i.e. random permutations of the train data) and selected the result of the single run which reached the highest value of the evaluation function. The average accuracy versus the number of features chosen, is presented in figure 5.

Simba significantly outperformed *Relief* and *Infogain*, especially in the small number of features regime. When less than 1000 features were used *Simba* achieved better generalization accuracy than both *Relief* and *Infogain* in more than 90% of the partitions

[4] *Infogain* ranks features according to the mutual information between each feature and the labels. *G-flip* was not applied due to computational constraints.

Figure 4. Excerpts from the face images dataset.

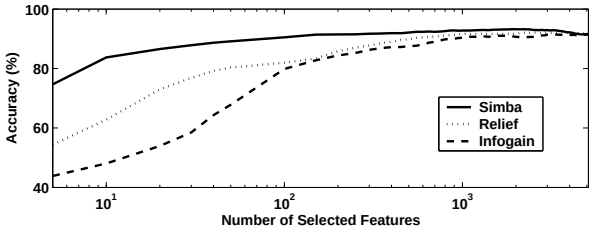

Figure 5. **Results for AR faces dataset.** The accuracy achieved on the AR faces dataset when using the features chosen by the different algorithms. The results were averaged over the 20 splits of the dataset. In order to validate the statistical significance we present the results on all the partitions in figure 6.

(figure 6). Moreover, the 1000 features that *Simba* selected enabled 1-NN to achieve accuracy of 92.8% which is better than the accuracy obtained with the whole feature set (91.5%). A closer look on the features selected by *Simba* and *Relief* (figure 7) reveals the difference between the two algorithms. *Relief* focused on the hair-line, especially around the neck, and on other contour areas in a left-right symmetric fashion. This choice is suboptimal as those features are highly correlated to each other and therefore a smaller subset is sufficient. *Simba* on the other hand selected features in other informative facial locations but mostly on one side (left) of the face, as the other side is clearly highly correlated and does not contribute new information to this task. Moreover, this dataset is biased in the sense that more faces are illuminated from the right. Many of them are saturated and thus *Simba* preferred the left side over the less informative right side.

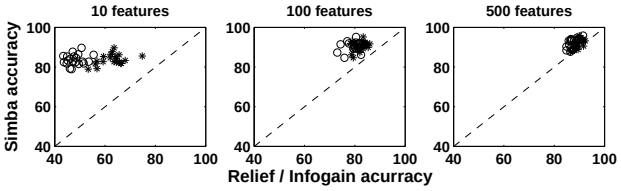

Figure 6. Accuracy of *Simba* vs. *Infogain* (circles) and *Relief* (stars) for each of the 20 partitions of the AR faces dataset. Note that any point above the diagonal means that *Simba* outperforms the alternative algorithm in the corresponding partition of the data.

(a) (b) (c) (d) (e) (f)

Figure 7. The features selected (in black) by *Simba* and *Relief* for the face recognition task. (a), (b) and (c) shows 100, 500 and 1000 features selected by *Simba*. (d), (e) and (f) shows 100, 500 and 1000 features selected by *Relief*.

5.3. The NIPS-03 Feature Selection Challenge

We applied *G-flip* as part of our experiments in the *NIPS-03 feature selection challenge* (Guyon & Gunn, 2003). It was applied on two datasets (ARCENE and MADELON) with both 1-NN and SVM-RBF classifiers. The obtained results were among the best submitted to the challenge. SVM-RBF gave better results than 1-NN, but the differences were minor. In the ARCENE data, the task was to distinguish between cancer and normal tissues gene-expression patterns. Each instance was presented by 10,000 features and there were 200 training examples. *G-flip* selected 76 features (when run after converting the data by PCA). SVM-RBF achieved balanced error rate of 12.66% using those features (the best result of the challenge on this data set was 10.76%). MADELON was a synthetic dataset. Each instance was represented by 500 features and there were 2600 training examples. *G-flip* selected only 18 features. SVM-RBF achieved 7.61% balanced error rate using these features (the best result on this dataset was 6.22%). A main advantage of our approach is its simplicity. For more information see the challenge results at *http://www.nipsfsc.ecs.soton.ac.uk/results*.

6. Summary and Further Research Directions

A margin-based criterion for measuring the quality of a set of features has been presented. Using this criterion we derived algorithms that perform feature selection by searching for the set that maximizes it. We have also showed that the well known *Relief* algorithm (Kira & Rendell, 1992) approximates a gradient ascent over this measure. We suggested two new methods for maximizing the margin based-measure, *G-flip* which does a naive local search, and *Simba* which performs a gradient ascent. These are just representatives of the variety of optimization techniques (search methods) which can be used. We have showed that *Simba* outperforms *Relief* on a face classification task and that it handles better correlated features. One of the main advantages of the margin based criterion is the high correlation that it exhibits with the features quality. This was demonstrated in figures 1 and 3.

Our main theoretical result is a new rigorous bound on the finite sample generalization error of the 1-*Nearest Neighbor* algorithm. This bound depends on the margin obtained following the feature selection.

Several research directions can be further investigated. One of them is to utilize a better optimization algorithm for maximizing our margin-based evaluation function. The evaluation function can be altered as well. It is possible to apply non-linear functions of the margin and achieve different tradeoffs between large margin and training error and thus better stability. It is also possible to apply our margin based criterion and algorithms in order to learn distance measures.

Another interesting direction is to link the feature selection algorithms to the LVQ (Kohonen, 1995) algorithm. As was shown in (Crammer et al., 2002), LVQ can be viewed as a maximization of the very same margin term. But unlike the feature selection algorithms presented here, LVQ does so by changing prototypes location and not the subset of the features. This way LVQ produces a simple but robust hypothesis. Thus, LVQ and our feature selection algorithms maximize the same margin criterion by controlling different (dual) parameters of the problem. In that sense the two algorithms are dual. One can combine the two by optimizing the set of features and prototypes location together. This may yield a winning combination.

References

Bartlett, P. (1998). The size of the wieghts is more important than the size of the network. *IEEE Transactions on Information Theory, 44*, 525–536.

Bellman, R. (1961). *Adaptive control processes: A guided tour*. Princeton University Press.

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

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classifier. *IEEE Trans. on Information Theory, 13*.

Crammer, K., Gilad-Bachrach, R., Navot, A., & Tishby, N. (2002). Margin analysis of the lvq algorithm. *Proc. 17'th Conference on Neural Information Processing Systems*.

Fix, E., & Hodges, j. (1951). *Discriminatory analysis. nonparametric discrimination: Consistency properties* (Technical Report 4). USAF school of Aviation Medicine.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*.

Globerson, A., & Tishby, N. (2003). Sufficient dimensionality reduction. *Journal of Machine Learning*, 1307–1331.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learnig Research*, 1157–1182.

Guyon, I., & Gunn, S. (2003). Nips feature selection challenge. http://www.nipsfsc.ecs.soton.ac.uk/.

Jolliffee, I. (1986). *Principal component analysis*. Springer Varlag.

Kira, K., & Rendell, L. (1992). A practical approach to feature selection. *Proc. 9th International Workshop on Machine Learning* (pp. 249–256).

Kohavi, R., & John, G. (1997). Wrapper for feature subset selection. *Artificial Intelligence*, *97*, 273–324.

Kohonen, T. (1995). *Self-organizing maps*. Springer-Verlag.

Martinez, A., & Benavente, R. (1998). *The ar face database* (Technical Report). CVC Tech. Rep. #24.

Quinlan, J. R. (1990). Induction of decision trees. In J. W. Shavlik and T. G. Dietterich (Eds.), *Readings in machine learning*. Morgan Kaufmann. Originally published in *Machine Learning* 1:81–106, 1986.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*.

Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1998). Boosting the margin : A new explanation for the effectiveness of voting methods. *Annals of Statistics*.

Shawe-Taylor, J., Bartlett, P., Williamson, R., & Anthony, M. (1998). Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory*, *44*, 1926–1940.

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing* (pp. 368–377).

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for SVMs. *Proc. 15th Conference on Neural Information Processing Systems (NIPS)* (pp. 668–674).

A. Complementary Proofs

We begin by proving a simple lemma which shows that the class of nearest neighbor classifiers is a subset of the class of 1-Lipschitz functions. Let $\mathbf{nn}_F^S(\cdot)$ be a function such that the sign of $\mathbf{nn}_F^S(x)$ is the label that the nearest neighbor rule assigns to x, while the magnitude is the sample-margin, i.e. the distance between x and the decision boundary.

Lemma 1 *Let F be a set of features and let S be a labeled sample. Then the for any $x_1, x_2 \in \mathcal{R}^N$:*

$$\left|\boldsymbol{nn}_F^S(x_1) - \boldsymbol{nn}_F^S(x_2)\right| \quad \leq \quad \|F(x_1) - F(x_2)\|$$

where $F(x)$ is the projection of x on the features in F.

The proof of this lemma is straightforward and is omitted due to space limitations. The main tool for proving theorem 1 is the following:

Theorem 2 *(Bartlett, 1998) Let $\mathcal{H}$ be a class of real valued functions. Let S be a sample of size m generated i.i.d. from a distribution $\mathcal{D}$ over $\mathcal{X} \times \{\pm 1\}$ then with probability $1-\delta$ over the choices of S, every $h \in \mathcal{H}$ and every $\gamma \in (0,1]$ let $d = fat_{\mathcal{H}}(\gamma/32)$:*

$$er_{\mathcal{D}}(h) \quad \leq \quad \hat{er}_S^{\gamma}(h) + \sqrt{\frac{2}{m}\left(d \ln\left(\frac{34em}{d}\right)\log_2(578m) + \ln\left(\frac{8}{\gamma\delta}\right)\right)}$$

We now turn to prove theorem 1:

Proof (of theorem 1): Let F be a set of features such that $|F| = n$ and let $\gamma > 0$. In order to use theorem 2 we need to compute the fat-shattering dimension of the class of nearest neighbor classification rules which use the set of features F. As we saw in lemma 1 this class is a subset of the class of 1-Lipschitz functions on these features. Hence we can bound the fat-shattering dimension of the class of NN rules by the dimension of Lipschitz functions.

Since $\mathcal{D}$ is supported in a ball of radius R and $\|x\| \geq \|F(x)\|$, we need to calculate the fat-shattering dimension of Lipschitz functions acting on points in $\mathcal{R}^n$ with norm bounded by R. The fat_γ-dimension of the 1-NN functions on the features F is thus bounded by the largest γ packing of a ball in $\mathcal{R}^n$ with radius R, which in turn is bounded by $(2R/\gamma)^{|F|}$.

Therefore, for a fixed set of features F we can apply to theorem 2 and use the bound on the fat-shattering dimension just calculated. Let $\delta_F > 0$ and we have according to theorem 2 with probability $1-\delta_F$ over sample S of size m that for any $\gamma \in (0,1]$

$$\text{er}_{\mathcal{D}}(\text{nearest-neighbor}) \quad \leq \quad \hat{\text{er}}_S^{\gamma}(\text{nearest-neighbor}) + \sqrt{\frac{2}{m}\left(d \ln\left(\frac{34em}{d}\right)\log_2(578m) + \ln\left(\frac{8}{\gamma\delta_F}\right)\right)} \qquad (5)$$

for $d = (64R/\gamma)^{|F|}$. By choosing $\delta_F = \delta / \left(N\binom{N}{|F|}\right)$ we have that $\sum_{F \subseteq [1\ldots N]} \delta_F = \delta$ and so we can apply the union bound to (5) and obtain the stated result. □