

Feature Selection Focused within Error Clusters

Sui-Yu Wang and Henry S. Baird

Computer Science & Engineering Dept., Lehigh University
19 Memorial Dr West, Bethlehem, PA 18017 USA

E-mail: syw2@lehigh.edu, baird@cse.lehigh.edu

Abstract

We propose a feature selection method that constructs each new feature by analysis of tight error clusters. This is a greedy, time-efficient forward selection algorithm that iteratively constructs one feature at a time, until a desired error rate is reached. The algorithm finds error clusters in the current feature space, then projects one tight cluster into the null space of the feature mapping, where a new feature that helps to classify these errors can be discovered. Tight error clusters indicate that the current features are unable to discriminate these samples. The approach is strongly data-driven and restricted to linear features, but otherwise general. Large scale experiments show that it can achieve a monotonically decreasing error rate within the feature discovery set, and a generally decreasing error rate on a distinct test set.

1. Introduction

For many classification problems in a vector space setting, the number of potentially useful subsets features is too large to be exhaustively searched [3]. Feature selection algorithms attempt to identify a set of features that are discriminating but few enough for classifiers to perform efficiently. Such methods can be divided into three categories: filters, wrappers, and embedded methods [11]. Among wrappers methods (e.g., Partial Least Square [18]) and embedded methods (e.g., our method), greedy methods are most popular. Forward selection and backward elimination each have their own advantages and weakness. Because some features may be more powerful when combined with certain other features [10, 7], backward selection may yield better performance, but at the expense of larger feature sets. Also, if the resulting feature set is reduced too far, performance may drop abruptly [12]. On the other

hand, forward selection usually ranks features and selects a smaller subset. A small set of features can also serve to predict models, visualization, and outlier detection [15].

Two important negative results constrain what's possible. The Ugly Duckling Theorem states that there is no universally best set of features [17]; thus features useful in one problem domain can be useless in another. The No Free Lunch Theorem states that there is no single classifier that works best for all problems [19, 20, 21]. Taken together, these suggest that it is essential to know *both* the problem and the classifier technology in order to identify a good set of features.

We propose a forward selection method that is based on empirical data and incrementally adds one new feature at a time, until the desired error rate is reached. Each new feature is constructed in a data-driven manner, not chosen from a preexisting set. We experiment with Nearest Neighbor (NN) classifiers because they can work on any distribution without prior knowledge of the parametric models of the distribution. Also, the NN classifier has the remarkable property that with unlimited number of prototypes, the error rate is never worse than twice the Bayes error in the two-category case [5].

We assume that we are working on a two-class problem and that we were given an initial set of features on which an NN classifier has been trained. Suppose the performance of this classifier is not satisfactory. We examine the distributions of erroneously classified samples and find there are almost always clusters containing both types of errors. These can be found by clustering algorithms or manifold algorithms [4]. Tight clusters indicate that the current feature set is not discriminating within these regions. Thus we believe that these clusters are good places to look for new features that will resolve some of these errors. We look for new features in the null space of the current feature mapping. This guarantees that any feature found in the null space is

orthogonal to all current features. As we show in Section 2, this method works well if the features are linear. We project only those samples in the selected tight error cluster to the null space, where we find a decision hyperplane and define the new feature to be a sample’s directed distance to the hyperplane. The reason for projecting only samples in certain cluster is because different error clusters may be the result of different decision boundary segments lost during the projection to the current feature set. Projecting only one cluster of errors to the null space and finding a hyperplane that separates these errors saves computation time and allows us to find more precise decision boundaries.

We conducted the experiments, described in Section 3, in a document image segmentation framework, by trying to classify pixels into machine-print or handwriting [2, 1]. A sequence of six classifiers trained with the augmented feature sets exhibits a monotonically decreasing error rate. The sixth augmented feature set dropped the error rate compared to the the first set by 31%.

2. Formal Definitions

We work on a 2-class problem. We assume that there is a source of labeled training samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$. Each sample \mathbf{x} is described by a large number D of real-valued features: i.e., $\mathbf{x} \in \mathbf{R}^D$. But D , we assume, is too large for use as a classifier feature space. We also have a much smaller set of d real-valued features $\mathbf{f}^d = \{f_1, f_2, \dots, f_d\}$, where $d \ll D$. Applying \mathbf{f}^d on sample \mathbf{x} we get a d -dimensional vector $\mathbf{f}^d(\mathbf{x}) = (x_1, x_2, \dots, x_d)$.

In the original sample space \mathbf{R}^D , there exists some kind of boundary between classes. After projecting the data by \mathbf{f}^d into \mathbf{R}^d , the boundary was not preserved well, and misclassification occurs. By projecting the data back to the null space \mathbf{R}^{D-d} , we restore information lost by \mathbf{f}^d , and can find a new feature independent of \mathbf{f}^d in this space. To do so we restrict the feature extractor to be linear, and that the given feature set is linearly independent.

The null space can be defined as $N(\mathbf{f}^d) = \{\mathbf{s} | \mathbf{f}^d(\mathbf{s}) = \mathbf{0}\}$. We give a brief introduction on how to find and project the data to the null space [13]. Given \mathbf{f}^d , a matrix that projects data from the sample space into the current feature space, a commonly used linear algebra method, the *singular value decomposition*, or SVD, can be used to find a set of vectors spanning the null space of \mathbf{f}^d . The SVD Theorem [8] states that a singular value decomposition of a matrix $d \times D \in \mathbf{R}^{d \times D}$ is a factorization $\mathbf{f}^d = U\Sigma V^T$, where $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p) \in \mathbf{R}^{d \times D}$ is a $d \times D$ diagonal

matrix with entries $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$, $p = \min(D, d)$ and both $U \in \mathbf{R}^{d \times d}$ and $V \in \mathbf{R}^{D \times D}$ are orthogonal matrices¹.

Let $\sigma_1 \geq \sigma_2 \geq \dots \sigma_r > 0$ be the positive singular values of \mathbf{f}^d . Then from the SVD theorem, we have

$$\begin{aligned} \mathbf{f}^d v_i &= \sigma_i u_i, & i &= 1, \dots, r \\ \mathbf{f}^d v_j &= 0, & j &= r + 1, \dots, D \end{aligned}$$

where $v \in \mathbf{R}^D$. Thus,

$$N(\mathbf{f}^d) = \text{span}\{v_{r+1}, \dots, v_D\}.$$

The columns of V corresponding to the zero singular values form an orthonormal basis of $N(\mathbf{f}^d)$ [6]. Then

$$P_{D-d} = VV^T$$

is the unique orthogonal projection onto $N(\mathbf{f}^d)$. Note that although V is not unique, P_{D-d} is [6]. To construct the new feature, we first find a separating hyperplane $\bar{\mathbf{w}}$ among the projected samples in the null space. The new feature was constructed by calculating a directed Euclidean distance of given samples to the hyperplane, $\bar{\mathbf{w}} \cdot \mathbf{x}$.

Naturally for real data we have no idea what the true decision boundary is, nor are we willing to work in the high dimensional space \mathbf{R}^D . We use the performance of the current classifier to guide our search. We identify clusters of errors in the feature space. Afterwards, we select one that is both “tight” and contains both types of errors. We measure tightness as average pairwise distance. We assume we draw the data randomly, that is, the drawn data fills the sample space with probability density functions similar to the underlying distribution. Thus a tighter cluster implies that by correctly classifying samples in the region, more errors are likely to be corrected. Different clusters of errors might come from different decision boundary segments. Thus we consider one cluster at a time, hoping to resolve *some* errors. We find a separating hyperplane using a linear discriminant analysis [9]. The algorithm is as follows.

Algorithm

Repeat

Draw sufficient data from the discovery set \mathbf{X} , project them into lower dimensional space by \mathbf{f}^d , then train and test NN on the data.

Find clusters of errors.

Repeat

Select a tight cluster containing both types of errors.

Draw more samples from \mathbf{X} into the cluster.

¹A square matrix U is *orthogonal* if $U^T U = U U^T = I$.

Project samples in the selected cluster back to the null space.

Find a separating hyperplane in the null space.

Construct a new feature and examine its performance.

Until the feature lowers the error rate sufficiently.

Add the feature to the feature set, and set $d = d + 1$

Until the error rate is satisfactory to the user.

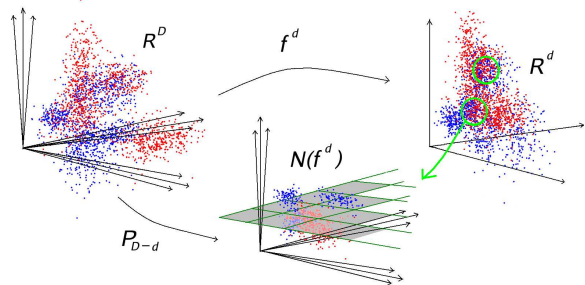


Figure 1: Projections between sample space \mathbf{R}^D , feature space \mathbf{R}^d , and null space $\mathbf{N}(f^d)$.

The algorithm is illustrated in Fig. 1. We begin with samples in \mathbf{R}^D , colored red and blue to indicate the two classes. These are projected, with f^d , into the current feature space \mathbf{R}^d , where clusters of errors are found and indicated by green circles. One cluster is chosen and projected into the null space $\mathbf{N}(f^d)$, as indicated by the green arrow. Finally a separating hyperplane is chosen in $\mathbf{N}(f^d)$.

Notice that there is no way of telling whether the errors are from the same region in the original space, or are clusters of errors that overlap because of f^d . However, clusters that span a smaller region with higher density are more likely to be from the same decision boundary lost in the original space. The clustering step, while allowing the method to adapt to real world data, involves more engineering choices than other parts of the algorithm.

3. Experiment

We conduct experiments in the document image content extraction framework [2, 1]. In this framework, each image pixel is treated as a sample. Possible features are extracted from a 25×25 pixels square, $D = 625$, with the classification result assigned to the center pixel. Pixels are represented by 8-bit greylevels. Possible content classes are handwriting (HW), machine print (MP), blank, and photos; for details see [2]. We chose MP and HW as our target classes.

We divide the data into three sets, training set, discovery set, and test set. After training, the classifier runs on the discovery set. A cluster with both types of errors is identified, and a new feature is discovered. The new feature is also tested on the test set to avoid potential overfitting on the discovery set. The training set consists of 4,469,740 MP samples and 943,178 HW samples. The test set consists of 816,673 MP samples and 649,113 HW samples. The feature discovery set consists of 4,980,418 MP and 1,496,949 HW samples.

We run the classifier with one manually chosen feature, followed by the procedure described in **Algorithm**. In addition, we normalize the length of vector \vec{w} because in the experiments, the length of the vector shrinks rapidly and may cause numerical instability. Each feature value is normalized to the interval $[0,255]$.

Fig. 2 shows the error rate of the six discovery sets and the test set. While the error rate on the discovery set is monotonically decreasing, the error rate on the test set is more erratic, but eventually decreasing. The initial error rate using the one manually chosen feature alone is 37.97%. The first discovered feature drop the error rate 50% to 18.9%. All six discovered features with the manually chosen feature drop the error rate 31% to 13.07%.

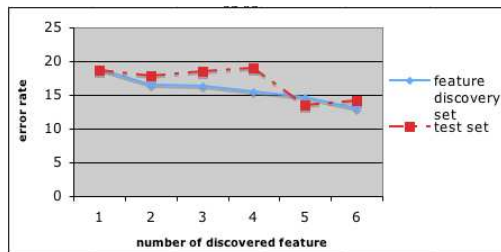


Figure 2: Error rate of discovery and test set.

Table 1 shows some statistics of the clusters selected. We use the k -means algorithm and assign 6-10 centers, according to the number of total errors in the discovery set. We notice that although it is not always the tightest cluster that gives us a useful new feature, a cluster that is too loose never gives an informative feature. The guideline to find a denser cluster only serves to suggest regions that are potentially interesting. On the other hand, according to information theory [16], clusters with approximately equal number of both types of error provide more information than unbalanced clusters. How to combine the two guidelines together is an open engineering problem.

Table 1: Statistics of the clusters.

	% of errors MP→HW	Avg. pairwise distance	Max pairwise distance
1st	48.2	4.60	13.00
2nd	48.2	12.27	45.18
3rd	40.3	28.12	142.06
4th	45.5	35.52	119.18
5th	37.4	49.70	151.11
6th	51.6	60.72	225.21

4. Discussion and Future Work

In this paper we present a forward feature selection algorithm guided by error clusters. Tight error clusters in the current feature space indicate that the current features are unable to distinguish samples in these regions. The algorithm works by projecting a tight error cluster found in the current feature space into the null space of the feature mapping, where new orthogonal features can be found. We then construct new features designed to correctly classify samples in these regions. We can draw an analogy between our approach and boosting methods [14]: boosting methods add weak classifiers that focus on previously misclassified samples, our approach instead add new features.

We observe that tighter clusters are more likely to yield discriminating features and that looser clusters never yield good features. Currently we choose clusters manually by ranking them according to three aspects: balanced errors of both classes, size of the cluster, and the tightness of the cluster, and test for useful new features. For future work, we wish to analyze quantitatively the effect of the three aspects. Although for this experiment we use a simple linear discriminant that assumes Gaussian distributions with a diagonal covariance matrix, the results seem to be promising. We were able to lower the error rate monotonically for the feature discovery set. For future work we wish to examine more complex assumption of data formation and look for separating hyperplane that couples with NN classifier to further improve the performance of the algorithm.

References

[1] H. S. Baird and M. R. Casey. Towards versatile document analysis systems. In *Document Analysis Systems*, pages 280–290, 2006.

[2] H. S. Baird, M. A. Moll, J. Nonnemaker, M. R. Casey, and D. L. Delorenzo. Versatile document image content

extraction. In *Proc., SPIE/IS&T Document Recognition & Retrieval XII Conf.*, San Jose, CA, January 2006.

[3] R. E. Bellman, editor. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.

[4] Y. Bengio, O. Delalleau, N. L. Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. *Feature Extraction: Foundations and Applications*, chapter Spectral Dimensionality Reduction, pages 519–549. Springer, 2003.

[5] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. on Information Theory*, 13:21–27, 1967.

[6] B. N. Datta. *Numerical Linear Algebra and Applications*. Brooks/Cole Publishing Company, 1994.

[7] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.

[8] J. Dongarra, V. Eijkhout, and J. Langou. *Handbook of Linear Algebra*. CRC press, 2006.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, 2nd Edition*. Wiley, New York, 2001.

[10] J. D. Elashoff, R. M. Elashoff, and G. Goldman. On the choice of variables in classification problems with dichotomous variables. *Biometrika*, 54:668–670, 1967.

[11] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[12] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, editors. *Feature Extraction: Foundations and Applications*. Studies in Fuzziness and Soft Computing. Springer, 2006.

[13] F. E. Hohn. *Elementary Matrix Algebra*. Dover Publications; 3rd edition, Jan 27 2003.

[14] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *In Advances in Neural Information Processing Systems 12*, pages 512–518. MIT Press, 2000.

[15] M. Momma and K. P. Bennett. *Feature Extraction: Foundations and Applications*, chapter Constructing Orthogonal Latent Features for Arbitrary Loss, pages 551–585. Springer, 2003.

[16] C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. Journal*, 27, 1948.

[17] S. Watanabe. *Pattern Recognition: Human and Mechanical*. Wiley, New York, 1985.

[18] H. Wold. The fix-point approach to interdependent systems. 1981).

[19] D. H. Wolpert, editor. *The Mathematics of Generalization*. Addison-Wesley, Reading, MA, 1995.

[20] D. H. Wolpert and W. G. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe, NM, 1995.

[21] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997.