Discriminative k-metrics

Arthur Szlam ASZLAM@MATH.UCLA.EDU
Department of Mathematics, UCLA

Guillermo Sapiro GUILLE@UMN.EDU
Electrical and Computer Engineering, University of Minnesota

Abstract

The k q-flats algorithm is a generalization of the popular k-means algorithm where q dimensional best fit affine sets replace centroids as the cluster prototypes. In this work, a modification of the k q-flats framework for pattern classification is introduced. The basic idea is to replace the original reconstruction only energy, which is optimized to obtain the k affine spaces, by a new energy that incorporates discriminative terms. This way, the actual classification task is introduced as part of the design and optimization. The presentation of the proposed framework is complemented with experimental results, showing that the method is computationally very efficient and gives excellent results on standard supervised learning benchmarks.

1. Introduction

The k q-flats algorithm, (Kambhatla & Leen, 1993; Mangasarian, 1998; Tseng, 1999; Cappelli et al., 2001), is a generalization of the k-means algorithm where we consider q-dimensional affine spaces ("flats") instead of points as prototypes. Thus, given a set of n points $X \subset \mathbb{R}^d$, we wish to find k q-dimensional flats $\{F_1, \ldots, F_k\}$ and a partition of X into $\{K_1, \ldots, K_k\}$, minimizing the energy

$$\sum_{j=1}^{k} \sum_{x \in K_j} ||x - P_{F_j} x||^2, \tag{1}$$

where $P_{F_j} x$ is the projection of the point x onto the affine space F_j. The minimization can be done using Lloyd's algorithm (EM):

- Initialize flats F_j, randomly or otherwise.
- Partition X into sets K_j by assigning each x to $$arg\min_j ||x - F_j||.$$
- Update the F_j by finding the L^2 best fit affine q-flat through the points in K_j.
- Repeat until convergence.

Just as for k-means, the Lloyd algorithm is guaranteed to converge to a local minimum.

In this paper we propose to learn q-flats in a supervised fashion, and modify energy (1) to include the task of classification. Before that, we proceed to present the interesting connection of k q-flats with sparse modeling and piecewise manifold approximation, and discuss the introduction of supervised learning in these approaches, further motivating our proposed framework.

1.1. k q-flats and sparse representation

Many signal processing problems are attacked using transform coding, which leverages the sparsity of the signals of interest, perhaps images or audio, in special bases, such as wavelets. In recent years, attention has been focused on adaptively choosing a basis so that a given class of signals has sparse coordinates. If such a basis can be found, the classical signal processing techniques can be applied even though the signals under study are not well represented by the classical basis functions.

A standard version of the sparse representation problem goes as follows: given n vectors $x \in \mathbb{R}^d$, which we consider as columns of a $d \times n$ matrix denoted by X, find an $d \times l$ basis B (again written as columns) and $l \times n$ matrix of coefficients R minimizing the error

$$||BR - X||_{\text{FRO}} \tag{2}$$

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

given that

$$||R_i||_0 \leq q, \tag{3}$$

where $||\cdot||_{\text{FRO}}$ is the Frobenious norm on matrices, i.e. L^2 on the matrix considered as a vector, R_i is the ith column of R, and $||\cdot||_0$ is the L^0 pseudo-norm, i.e. the number of nonzero entries. Several authors have noticed that the sparse representation problem is very closely related to the clustering problem, for example see (Tropp, 2004). In fact, if one restricts the entries of R to be either zeros or ones, and sets $q = 1$, the problem as written above is exactly k-means (Aharon et al., 2006). If we consider only q flats passing through the origin[1], the k-q-flats algorithm is an attempt at a minimizing (2) subject to (3) and an additional constraint forcing R to have a block structure. Each block corresponds to a K_j, as the elements in K_j are only allowed coefficients using basis elements in F_j; in this case, the number of atoms in B satisfies $l = kq$.

1.2. Piecewise linear manifold approximation

A recent trend in machine learning has been to use the fact that data often has an intrinsic dimensionality which is far smaller than the dimension of the ambient space in which it is presented. One often speaks of the data lying on a low dimensional manifold. A salient feature of a set with manifold structure is that it should be locally well approximated by affine spaces, namely the tangent spaces. If we know that there is some good approximation of the set by affine spaces, it makes sense to try to find the best one given some information theoretic constraints. This is exactly the k q-flats construction; the k (number of planes) and the q (dimension of each plane) serve as bounds on the capacity, and "best" is interpreted in the sense of mean square distance from the approximating secant planes. From this perspective, the construction can be considered to perform a locally linear dimension reductions, where the neighborhoods are chosen jointly with the projections so that the distortion from the projections are as small as possible.

Thus the k q-flats construction can be thought of as a method of building adapted dictionaries, which gives a notion of the distance from a point to set of points in such a way that meaningful information is abstracted; and of finding a best piecewise approximation to a data manifold, and so parameterizing a given data set. It thus sits in the intersection of the frameworks of sparse representation and manifold learning (for a nice look at the interplay between these sets of ideas, see (Wakin, 2006)), two of the most successful signal processing paradigms in recent years. In this article we will demonstrate the k q-flats utility for supervised learning and show how it can be modified to better serve this purpose.

[1] While this seems to be a big restriction, any q dimensional affine set is contained in a $q+1$ dimensional linear subspace, and so one often does not loose too much intuition by thinking of the F_j as actual subspaces, especially when the ambient dimension is larger

1.3. Sparse representation and q-flats for classification

If we have a supervised learning problem where the classes are expected to have fundamentally different representations, we can consider using the k q-flats algorithm for classification by training a dictionary of planes for each class. Given the set $X \subset \mathbb{R}^d$ consisting of n points with labels $1, ..., m$ (i.e., a partition of X into m subsets $C_1, ..., C_m$), the simplest supervised k-q flats algorithm (Cappelli et al., 2001) associates planes F_{ij} to the sets K_{ij} minimizing the energy

$$\sum_{i=1}^{m} \sum_{j=1}^{k} \sum_{x \in K_{ij}} ||x - P_{F_{ij}} x||^2. \tag{4}$$

Here, for a fixed i, the sets K_{ij}, where $j \in \{1, 2, ..., k\}$, form a partition of C_i. Given a new point to classify, we assign it to the class associated to its nearest flat.

There are many instances where different classes can be expected to have different representations, for example, sets of patches of different textures in images and sets of short clips of different musical instruments. In general, as mentioned above, one might expect "manifold" type data to be well approximated by the k-q-flats algorithm, as it would be searching for an optimal set of k secant planes, given a bound on the capacity q of each of the planes. The key here for classification is of course that the different classes are compressible in different dictionaries, not just that the data as a whole is compressible. To illustrate what could go wrong, consider the simplest example of manifold like data, say a line in $\mathbb{R}^d$, and a two class problem, say everything on one side of the line is class one, and on the other is class two. This would be a serious challenge for the above mentioned supervised k q-flats. However, while in low dimensions we imagine planes as intersecting, in high dimensions the vast amount of space makes this unlikely, and thus situations as this one should be unusual; it is simply a result of the data being linearly embeddable in a very low dimensional space.

In contrast, the *digit1* data set from (Chapelle et al., 2006), which consists of various images of the numeral one at different angles, jittered, and noised, and has

classes determined by the angular variable (positive angle is class one, and negative angle is class two), is a much more realistic example of what data parameterized by a single variable looks like. Although topologically it is identical to the previous example, as a function of angle, the curve traces out a "Hilbert spiral," that is, each "tangent" line points in a new direction. In particular, the best fit 1-flats passing through different pieces of the set do not intersect, and the angle of the 1 is correlated with the local best fit directions. In many interesting situations, we may even expect the different classes to be concentrated about different manifolds. In fact, experimentally, the supervised k q-flats gives excellent classification results on many data sets, especially relative to its speed and the simplicity of the approach (the entire code can be written in just a few lines in Matlab).

However, there is much room for improvement over reconstruction-only dictionaries for classification tasks, as shown for example for the case of sparse modeling in (Mairal et al., 2008). The basic k q-flats algorithm is representational: it does not try to explicitly encode the differences between the classes. In this work we will rectify this by changing the energy functional (4) to punish configurations of the flats passing through one class that get too close to points in another class. The idea is to get flats that represent one class well, but the other classes poorly. In addition to changing the energy functional to make the k q-flats framework discriminative, we generalize the affine spaces that are the class representatives in the k q-flats method, which can be thought of as very simple positive semi-definite matrices, to *Mahalanobis metrics*. We experimentally demonstrate that these changes greatly improve the classification accuracy of the method, and in fact lead to a computationally very efficient classifier with state of the art performance on several standard learning benchmarks.

2. A discriminative k q-metrics framework

The energy in Equation (4) does not explicitly see any information about the differences between the classes; it strictly measures representation errors. If we want to use flats for classification, we should modify this energy so that it penalizes a flat associated to a given class being too close to points from another class.

We will do this; however we also generalize the prototype we associate to a set of points from a flat to a positive semidefinite matrix A, or a Mahalanobis metric. If $x, y \in \mathbb{R}^d$, a Mahalanobis metric modifies

$$||x-y||^2 = (x-y)^T(x-y)$$

to

$$||x-y||_A^2 = (x-y)^T A(x-y),$$

where A is a positive semi-definite $d \times d$ matrix. The distance associated to A linearly crushes some directions more than others. Note that projecting onto a flat F passing through the origin is a very simple choice of A, namely $A = P_F$. Several recent papers have studied methods for finding Mahalanobis metrics to improve classification by k-nn; see (Weinberger et al., 2006; Davis et al., 2007; Xing et al., 2003) and the references therein.

In this work the metrics will be used as representatives for sets of points rather than a means of comparing single points with each other directly, as in the papers mentioned above. We use metrics instead of flats for two reasons. The first is that it allows a more flexible and richer set of classifiers. The second is that the constraint that all the eigenvalues of A are one or zero (as they would be if $A = P_F$ is computationally burdensome in our framework.

If we restrict ourselves to the situation where the flats are constrained to pass through the origin (the k-subspaces algorithm), the square distance of a point x to a plane F is given by

$$\begin{aligned} ||x - P_F x||^2 &= ||x - F^T F x||^2 \\ &= ||x||^2 - ||Fx||^2, \end{aligned}$$

where here and for the rest of this article, we abuse notation and allow F to refer to both the q-plane and a set of q vectors spanning the q-plane written as rows. Because the norm of each x is fixed, minimizing energy (4) is equivalent to minimizing

$$\sum_{i=1}^{m} \sum_{j=1}^{k} \sum_{x \in K_{ij}} -||F_{ij}x||^2.$$

This motivates us to consider energies of the following form:

$$E(\mathcal{K}, \mathcal{F}) = \tag{5}$$
$$\sum_{i=1}^{m} \sum_{j=1}^{k} \left[\sum_{x \in K_{ij}} g_1(||F_{ij}x||^2) + \sum_{x \notin C_i} g_2(||F_{ij}x||^2) \right],$$

where $K_{i,\cdot}$ is a partition of class i, $\mathcal{K} = \{K_{ij}\}$ is the collection of all these sets, $F_{i,j}$ is the set of basis vectors associated to K_{ij}, and $\mathcal{F} = \{F_{ij}\}$ is the collection of all these vectors. We will not force the elements of an

$F_{i,j}$ to be orthogonal, and since $||Fx||^2$ is the norm of x in the Mahalanobis metric given by $A = F^T F$, "k-subspaces" is replaced by discriminative "k-metrics".

We will choose g_1 to be large when its argument is small, penalizing for points far away from their class representative, and g_2 to be large when its argument is large, penalizing points too close to other classes' representatives. If $\{\mathcal{K}, \mathcal{F}\}$ has small energy, each F_{ij} well represents points in K_{ij}, but poorly represents points in other classes. More specifically, for the rest of this paper, we will set

$$g_1(z) = \alpha_1 \left[(\mu_1 - z)_+\right]^2,$$

and

$$g_2(z) = \alpha_2 \left[(z - \mu_2)_+\right]^2,$$

where

$$a_+ := \begin{cases} a & a > 0 \\ 0 & \text{otherwise,} \end{cases}$$

and μ (the margins) and (α_1, α_2) are parameters. Although the exact shape of g_i does not seem to be crucial, some form of margin seems to be important, as does the g_i tapering to 0 at the margins.

If the F were orthonormalized to actually be subspaces, the use of planes and the margin in the energy 5 recalls SVM's. However, in this framework, the planes in question are not separating hyperplanes between classes; rather, they are exemplars of a class, chosen so that the distance from a class to its set of planes is as small as possible, while keeping those planes far away from the other classes. For planes (in general position) of dimension greater than one in ambient dimension larger than three, the decision boundaries are not linear, or even piecewise linear. On the other hand, in the zero dimensional "affine" case, where the F_{ij} are just points, and the algorithm is a discriminative version of k-means, the decision boundaries are piecewise linear. In this special case, the $\mathcal{F}$ can be considered a convenient device for parameterizing the decision boundary of an SVM-like classifier where the margin is specified in advance, rather than optimized.

From the viewpoint presented in Section 1.1, the discriminative k-metrics framework can be though of as a method of building adapted dictionaries, where the analysis coefficients of a signal in a given dictionary are large if the signal is associated with that dictionary, and are forced to be small otherwise. Interpreted as in 1.2, the frameworks is that of locally linear (soft) dimension reduction (to $\mathbb{R}^q$). Rather than trying to minimize distortion, at each of the k clusters assigned to a class C_i, we wish for the points in that cluster be kept far from the origin, and points not in C_i to be kept close to the origin. The clusters are learned simultaneously with the mappings, and the dimension reduction is "soft" because we use metrics instead of flats.

Finally, we emphasize to the reader the difference between the approach introduced here and the metric learning approach mentioned earlier: we are not searching for A_{ij} so that points in different classes are far from each other and points in the same class are close, but rather we want A_{ij} so that the norm of points in C_i belonging to K_{ij} is large with respect to A_{ij}, and the norm with respect to A_{ij} of points in C_r, $r \neq i$, is small. In particular, our A_{ij} will not keep points in the same class or even points in K_{ij} close together. Instead, as in the simple supervised k q-flats method, we take an abstraction of a number of points and collect them into a larger object that test points can be measured against; before it was an affine set, now it is a Mahalanobis metric.

3. Computing the discriminative k q-metrics

To minimize the functional 5, we use a stochastic gradient descent. The gradient of the energy with respect to a metric F_{ij} is given by:

$$\frac{\partial}{\partial F_{ij}} \left[\sum_{x \in K_{ij}} g_1(||F_{ij}x||^2) + \sum_{x \notin C_i} g_2(||F_{ij}x||^2) \right]$$

$$= \sum_{x \in K_{ij}} 2g_1'(||F_{ij}x||^2) F_{ij} x x^T + \sum_{x \notin C_i} 2g_2'(||F_i x||^2) F_{ij} x x^T.$$

In all the experiments below, we fix a time step $dt = .1$, $\alpha_1 = 2$, and, $\alpha_2 = 1$. We sequentially pass through each x in the dataset, updating all the F_{ij} as follows:

- Calculate the norm of x in each F_{ij}.
- Place $x \in K_{i_x j_x}$, where j_x maximizes $||F_{i_x \cdot}||^2$.
- If $||F_{i_x j_x} x||^2 < \mu_1$, $F_{i_x j_x} \longmapsto F_{i_x j_x} + dt \cdot \left(\mu_1 - ||F_{i_x j_x} x||^2\right) \cdot F_{i_x j_x} x x^T$.
- If for some $i \neq i_x$, and some j, $||F_{ij}x||^2 > \mu_2$, $F_{ij} \mapsto F_{i,j} + dt \cdot \left(\mu_2 - ||F_{ij}x||^2\right) \cdot F_{ij} x x^T$.
- Get the next x from the random order, and repeat.

We always initialize using the (representational only) k-subspaces algorithm, which is itself initialized taking neighborhoods of random points.

For each x, the computational complexity of the above algorithm is at worst $O(dkqm)$ operations, where the

data lies in d-dimensional space. The cost comes from the time it takes to compute the square norms of x in each of the metrics, which requires multiplying a d vector by a $q \times d$ matrix, and then squaring and summing, and the time it takes to update each metric, which requires multiplying x by the $q \times 1$ result of the matrix-vector multiplication above, scaling, and then adding to each of the affected metrics. This needs to be done for km metrics. If there are n points, and we make p passes through the points, the total cost (not including the initialization) would be $O(npdkqm)$ operations. In all of the experiments below, we obtained good results before 20 passes through the data. The memory cost here is only the metrics themselves; the algorithm as described is "online" in the sense that the data does not need to be stored in memory.[2]

Note that although the optimal $\mathcal{F}$ for a given problem will each represent the points in a given class well, and the points in the other classes not so well, the set of F for some class are independent of the F for a different class, so that we may work on the set of F for each class separately, leading to a trivially parallelized version of the algorithm. It could further be parallelized by sending the computations dependent on different x to different cores, but here the shared memory management would have to be more sophisticated, because each computation on each core would need to access the metrics as they have been updated by other points.

4. Experimental results

We first test the proposed discriminative k q-metrics algorithm on three standard machine learning datasets:

- The MNIST digits, consisting of 70000 28×28 images of handwritten digits divided into 60000 training examples and 10000 test examples. The data is preprocessed by projection onto the first 50 principal components.
- The 20-newsgroups dataset, consisting of 18477 documents from one of 20 newsgroups represented by its binary term document matrix, with a vocabulary of the 5000 most common words, as in (Larochelle & Bengio, 2008). The data is divided into the standard 7505/11269 test/train split.
- The ISOLET datset, consiting of 200 speakers saying each letter of the alphabet twice. 617 audio features have been extracted from each sample. The data is divided into a standard training set of the first 150 speakers, and a test set of the last 50.

All three datasets are projected onto the unit sphere.

From each data set, we extract the last 20% of the training points as a validation set for model selection. More specifically, for each data set, we choose k from $\{1, 2, 4, 8, 16\}$, q from $\{20, 40, 80, 160\}$, and (μ_1, μ_2) from $\{(1.05, .95), (1.1, .9), (1.2, .8), (1.4, .6)\}$, by training a classifier for 40 passes through the first 80% of the training data with each of the combinations of parameters, and then choose parameters by the performance of the classifier on the held out 20% of the training data. We fix $\alpha_1 = 2$, $\alpha_2 = 1$, and set the descent timestep equal to .1 for all the data sets.

Figures 1, 2, and 3 display the results of running the algorithm on the various datasets with the parameters chosen as above. In each of the figures, the blue curve is the average error on the test set over 25 runs at a given number of passes through the training set, with each mark on the x axis corresponding to one pass. We noticed that it is always better to take a vote amongst sets of F generated by restarting the algorithm. The red curve is the average majority vote of five restarts. Note that while the majority vote is always better, for MNIST it is much better, while for ISOLET and 20 newsgroups it is only marginally better. This is because for the latter two data sets k was 1. Because we initialize the F's for a given class with the k-subspace algorithm, if k is 1, then the initial F for each class is simply the principal vectors of the points in that class. In particular, each restart begins with the same initialization, and so they are not so independent.

In the case of Isolet, the classification error of the algorithm approaches but does not surpass that of SVM's; on the other two data sets, the results are better than SVM. See Figure 4 for a recap of the results.

[2] In fact, the descent we use in the experiments is "semi stochastic" in order to speed up Matlab computations. Before each pass through the data, all the distances are computed, and then the points found to violate the margins at the beginning of the pass are used in the normal way. This allows most of the distance computations to be vectorized, and does not hurt classification accuracy. On the other hand, we must store in memory all of the distances to all of the metrics, and at least after each pass, the data points.

[4] We use a vocabulary of 5000 words to better compare with (Larochelle & Bengio, 2008), which could not use the sparsity of the data. If we run our algorithm on the dataset with a 30000 word vocabulary (but still the same binary term document data unnormalized except for projection onto the sphere), we achieve an error of .216, and the training time only increases to 900 seconds for 40 passes through the data.

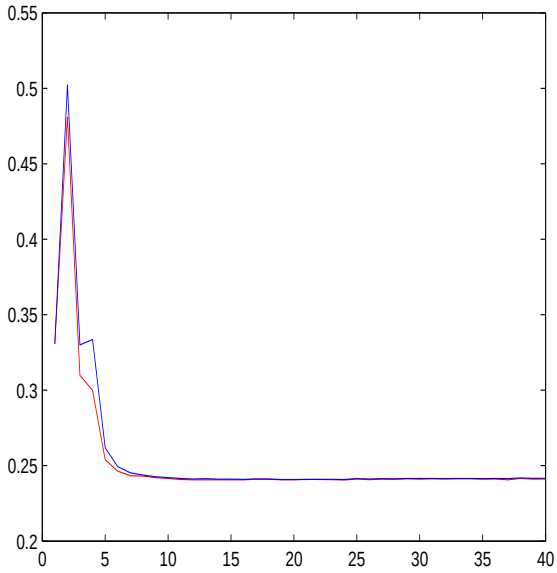

Figure 1. Misclassification on the 20 newsgroups dataset, split into 7505/11269 test/train. Each mark on the x axis corresponds to one pass through the training set, while the y axis represents the classification error. The blue curve represents the proposed algorithm, averaged over 25 runs, and the red is the average majority of five classifiers decision. Here $k = 1$, $q = 160$, $\mu_1 = 1.4$, and $\mu_2 = .6$. SVM misclassification (taken from (Larochelle & Bengio, 2008)) is .3.; their best result is .238. [4] *The entire x axis takes* 820 *seconds for the blue curve, and* $5 \cdot 820$ *seconds for the red curve.*

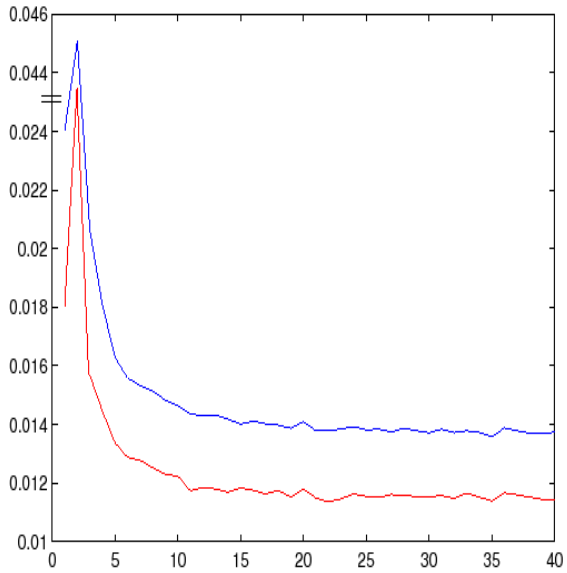

Figure 2. Misclassification on the MNIST dataset, with the standard 10000/60000 test/train split. Each mark on the x axis corresponds to one pass through the training set, while the y axis represents the classification error. The blue curve represents the proposed algorithm, averaged over 25 runs, and the red is the average majority of five classifiers decision. Here $k = 8$, $q = 20$, $\mu_1 = 1.05$, and $\mu_2 = .95$. For comparison, SVM misclassification rate is .014 without image dependent regularization, (see `http://yann.lecun.com/exdb/mnist/`*) The entire x axis takes just over* 300 *seconds for the blue curve, and* $5 \cdot 300$ *seconds for the red curve.*

4.1. Graz bikes

We also tested our approach on the Graz Bikes data set, located at `http://www.emt.tugraz.at/~pinz/data/GRAZ_02/`. The Graz data set consists of 640x480 or 480x640 pictures of bikes, cars, and persons in various enviornments. The data set has 300 images of each of the three classes, and 300 additional "background" images with none of the other objects. The bikes vs. none pixelwise classification problem has appeared in several articles, for example (Pantofaru et al., 2006; Tuytelaars & Schmid, 2007). Given a pixel from one of the bikes and background images, the problem is to determine whether or not it is in a bike using only local information. Each bike image has been segmented into bike and background, but following the above references, we use the ground truth segmentations only for testing. We subsample the images by 4, and to build the training set, we randomly extract 150000 9×9 patches from the odd images labeled bike, and 150000 patches from the odd images labeled background (again, we do not use the pixelwise ground truth segmentations, so the majority of the "bike" patches are not actually from bikes). All the patches are projected onto the unit sphere in 243 (rgb times 81) dimensional space.

We train an 80 dimensional discriminative dictionary i.e. $k = 1$ and $q = 80$) on each of the sets of patches with $\mu_1 = 1.05$ and $\mu_1 = .95$. To test an image, we calculate the square norms of all of the patches in the image in the two dictionaries, giving us two scalar functions, which are then smoothed with a box filter. In Figure 5 we show the precision-recall curve obtained, where the errors are measured using the ground truth pixelwise segmentation over all the patches 9 pixels or further from the image boundary in all the even numbered images. Our results are better than either of (Pantofaru et al., 2006; Tuytelaars & Schmid, 2007), except at the low recall high precision section of the curve, even though they used image specific tools and

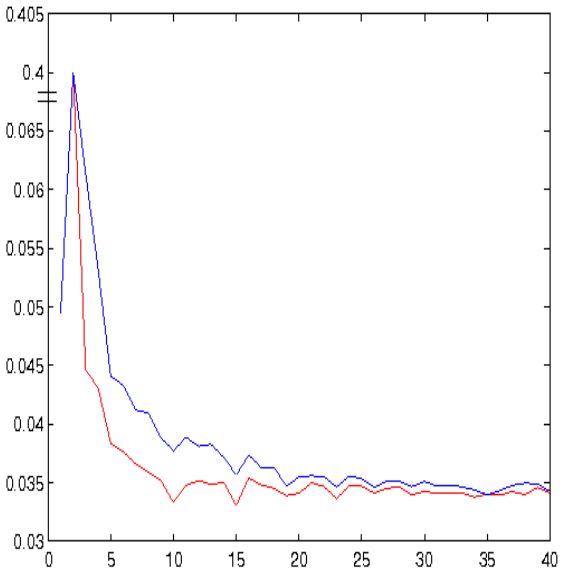

Figure 3. Misclassification on the ISOLET dataset, with the standard 25%/75% test/train split. Each mark on the x axis corresponds to one pass through the training set, while the y axis represents the classification error. The blue curve represents the proposed algorithm, averaged over 25 runs, and the red is the average majority of five classifiers decision. Here $k = 1$, $q = 20$, $\mu_1 = 1.1$, and $\mu_2 = .9$. SVM misclassification (taken from (Weinberger et al., 2006)) is .033. The entire x axis takes 23 seconds for the blue curve, and $5 \cdot 23$ seconds for the red curve.

feature extractors (SIFT, etc).

5. Conclusions and future work

This article presented a discriminative version of the k-q flats algorithm for supervised classification problems. This method gives state of the art error rates on some standard benchmarks, and is fast enough to be reasonably applied to datasets with hundreds of thousands of points in hundreds of dimensions on a desktop computer.

However, there is still much to be done. For example, the algorithm presented for minimizing energy 5 is relatively primitive, and could be greatly sped-up with some care. Following the analogy with SVM's, the margin α in the energy should be optimized for the data rather than taken as a parameter. Kernelization could possibly be useful. Finally, we are currently developing a semi-supervised version of the proposed framework. Due to the computational efficiency of the algorithm, this will open the door to the use of very large available datasets, such as image collections from flickr.com.

	MNIST	20 newsgroups	Isolet
k q-metrics (m)	1.15	24.0	3.4
k q-metrics	1.36	24.0	3.5
k q-flats	1.6	33.1	4.3
SVM	1.4	30	3.3

Figure 4. Summary of results on benchmark datasets. The row marked k q-metrics shows the average errors over 25 runs and over over passes 20 through 40. The row marked k q-metrics (m) is the same, but with the majority of 5 classifier. In each case, the parameters are determined by the automatic model selection process as above. The supervised k q-flats error is with the best manual choice of k and q (no model selection). The SVM results for ISOLET are taken from (Weinberger et al., 2006), and for the 20 newsgroups and MNIST from (Larochelle & Bengio, 2008). All errors in percent.

6. Acknowledgements

AS thanks the NSF for generous support by DMS-0811203. The work of GS was partially supported by NSF, ONR, NGA, ARO, and DARPA. We thank the anonymous reviewers for their useful comments.

References

Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, *54*, 4311–4322.

Cappelli, R., Maio, D., & Maltoni, D. (2001). Multispace kl for pattern representation and classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, *23*, 977–996.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.

Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. *ICML* (pp. 209–216).

Kambhatla, N., & Leen, T. K. (1993). Fast non-linear dimension reduction. *NIPS* (pp. 152–159).

Larochelle, H., & Bengio, Y. (2008). Classification using discriminative restricted boltzmann machines. *ICML '08: Proceedings of the 25th international conference on Machine learning* (pp. 536–543). Helsinki, Finland.

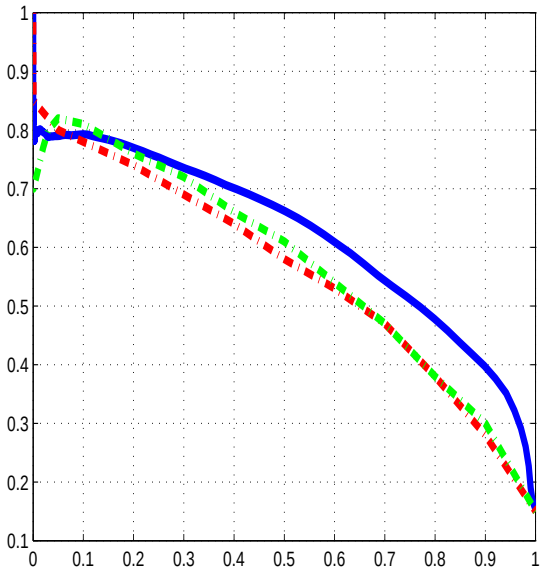

Figure 5. Precision-recall curve for the Graz bikes vs background pixelwise classification. Blue is the proposed method, red dashed is the method in (Tuytelaars & Schmid, 2007), and green dashed is the method in (Pantofaru et al., 2006).

Mairal, J., Bach, F., Ponce, J., Sapiro, G., & Zisserman, A. (2008). Supervised dictionary learning. *Advances in Neural Information Processing Systems*. Vancouver, Canada.

Mangasarian, P. S. B. . O. L. (1998). *k-plane clustering* (Technical Report MP-TR-1998-08).

Pantofaru, C., Schmid, C., & Hebert, M. (2006). Combining regions and patches for object class localization. *The Beyond Patches Workshop in conjunction with the IEEE conference on Computer Vision and Pattern Recognition* (pp. 23 – 30).

Tropp, J. (2004). *Topics in sparse approximation*. Doctoral dissertation, Computational and Applied Mathematics, The University of Texas at Austin.

Tseng, P. (1999). *Nearest q-flat to m points* (Technical Report). Seattle, WA.

Tuytelaars, T., & Schmid, C. (2007). Vector quantizing feature space with a regular lattice. *Proceedings of IEEE International Conference on Computer Vision* (pp. 1–8).

Wakin, M. (2006). *The geometry of low-dimesnional signal models*. Doctoral dissertation, Rice university.

Weinberger, K., Blitzer, J., & Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf and J. Platt (Eds.), *Advances in neural information processing systems 18*, 1473–1480. Cambridge, MA: MIT Press.

Xing, E., Ng, A., Jordan, M., & Russell, S. (2003). Distance metric learning, with application to clustering with side-information.