Supervised Learning via Discriminative k q-Flats

Arthur Szlam and Guillermo Sapiro

***Abstract*—The k q-flats algorithm is a generalization of the popular k-means algorithm, where k different q-dimensional affine spaces are considered, instead of points, as prototypes. A new modification of the k-q-flats algorithm for pattern classification is introduced in this work. The basic idea is to replace the original reconstruction only energy, which is optimized to obtain the k affine spaces, by a new one that incorporates discriminative terms. 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.**

***Index Terms*—Multi space Karhunen Loeve, local factor analysis, k-q-flats, k-q-planes, supervised learning.**

The k q-flats algorithm, [KL93], [Man98], [Tse99], [CMM01], is a generalization of the k-means algorithm where we take q-dimensional affine spaces ("flats") instead of points as prototypes. Thus, given a set of n points $X \in \mathbb{R}^d$, we wish to find k q-dimensional flats $\{F_1, \dots, F_k\}$ and a partition of X into $\{K_1, \dots, K_k\}$, 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 plane F_j. The minimization can be done using Lloyd's algorithm (EM), or the online method, both of which are guaranteed to converge to a local minimum.

We can consider the k-q-flats algorithm for supervised learning by training a set planes for each class. Given the set $X \subset \mathbb{R}^d$ consisting of n points with labels $i_1, ...i_m$ (m classes of objects), the supervised k-q flat algorithm associates planes $F_{i,j}$ to each class, minimizing the energy

$$\sum_{x} ||x - P_{F_x} x||^2, \tag{.2}$$

where $F_x = F_{i_x, j_x}$ is the nearest flat to x which has been associated to the points with label i_x, and where $P_F x$ is projection of a point x onto the corresponding q-flat F. Given a new point to classify, we assign it to the class associated to its nearest flat.

On many data sets, this simple algorithm gives excellent classification results, 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. The k q-flats algorithm is representational (reconstructive), it does not explicitly encodes the differences between the classes. In [Szl08] it was suggested to change the reconstructive energy functional (.1) to punish configurations of the flats passing through one class that get too close to points in another class. A method was proposed that improved the accuracy of k-q flats but at the cost of speed, the method was too slow to be reasonably tested on large benchmarks like the MNIST digits. In this short note, which is an continuation of [Szl08], this deficiency is rectified with a much more efficient algorithm for designing discriminative k-q flats.

Arthur Szlam is with the Department of Mathematics, UCLA; Guillermo Sapiro is with the Department of Electrical and Computer Engineering, University of Minnesota.

I. A discriminative k-q-flats framework

The energy in Equation .2 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 classification errors. In this note we consider the following energy:

$$\sum_{x} \left[\sum_{i \neq i_x, j} \left(||x - P_{F_x} x||^2 - ||x - P_{F_{i,j}}(x)||^2 + \mu \right)_+ \right], \tag{I.1}$$

where

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

and μ (the margin) is a parameter. More generally one might use an energy of the form

$$\sum_{x} \Bigg[\alpha ||x - P_{F_x} x||^2 + \beta \sum_{i \neq i_x, j} g\left(||x - P_{F_x} x||^2 - ||x - P_{F_{i,j}}(x)||^2 \right) \Bigg], \tag{I.2}$$

where g is some (increasing) function. This is an analog for k-q flats of the energy in [MBP+08] for the discriminative extension of sparse dictionary learning via K-SVD; in that case, g is a smoothed step function. Note that if $\alpha = 1$ and $\beta = 0$, this is the standard k-q flats energy.

The use of planes and the margin in the energy I.1 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 case, where the $F_{i,j}$ are just points, and the algorithm is a discriminative version of k-means, the decision boundaries are piecewise linear. In this special case, the 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.

II. Computing the discriminative k-q flats

To minimize the functional I.1, we use a stochastic gradient projection. In all the experiments presented below, the version of the algorithm where all planes pass through the origin is used. In this case, the square distance of a point x to a plane F is given by

$$||x - P_F x||^2 = ||x - F^T F x||^2$$

$$= ||x||^2 - ||Fx||^2,$$

where here and for the rest of this note, F refers to both the q-plane and a set of q vectors spanning the q-plane written as rows.

For each point x and a set of $F_{i,j}$, set

$$I(x) := \{i, j \text{ s.t. } i \neq i_x \text{ and}$$

$$||x - P_{F_x} x||^2 - ||x - P_{F_{i,j}}(x)||^2 + \mu > 0\}.$$

Then the gradient of the energy I.1 with respect to a given plane is given by

$$\begin{aligned}
&\frac{\partial}{\partial F_{i,j}} \sum_x \left[\sum_{i \neq i_x, j} \left(||x - P_{F_x} x||^2 - ||x - P_{F_{i,j}}(x)||^2 + \mu \right)_+ \right] \\
&= - \sum_{\substack{x \text{ s.t.} \\ F_x = F_{i,j}}} |I(x)| \frac{\partial}{\partial F_{i,j}} ||F_{i,j} x||^2 + \sum_{\substack{x \text{ s.t.} \\ i,j \in I(x)}} \frac{\partial}{\partial F_{i,j}} ||F_{i,j} x||^2, \\
&= -2 \sum_{\substack{x \text{ s.t.} \\ F_x = F_{i,j}}} |I(x)| F_{i,j} x x^T + 2 \sum_{\substack{x \text{ s.t.} \\ i,j \in I(x)}} F_{i,j} x x^T.
\end{aligned}$$

This suggests the following stochastic gradient descent with projections:

1. Choose parameters μ, dt, k, and q.
2. Initialize k-q planes for each class (with the original k-q flats algorithm, without discriminative term, for example).
3. Pick an x at random.
4. Update the $F_{i,j}$ by
   - $F_x \mapsto \mathcal{O}(F_x + dt \cdot |I| \cdot F_x x x^T)$,
   - $F_{i,j} \mapsto \mathcal{O}(F_x + dt \cdot F_{i,j} x x^T)$, $\{i, j\} \in I$,

   where $\mathcal{O}(A)$ is an orthonormal basis for the columns of A.
5. Repeat from 2.

Here "projections" refers to the re-orthonormalizations of each F after its modification by a point x. In fact, as we will see below, this step is sometimes unnecessary. Skipping it changes the functional, but on some data sets the change not only speeds-up the algorithm, but sometimes even improves the classification accuracy.

Note that the algorithm could be easily parallelized by sending the computations dependent on different x to different cores.

III. Experimental results

We test the discriminative k-q flats 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 300 principal components.
- The 20-newsgroups dataset, consisting of 18477 documents from one of 20 newsgroups represented by its *tf-idf* normalized term document matrix. This data is projected onto 500 principal components and then randomly divided into 16000 training examples and 2774 test examples.
- 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, and preprocessed by projection onto its first 300 principal components.

All three datasets are then projected onto the unit sphere.

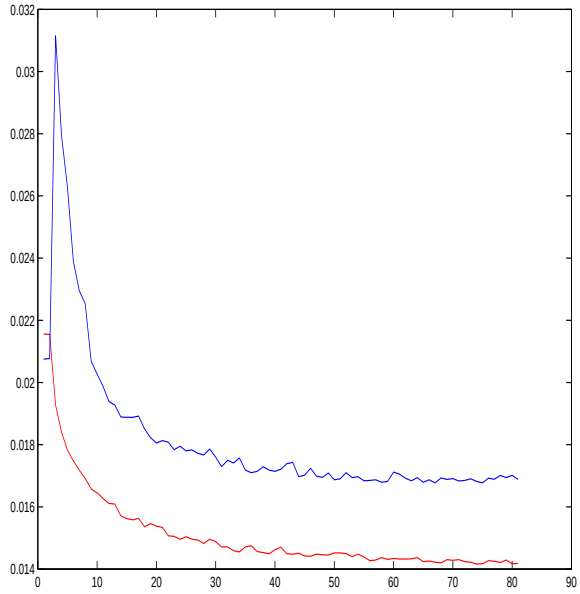

Fig. 1. *Misclassification on the MNIST dataset, averaged over ten runs with the standard 10000/60000 test/train split. Each mark on the x axis corresponds to 10,000 iterations, while the y axis represents the classification error. The blue curve represents proposed the algorithm with orthonormalized F, and the red represents the algorithm with non-orthonormalized F. Here $k = 20$, $q = 40$, and $\mu = .2$. For comparisson, SVM misclassification rate is .014 without image dependent regularization, and .011 with deskewing (see* `http://yann.lecun.com/exdb/mnist/`*). This is comparable with our results.*

Figures 1, 2, and 3 display the results of running the algorithm on the various datasets (with and without orthogonalizations of the F's). In each case, the classification error of the orthogonalized algorithm approaches but does not surpass that of SVM's. In the ISOLET and 20 newsgroups, not orthogonalizing does not hurt classifica-

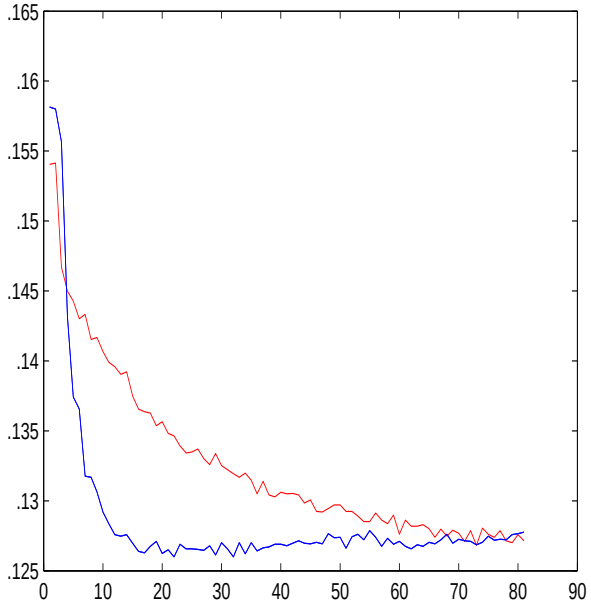

Fig. 2. *Misclassification on the 20 newsgroups dataset, averaged over ten runs with random splits into 2000/16774 test/train. The blue curve represents the proposed algorithm with orthonormalized F, and the red represents the algorithm with non-orthonormalized F. The x axis is iterations (each mark is 10,000), and the y axis is classification error. Here $k = 2$, $q = 80$, and $\mu = .4$. SVM misclassification (taken from [WBS06]) is .124.*

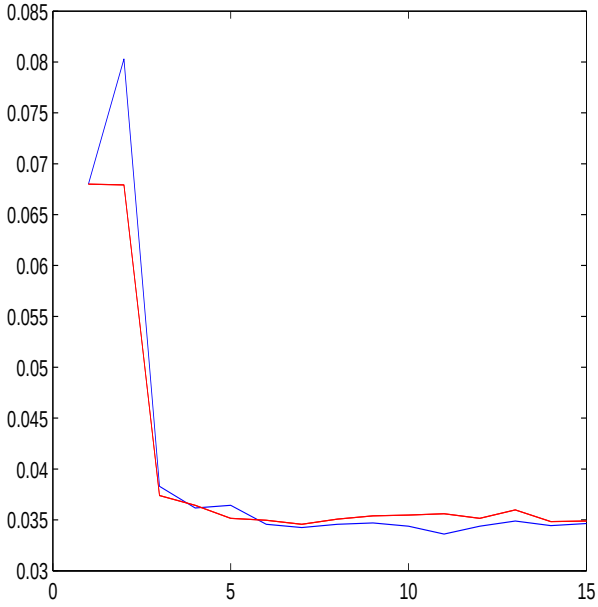

Fig. 3. *Misclassification on the ISOLET dataset, averaged over ten runs with the standard 25%/75% test/train split. The blue curve represents the proposed algorithm with orthonormalized F, and the red represents the algorithm with non-orthonormalized F. The x axis is iterations (each mark is 10,000), and the y axis is classification error. Here $k = 1$, $q = 40$, and $\mu = .2$. SVM misclassification (taken from [WBS06]) is .033.*

tion error, and in fact improves it on the 20 newsgroups.

Figure 4 shows the timings for the various datasets. All the code has been written in Matlab; there should be substantial gains simply from writing in a compiled language. On the other hand, note that to get good results using the non-orthonormalized version of the algorithm on the newsgroups takes less than 15 minutes of training time, and less than a minute for ISOLET.

	orthogonalized	non-orthogonalized
MNIST	.01	.005
20 newsgroups	.035	.004
isolet	.005	.001

Fig. 4. *Timings per iteration in seconds. The timings were run on an Intel core duo @2.4 gigaherz with 2 gigabytes of ram.*

IV. Conclusions and future work

This note presents a discriminative version of the k-q flats algorithm for supervised classification problems. This method gives near 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 I.1 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.

V. Acknowledgements

AS thanks the NSF for generous support by DMS-0811203. The work of GS was partially supported by NSF, ONR, NGA, DARPA, and ARO.

References

[CMM01] Raffaele Cappelli, Dario Maio, and Davide Maltoni. Multispace kl for pattern representation and classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(9):977–996, 2001.

[KL93] Nanda Kambhatla and Todd K. Leen. Fast non-linear dimension reduction. In *NIPS*, pages 152–159, 1993.

[Man98] P. S. Bradley & O. L. Mangasarian. k-plane clustering. Technical Report MP-TR-1998-08, 1998.

[MBP+08] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. *Computer Vision and Pattern Recognition, 2008. CVPR '08. IEEE Conference on*, 2008.

[Szl08] A. Szlam. Modifications of k q-flats for supervised learning. Technical Report CAM Rep. 07-46, Computational and Applied mathematics, UCLA, April 2008.

[Tse99] P. Tseng. Nearest q-flat to m points. Technical report, Seattle, WA, 1999.

[WBS06] Kilian Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf, and J. Platt,

editors, *Advances in Neural Information Processing Systems 18*, pages 1473–1480. MIT Press, Cambridge, MA, 2006.