Discriminative Features via Generalized Eigenvectors

Nikos Karampatziakis NIKOSK@MICROSOFT.COM
Paul Mineiro PMINEIRO@MICROSOFT.COM
Microsoft CISL, 1 Microsoft Way, Redmond, WA 98052 USA

Abstract

Representing examples in a way that is compatible with the underlying classifier can greatly enhance the performance of a learning system. In this paper we investigate scalable techniques for inducing discriminative features by taking advantage of simple second order structure in the data. We focus on multiclass classification and show that features extracted from the generalized eigenvectors of the class conditional second moments lead to classifiers with excellent empirical performance. Moreover, these features have attractive theoretical properties, such as inducing representations that are invariant to linear transformations of the input. We evaluate classifiers built from these features on three different tasks, obtaining state of the art results.

1. Introduction

Supervised learning has been a great success story for machine learning, both in theory and in practice. In theory, we have a good understanding of the conditions under which supervised learning can succeed (Vapnik, 1998). In practice, supervised learning approaches are profitably employed in many domains, from movie recommendation to speech and image recognition (Koren et al., 2009; Hinton et al., 2012a; Krizhevsky et al., 2012). The success of all of these systems crucially hinges on the compatibility between the model and the representation used to solve the problem.

For some problems, the kinds of representations and models that lead to good performance are well-known. In text classification, for example, unigram and bigram features together with linear classifiers are known to work well for a variety of related tasks (Halevy et al., 2009). For other problems, such as drug design, speech, and image recognition, far less is known about which combinations are effective. This has fueled interest in methods that can learn the appropriate representations directly from the raw signal, with techniques such as dictionary learning (Mairal et al., 2008) and deep learning (Krizhevsky et al., 2012; Hinton et al., 2012a) achieving state of the art performance in many important problems.

In this work, we explore conceptually and computationally simple ways to create discriminative features that can scale to a large number of examples, even when data is distributed across many machines. Our techniques are not a panacea. They are exploiting simple second order structure in the data and it is very easy to come up with sufficient conditions under which they will not give any advantage over learning using the raw signal. Nevertheless, they empirically work remarkably well.

Our setup is the usual multiclass setting where we are given labeled data $\{x_i, y_i\}_{i=1}^n$, sampled iid from a distribution $\mathcal{D}$ on $\mathbb{R}^d \times [k]$, and we need to come up with a classifier $h : \mathbb{R}^d \to [k]$ with low generalization error $\mathbb{P}_{\mathcal{D}}(h(x) \neq y)$. Abusing notation, we will sometimes use y to refer to the one hot encoding of y that identifies each class with one of the vertices of the standard $k-1$-simplex. To keep the focus on the quality of our feature representation we will restrict ourselves to h being linear, such as a multiclass linear SVM or multinomial logistic regression. We suspect representations that improve the performance of linear classifiers will also beneficially compose with nonlinear techniques.

2. Method

One of the simplest possible statistics involving both features and labels is the matrix $\mathbb{E}[xy^\top]$, which in multiclass classification is the collection of class-conditional mean feature vectors. This statistic has been thoroughly explored, e.g., Fisher LDA (Fisher, 1936) and Sliced Inverse Regression (Li, 1991). However, in many practical applications we expect that the data distribution contains much more information than that contained in the first moment statistics. The natural next object of study is the tensor $\mathbb{E}[x \otimes x \otimes y]$.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

In multiclass classification, the tensor $\mathbb{E}[x \otimes x \otimes y]$ is simply a collection of the conditional second moment matrices $C_i = \mathbb{E}[xx^\top | y = i]$. There are many standard ways of extracting features from these matrices. For example, one could try per-class PCA (Wold & Sjostrom, 1977) which will find directions that maximize $v^\top \mathbb{E}[xx^\top | y = i]v = \mathbb{E}[(v^\top x)^2 | y = i]$, or VCA (Livni et al., 2013) which will find directions that minimize the same quantity. The subtlety here is that there is no reason to believe that these directions are specific to class i. In other words, the directions we find might be very similar for all classes and, therefore, not be discriminative.

A simple alternative is to work with the quotient

$$R_{ij}(v) = \frac{\mathbb{E}[(v^\top x)^2 | y = i]}{\mathbb{E}[(v^\top x)^2 | y = j]} = \frac{v^\top C_i v}{v^\top C_j v}, \tag{1}$$

whose local maximizers are the generalized eigenvectors solving $C_i v = \lambda C_j v$.[1] Efficient and robust routines for solving these types of problems are part of mature software packages such as LAPACK.

Since objective (1) is homogeneous in v, we will assume that each eigenvector v is scaled such that $v^\top C_j v = 1$. Then we have that $v^\top C_i v = \lambda$, i.e. on average, the squared projection of an example from class i on v will be λ while the squared projection of an example from class j will be 1. As long as λ is far from 1, this gives us a direction along which we expect to be able to discriminate the two classes by simply using the magnitude of the projection. Moreover, if there are many eigenvalues substantially different from 1 all associated eigenvectors can be used as feature detectors.

2.1. Useful Properties

The feature detectors resulting from maximizing equation (1) have two useful properties which we list below. For simplicity we state the results assuming full rank exact conditional moment matrices, and then discuss the impact of regularization and finite samples.

Proposition 1. *(Invariance) Under the above assumptions, the embedding $v^\top x$ is invariant to invertible linear transformations of x.*

Proof. Let $A \in \mathbb{R}^{d \times d}$ be invertible and $x' = Ax$ be the transformed input. Let $C_m = \mathbb{E}[xx^\top | y = m]$ be the second moment matrix given $y = m$ for the original data with Cholesky factorization $C_m = L_m L_m^\top$. For the transformed data, the conditional second moments are $\mathbb{E}[x'x'^\top | y = m] = A\mathbb{E}[xx^\top | y = m]A^\top = AC_m A^\top$ and the corresponding generalized eigenvector v' satisfies $AC_i A^\top v' = \lambda AC_j A^\top v'$. Letting $v' = A^{-\top} L_j^{-\top} u'$ we see that u' also satisfies $L_j^{-1} C_i L_j^{-\top} u' = \lambda u'$. Finally, the embedding involves only $v'^\top x' = u'^\top L_j^{-1} A^{-1} Ax = u^\top L_j^{-1} x$ which is the same as the embedding for the original data. □

It is worth pointing out that the results of some popular methods, such as PCA, are not invariant to linear transformations of the inputs. For such methods, differences in preprocessing and normalization can lead to vastly different results. The practical utility of an "off the shelf" classifier is greatly improved by this invariance, which provides robustness to data specification, e.g., differing units of measurement across the original features.

Proposition 2. *(Diversity) Two feature detectors v_1 and v_2 extracted from the same ordered class pair (i, j) have uncorrelated responses $\mathbb{E}[(v_1^\top x)(v_2^\top x) | y = j] = 0$.*

Proof. This follows from the orthogonality of the eigenvectors in the induced problem $L_j^{-1} C_i L_j^{-\top} u = \lambda u$ (c.f. proof of Proposition 1) and the connection $v = L_j^{-\top} u$. If u_1 and u_2 are eigenvectors of $L_j^{-1} C_i L_j^{-\top}$ then $0 = u_1^\top u_2 = v_1^\top L_j L_j^\top v_2 = v_1^\top \mathbb{E}[xx^\top | y = j]v_2 = \mathbb{E}[(v_1^\top x)(v_2^\top x) | y = j]$. □

Diversity indicates the different generalized eigenvectors per class pair provide complementary information, and that techniques which only use the first generalized eigenvector are not maximally exploiting the data.

2.2. Finite Sample Considerations

Even though we have shown the properties of our method assuming knowledge of the expectations $\mathbb{E}[xx^\top | y = m]$, in practice we estimate these quantities from our training samples. The empirical average

$$\hat{C}_m = \frac{\sum_{i=1}^n \mathbb{I}[y_i = m] x_i x_i^\top}{\sum_{i=1}^n \mathbb{I}[y_i = m]} \tag{2}$$

converges to the expectation at a rate of $O(n^{-1/2})$. Here and below we are suppressing the dependence upon the dimensionality d, which we consider fixed. Typical finite sample tail bounds become meaningful once $n = O(d \log d)$ (Vershynin, 2010).

Given $\hat{C}_m = C_m + E_m$ with $||E_m||_2 = O(n^{-1/2})$, we can use results from matrix perturbation theory to establish that our finite sample results cannot be too far from those obtained using the expected values. For example, if the *Crawford number*

$$c(C_i, C_j) \doteq \min_{||v||=1} (v^\top C_i v)^2 + (v^\top C_j v)^2 > 0,$$

[1] An alternative would be to use the covariance matrix instead of the second moment in the denominator. This leads to an offset term in our feature detector that sometimes leads to better empirical results. For ease of exposition we do not explore this in the remainder of this paper.

Algorithm 1 Generalized Eigenvectors for Multiclass

Require: $S = \{(x_i, y_i)\}_{i=1}^n$, $\theta \geq 0$ and $\gamma \geq 0$
$F \leftarrow \emptyset$
for $(i, j \neq i) \in \{1, \ldots, k\}^2$ **do**
  Solve $\hat{C}_i V = (\hat{C}_j + \frac{\gamma}{d} \text{Trace}(\hat{C}_j) I) V \Lambda$
  $F \leftarrow F \cup \{V_q | \Lambda_{qq} \geq \theta\}$
end for
$\psi_{v,\alpha,\delta}(x) \doteq \max(0, \delta v^\top x)^{\alpha/2}$
$\phi(x) \doteq [\psi_{v,\alpha,\delta}(x) | v, \alpha, \delta \in F \times \{1, 2, 3\} \times \{-1, 1\}]$
$w = \text{MultiLogit}(\{(\phi(x), y) | (x, y) \in S\})$

and the perturbations E_i and E_j satisfy

$$||E_i||_2^2 + ||E_j||_2^2 < c(C_i, C_j),$$

then (Golub & Van Loan, 2012) for all $q \in [d]$

$$\tan(|\tan^{-1}(\lambda_q) - \tan^{-1}(\hat{\lambda}_q)|) \leq O\left(\frac{1}{\sqrt{n} c(C_i, C_j)}\right),$$

where $\lambda_q, \hat{\lambda}_q$ are the q-th generalized eigenvalues of the matrix pairs C_i, C_j and $\hat{C}_i, \hat{C}_j$ respectively. Similar results apply to the sine of the angle between an estimated generalized eigenvector and the true one (Demmel et al., 2000) Section 5.7.

2.3. Regularization

An additional concern with finite samples is that $\hat{C}_m$ may not be full rank as we have assumed until now. In particular, if there are fewer than d examples in class m, then $\hat{C}_m$ is guaranteed to be rank deficient. When such a matrix appears in the denominator of (1), estimation of the eigenvectors can be unstable and overly sensitive to the sample at hand. A common solution (Platt et al., 2010) is to regularize the denominator matrix by adding a multiple of the identity to the denominator, i.e., maximizing

$$R_{ij}^\gamma(v) = \frac{v^\top \hat{C}_i v}{v^\top (\hat{C}_j + \gamma I) v}, \tag{3}$$

which is equivalent to maximizing equation (1) with an additional upper-bound constraint on the norm of v. We typically set γ to be a small multiple of the average eigenvalue of $\hat{C}_j$ (Friedman, 1989) which can be easily obtained as the trace of $\hat{C}_j$ divided by d. In Section 4 we find this strategy empirically effective.

2.4. An Algorithm

We are left with specifying a full algorithm for multiclass classification. First we need to specify how to use the eigenvectors $\{v_i\}$. The eigenvectors define an embedding for each example x using the projection magnitudes $\{v_i^\top x\}$ as new coordinates. However the embedding is linear, therefore composition with a linear classifier is equivalent to learning a linear classifier in the original space, perhaps with a different regularization. This motivates the use of nonlinear functions of the projection magnitude.

To construct nonlinear maps, we can get inspiration from the optimization criterion in equation (1), i.e., the ratio of expected projection magnitudes conditional on different class labels. For example, we could use a nonlinear map such as $(v^\top x)^2$. This type of nonlinearity can be sensitive (for example, it is not Lipschitz) so in practice more robust proxies can be used such as $|v^\top x|$ or even $|v^\top x|^{1/2}$.[2] In principle, smoothing splines or any other flexible set of univariate basis functions could be used. In our experiments we simply fit a piecewise cubic polynomial on $|v^\top x|^{1/2}$. The polynomial has only two pieces, one for $v^\top x > 0$ and one for $v^\top x \leq 0$. We briefly experimented with interaction terms between projection magnitudes, but did not find them beneficial.

Additionally, we need to address from which class pairs to extract eigenvectors. A simple and empirically effective approach, suitable when the number of classes is modest, is to just use all ordered pairs of classes. This can be wasteful if two classes are never confused. The alternative, however, of leaving out a pair (i, j) is that the classifier might have no way of distinguishing between these two classes. Since we do not know upfront which pairs of classes will be confused, our brute force approach is just a safe way to endow the classifier with enough flexibility to deal with any pair of classes that could potentially be confused. Of course, as the number of classes grows, this brute force approach becomes less viable both computationally (due to the quadratic increase in generalized eigenvalue problems) and statistically (due to the increase in the number of features for the final classifier). We discuss issues regarding large numbers of classes in Section 5.

Finally, the generalized eigenvalues can guide us in picking a subset of the d generalized eigenvectors we could extract from each class pair, i.e., generalized eigenvalues are useful for feature selection. A generalized eigenvector v with eigenvalue λ has $\mathbb{E}[(v^\top x)^2 | y]$ equal to 1 for the denominator class $y = j$ and equal to λ for the numerator class $y = i$. Therefore, eigenvalues far from 1 correspond to highly discriminative features. Similar to (Platt et al., 2010), we extract the top few eigenvectors, as top eigenspaces are cheaper to compute than bottom eigenspaces. To guard against picking non-discriminative eigenvectors, we discard those whose eigenvalues are less than a threshold $\theta > 1$.

[2] These choices are simple and yield only slightly worse results than what we report in our experiments.

Method	Signal	Noise
PCA	$\mathbb{E}[xx^\top]$	I
VCA	I	$\mathbb{E}[xx^\top]$
Fisher LDA	$\mathbb{E}_y[\mathbb{E}[x\|y]\mathbb{E}[x\|y]^\top]$	$\sum_y \mathrm{Cov}[x\|y]$
SIR	$\sum_y \mathbb{E}[w\|y]\mathbb{E}[w\|y]^\top$	I
Oriented PCA	$\mathbb{E}[xx^\top]$	$\mathbb{E}[zz^\top]$
Our method	$\mathbb{E}[xx^\top\|y=i]$	$\mathbb{E}[xx^\top\|y=j]$

Table 1. Table of related methods (assuming $\mathbb{E}[x]=0$) for finding directions that maximize the signal to noise ratio. $\mathrm{Cov}[x|y]$ refers to the conditional covariance matrix of x given y, w is a whitened version of x, and z is any type of noise meaningful to the task at hand.

The above observations lead to the GEM procedure outlined in Algorithm 1. Although Algorithm 1 has proven sufficiently versatile for the experiments described herein, it is merely an example of how to use generalized eigenvalue based features for multiclass classification. Other classification techniques could benefit from using the raw projection values without any nonlinear manipulation, e.g., decision trees; additionally the generalized eigenvectors could be used to initialize a neural network architecture as a form of pre-training.

We remark that each step in Algorithm 1 is highly amenable to distributed implementation: empirical class-conditional second moment matrices can be computed using map-reduce techniques, the generalized eigenvalue problems can be solved independently in parallel, and the logistic regression optimization is convex and therefore highly scalable (Agarwal et al., 2011).

3. Related Work

Our approach resembles many existing methods that work by finding eigenvectors of matrices constructed from data. One can think of all these approaches as procedures for finding directions v that maximize a signal to noise ratio, with symmetric matrices S and N chosen such that the quadratic forms $v^\top Sv$ and $v^\top Nv$ represent the signal and the noise, respectively, captured along direction v,

$$R(v) = \frac{v^\top Sv}{v^\top Nv}. \tag{4}$$

In Table 1 we present many well known approaches that could be cast in this framework. Principal Component Analysis (PCA) finds the directions of maximal variance without any particular noise model. The recently proposed Vanishing Component Analysis (VCA) (Livni et al., 2013) finds the directions on which the projections vanish so it can be thought as swapping the roles of signal and noise in PCA. Fisher LDA maximizes the variability in the class means while minimizing the within class variance. Sliced Inverse Regression first whitens x, and then uses the second moment matrix of the conditional whitened means as the signal and, like PCA, has no particular noise model. Finally, oriented PCA (Diamantaras & Kung, 1996; Platt et al., 2010) is a very general framework in which the noise matrix can be the correlation matrix of any type of noise z meaningful to the task at hand.

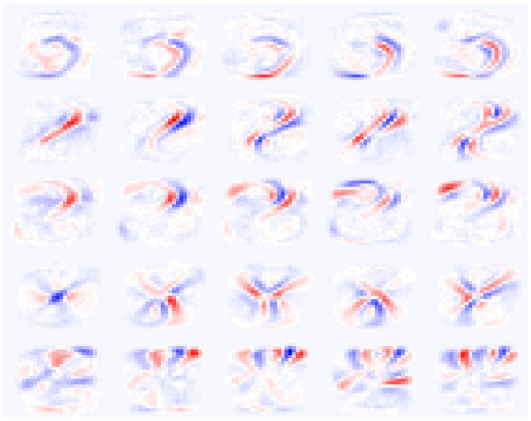

Figure 1. Pictures of the top 5 generalized eigenvectors for MNIST for class pairs $(3,2)$ (top row), $(8,5)$ (second row), $(3,5)$ (third row), $(8,0)$ (fourth row), and $(4,9)$ (bottom row) with $\gamma = 0.5$. Filters have large response on the first class and small response on the second class. Best viewed in color.

By closely examining the signal and noise matrices, it is clear that each method can be further distinguished according to two other capabilities: whether it is possible to extract many directions, and whether the directions are discriminative. For example, PCA and VCA can extract many directions but these are not discriminative. In contrast, Fisher LDA and SIR are discriminative but they work with rank-k matrices so the number of directions that could be extracted is limited by the number of classes. Furthermore both of these methods lose valuable fidelity about the data by using the conditional means.

Oriented PCA is sufficiently general to encompass our technique as a special case. Nonetheless, to the best of our knowledge, the specific signal and noise models in this paper are novel and, as we show in Section 4, they empirically work very well.

4. Experiments

4.1. MNIST

We begin with the MNIST database of handwritten digits (LeCun et al., 1998), for which we can visualize the generalized eigenvectors, providing intuition regarding the discriminative nature of the computed directions. For each of the ten classes, we estimated $C_m = \mathbb{E}[xx^\top|y=m]$ using (2) and then extracted generalized eigenvectors for each class pair (i,j) by solving $\hat{C}_i v = \lambda(\frac{\gamma}{d}\,\mathrm{Trace}(\hat{C}_j)I + \hat{C}_j)v$. Figure 1 shows a sample of results from this procedure for

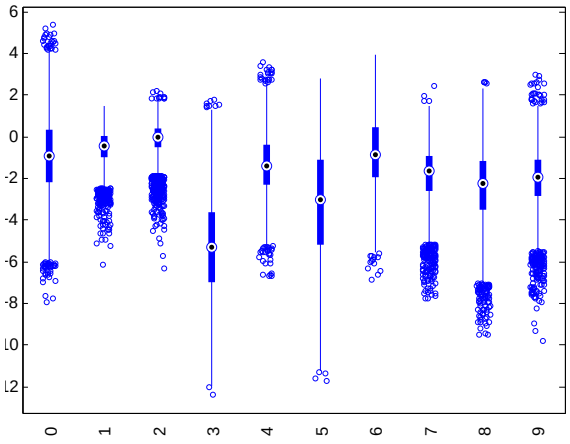

Figure 2. Boxplot of the projection onto the first generalized eigenvector for class pair $(3, 2)$ across the MNIST training set grouped by label. Squared projection magnitude on 2s is on average unity, whereas on 3s it is the eigenvalue. Large responses can appear in other classes (e.g., 5s and 8s), but this is not guaranteed by construction.

five class pairs (one in each row) and $\gamma = 0.5$. In the top row we use class pair $(3, 2)$ and we observe that the eigenvectors are sensitive to the circular stroke of a typical 3 while remaining insensitive to the areas where 2s and 3s overlap. Similar results are seen in the second and third rows where we use class pairs $(8, 5)$ and $(3, 5)$: the strokes we find are along areas used by the first class and mostly avoided by the second class. In the fourth row we use class pair $(8, 0)$. Here we observe two patterns. First, a dot in the center that avoids the 0s. The other 4 detectors consist of positive (red) and negative (blue) strokes arranged in a way that would cancel each other if we take the inner product of the detector with a radially symmetric pattern such as a 0. Similarly in the bottom row with class pair $(4, 9)$, the detector attempts to cancel the horizontal stroke corresponding to the top of the 9, where a typical 4 would be open.

Figure 2 shows for each of the ten classes the distribution of values obtained by projecting the training examples in that class onto the first eigenvector for class pair $(3, 2)$, i.e., the top left image in Figure 1. The projection pattern inspires two comments. First, while the magnitude of the projection is itself discriminative for distinguishing between 2s and 3s, there is additional information in knowing the sign of the projection. This motivates our particular choice of nonlinear expansion in Algorithm 1. Second, the detector is discriminative for class 3 vs. class 2 as per design, but also useful for distinguishing other classes from 2s. However certain classes such as 1s and 7s would be completely confused with 2s were this the only feature. The number of classes in MNIST is modest ($k = 10$) so we can easily afford to extract features for all $k(k-1)$ class pairs for excellent discrimination. For problems with a large number of classes, however, we need to carefully pick the subproblems we need to solve so that the resulting set of features is discriminative, diverse, and complete. We revisit this topic in Section 5.

Method	Test Errors
Random	283
Dropout	120
DropConnect	112
GEM	108
deep GEM	96
Maxout	94

Table 2. Test errors on MNIST. All techniques are permutation invariant and do not augment the training set.

Table 2 contains results for algorithm 1 on the MNIST test set. To determine the hyperparameter settings γ and θ, we held out a fraction of the training set for validation. Once γ and θ were determined, we trained on the entire training set. We also include baseline results with (an equal number of) randomly generated directions to help isolate the contribution of the generalized eigenvector extraction from the subsequent nonlinear basis expansion. This is denoted as "Random".

For "deep GEM" we applied GEM to the representation created by GEM, i.e., line 7 of Algorithm 1. Because of the intermediate nonlinearity this is not equivalent to a single application of GEM, and we do observe an improvement in generalization. Subsequent recursive compositions of GEM degrade generalization, e.g., 3 levels of GEM yields 110 test errors. We would like to better understand the conditions under which composing GEM with itself is beneficial.

Our results occupy an intermediate position amongst state of the art results on MNIST. For comparison we include results from other permutation-invariant methods from (Wan et al., 2013) and (Goodfellow et al., 2013). These methods rely on generic non-convex optimization techniques and face challenging scaling issues in a distributed setting (Dean et al., 2012). While maximization of the Rayleigh quotient (1) is non-convex, mature implementations are computationally efficient and numerically robust. The final classifier is built using convex techniques and our pipeline is particularly well suited to the distributed setting, as discussed in Section 5.

4.2. Covertype

Covertype is a multiclass data set whose task is to predict one of 7 forest cover types using 54 cartographic variables (Blackard & Dean, 1999). RBF kernels provide state of the art performance on Covertype, and consequently it has been a benchmark dataset for fast approximate ker-

Method	Test Error Rate
GEM	12.9%
RFF	12.7%
deep GEM	9.8%
GEM + RFF	8.4%
RBF kernel (exact)	8.8%

Table 3. Test error rates on Covertype. The RBF kernel result is from (Jose et al., 2013) where they also use a 90%-10% (but different) train-test split.

nel techniques (Rahimi & Recht, 2007; Jose et al., 2013). Here, we demonstrate that generalized eigenvector extraction composes well with randomized feature maps in the primal. This approximates generalized eigenfunction extraction in the RKHS, while retaining the speed and compactness of primal approaches.

Covertype does not come with a designated test set, so we randomly permuted the data set and used the last 10% for testing, utilizing the same train-test split for all experiments. We followed the same experimental protocol as the previous section, i.e., held out a portion of the training set for validation to select hyperparameters.

Table 3 summarizes the results.[3] GEM and deep GEM are exactly the same as in the previous section, i.e., Algorithm 1 without and with self-composition respectively. RFF stands for Random Fourier Features (Rahimi & Recht, 2007), in which the Gaussian kernel is approximated in the primal by a randomized cosine map; we used logistic regression for the primal learning algorithm. We treated the bandwidth and number of cosines as hyperparameters to be optimized.

The relatively poor classification performance of RFF on Covertype has been noted before (Rahimi & Recht, 2007), a result we reproduce here. Instead of using the randomized feature map directly, however, we can apply Algorithm 1 to the representation induced by RFF, which we denote GEM + RFF. This improves the classification error with only modest increase in computation cost, e.g., in MATLAB it takes 8 seconds to compute the randomized Fourier features, 58 seconds to (sequentially) solve the generalized eigenvalue problems and compute the GEM feature representation, and 372 seconds to optimize the logistic regression. The final error rate of 8.4% is a new record for this task.

4.3. TIMIT

TIMIT is a corpus of phonemically and lexically annotated speech of English speakers of multiple genders and dialects (Fisher et al., 1986). Although the ultimate problem is sequence annotation, there is a derived multiclass classification problem of predicting the phonemic annotation associated with a short segment of audio. Such a classifier can be composed with standard sequence modeling techniques to produce an overall solution, which has made the multiclass problem a subject of research (Hinton et al., 2012b; Hutchinson et al., 2012). In this experiment we focus exclusively on the multiclass problem.

We use a standard preprocessing of TIMIT as our initial representation (Hutchinson et al., 2012). Specifically the speech is converted into feature vectors via the first to twelfth Mel frequency cepstral coefficients and energy plus first and second temporal derivatives. This results in 39 coefficients per frame, which is concatenated with 5 preceding and 5 following frames to produce a 429 coefficient input to the classifier. The targets for the classifier are the 183 phone states (i.e., 61 phones each in 3 possible states).

We use the standard training, development, and test sets of TIMIT. As in previous experiments herein, hyperparameters are optimized on the development set (using cross-entropy as the objective), but unlike previous experiments we do not retrain with the development set once hyperparameters are determined, in correspondence with the experimental protocol used with the T-DSN (Hutchinson et al., 2012).

With 183 classes the all-pairs approach for generalized eigenvector extraction is unwieldy, so we used a randomized procedure to select from which class pairs to extract features, by randomly positioning the class labels on a hypercube and extracting generalized eigenvectors only for immediate hyperneighbors. For k classes this results in $O(k \log k)$ generalized eigenvalue problems. Although we did not attempt a thorough exploration of different strategies for subproblem selection, the hypercube heuristic yielded better results for a given feature budget than either uniform random selection over all class pairs or stratified random selection over class pairs ensuring equal numbers of denominator or numerator classes. The resulting performance for five different choices of random hypercube is shown in the row of Table 4 denoted GEM. We show both multiclass error rate as well as cross entropy, the objective we are actually optimizing.

The random subproblem selection creates an opportunity to ensemble, and empirically the resulting classifiers are sufficiently diverse that ensembling yields a substantial improvement. In Table 4, denoted GEM ensemble, we show the performance of the ensemble prediction of the 5 classifiers using the geometric mean prediction (this is the prediction that minimizes its average KL-divergence to each element of the ensemble). The result matches the classification error and improves upon the cross-entropy loss of the best published T-DSN. This is remarkable considering the T-DSN is a deep architecture employing between 8 and

[3] When comparing with other published results, be aware that many authors adjust the task to be a binary classification task.

Method	Frame State Error (%)	Cross Entropy
GEM	41.87 ± 0.073	1.637 ± 0.001
T-DSN	40.9	2.02
GEM (ensemble)	40.86	1.581

Table 4. Results on TIMIT test set. T-DSN is the best result from (Hutchinson et al., 2012).

13 stacked layers of nonlinear transformations, whereas the GEM procedure produces a shallow architecture with a single nonlinear layer.

5. Discussion

Given the simplicity and empirical success of our method, we were surprised to find considerable work on methods that only extract the first generalized eigenvector (Mika et al., 2003) but very little work on using the top m generalized eigenvectors. Our experience is that additional eigenvectors provide complementary information. Empirically, their inclusion in the final classifier far outweighs the necessary increase in sample complexity, especially given typical modern data set sizes. Thus we believe this technique should be valuable in other domains.

Of course our method will not be able to extract anything useful if all classes have the same second moment but different higher order statistics. While our limited experience here suggests second moments are informative for natural datasets, there are potential benefits in using higher order moments. For example, we could replace our class-conditional second moment matrix with a second moment matrix conditioned on other events, informed by higher order moments.

As the number of class labels increases, say $k \geq 1000$, our brute force all-pairs approach, which scales as $O(k^2)$, becomes increasingly difficult both computationally and statistically: we need to solve $O(k^2)$ eigenvector problems (possibly in parallel) and deal with $O(k^2)$ features in the ultimate classifier. Taking a step back, the object of our attention is the tensor $\mathbb{E}[x \otimes x \otimes y]$ and in this paper we only studied one way of selecting pairs of slices from it. In particular, our slices are tensor contractions with one of the standard basis vectors in $\mathbb{R}^k$. Clearly, contracting the tensor with any vector u in $\mathbb{R}^k$ is possible. This contraction leads to a $d \times d$ second moment matrix which averages the examples of the different classes in the way prescribed by u. Any sensible, data-dependent way of picking a good set of vectors u should be able to reduce the dependence on k^2.

The same issues also arise with a continuous y: how to define and estimate the pairs of matrices whose generalized eigenvectors should be extracted is not immediately clear. Still, the case where y is multidimensional (vector regression) can be reduced to the case of univariate y using the same technique of contraction with a vector u. Feature extraction from a continuous y can be done by discretization (solely for the purpose of feature extraction), which is much easier in the univariate case than in the multivariate case.

In domains where examples exhibit large variation, or when labeled data is scarce, incorporating prior knowledge is extremely important. For example, in image recognition, convolutions and local pooling are popular ways to generate representations that are invariant to localized distortions. Directly exploiting the spatial or temporal structure of the input signal, as well as incorporating other kinds of invariances in our framework, is a direction for future work.

High dimensional problems create both computational and statistical challenges. Computationally, when $d > 10^6$, the solution of generalized eigenvalue problems can only be performed via specialized libraries such as ScaLAPACK, or via randomized techniques, such as those outlined in (Halko et al., 2011; Saibaba & Kitanidis, 2013). Statistically, the finite-sample second moment estimates can be inaccurate when the number of dimensions overwhelms the number of examples. The effect of this inaccuracy on the extracted eigenvectors needs further investigation. In particular, it might be unimportant for datasets encountered in practice, e.g., if the true class-conditional second moment matrices have low effective rank (Bunea & Xiao, 2012).

Finally, our approach is simple to implement and well suited to the distributed setting. Although a distributed implementation is out of the scope of this paper, we do note that aspects of Algorithm 1 were motivated by the desire for efficient distributed implementation. The recent success of non-convex learning systems has sparked renewed interest in non-convex representation learning. However, generic distributed non-convex optimization is extremely challenging. Our approach first decomposes the problem into tractable non-convex subproblems and then subsequently composes with convex techniques. Ultimately we hope that judicious application of convenient non-convex objectives, coupled with convex optimization techniques, will yield competitive and scalable learning algorithms.

6. Conclusion

We have shown a method for creating discriminative features via solving generalized eigenvalue problems, and demonstrated empirical efficacy via multiple experiments. The method has multiple computational and statistical desiderata. Computationally, generalized eigenvalue extraction is a mature numerical primitive, and the matrices which are decomposed can be estimated using map-reduce techniques. Statistically, the method is invariant to invert-

ible linear transformations, estimation of the eigenvectors is robust when the number of examples exceeds the number of variables, and estimation of the resulting classifier parameters is eased due to the parsimony of the derived representation.

Due to this combination of empirical, computational, and statistical properties, we believe the method introduced herein has utility for a wide variety of machine learning problems.

Acknowledgments

We thank John Platt and Li Deng for helpful discussions and assistance with the TIMIT experiments.

References

Agarwal, Alekh, Chapelle, Olivier, Dudík, Miroslav, and Langford, John. A reliable effective terascale linear learning system. *CoRR*, abs/1110.4198, 2011.

Blackard, Jock A and Dean, Denis J. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24 (3):131–151, 1999.

Bunea, F. and Xiao, L. On the sample covariance matrix estimator of reduced effective rank population matrices, with applications to fPCA. *ArXiv e-prints*, December 2012.

Dean, Jeffrey, Corrado, Greg, Monga, Rajat, Chen, Kai, Devin, Matthieu, Le, Quoc V., Mao, Mark Z., Ranzato, Marc'Aurelio, Senior, Andrew W., Tucker, Paul A., Yang, Ke, and Ng, Andrew Y. Large scale distributed deep networks. In *NIPS*, pp. 1232–1240, 2012.

Demmel, James, Dongarra, Jack, Ruhe, Axel, van der Vorst, Henk, and Bai, Zhaojun. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. Society for Industrial and Applied Mathematics, 2000.

Diamantaras, Konstantinos I and Kung, Sun Y. *Principal component neural networks*. Wiley New York, 1996.

Fisher, Ronald A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

Fisher, W., Doddington, G., and Marshall, Goudie K. The DARPA speech recognition research database: Specification and status. In *Proceedings of the DARPA Speech Recognition Workshop*, pp. 93–100, 1986.

Friedman, Jerome H. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405): 165–175, 1989.

Golub, Gene H and Van Loan, Charles F. *Matrix computations*, volume 3. JHU Press, 2012.

Goodfellow, Ian J, Warde-Farley, David, Mirza, Mehdi, Courville, Aaron, and Bengio, Yoshua. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Halevy, Alon, Norvig, Peter, and Pereira, Fernando. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009.

Halko, Nathan, Martinsson, Per-Gunnar, and Tropp, Joel A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Hinton, Geoffrey, Deng, Li, Yu, Dong, Dahl, George E, Mohamed, Abdel-rahman, Jaitly, Navdeep, Senior, Andrew, Vanhoucke, Vincent, Nguyen, Patrick, Sainath, Tara N, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29 (6):82–97, 2012a.

Hinton, Geoffrey E, Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012b.

Hutchinson, Brian, Deng, Li, and Yu, Dong. A deep architecture with bilinear modeling of hidden representations: Applications to phonetic recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4805–4808. IEEE, 2012.

Jose, Cijo, Goyal, Prasoon, Aggrwal, Parv, and Varma, Manik. Local deep kernel learning for efficient nonlinear svm prediction. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 486–494, 2013.

Koren, Yehuda, Bell, Robert, and Volinsky, Chris. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoff. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pp. 1106–1114, 2012.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, Ker-Chau. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

Livni, Roi, Lehavi, David, Schein, Sagi, Nachliely, Hila, Shalev-Shwartz, Shai, and Globerson, Amir. Vanishing component analysis. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 597–605, 2013.

Mairal, Julien, Bach, Francis, Ponce, Jean, Sapiro, Guillermo, and Zisserman, Andrew. Supervised dictionary learning. *arXiv preprint arXiv:0809.3083*, 2008.

Mika, Sebastian, Ratsch, Gunnar, Weston, Jason, Scholkopf, B, Smola, Alex, and Muller, K-R. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):623–628, 2003.

Platt, John C, Toutanova, Kristina, and Yih, Wen-tau. Translingual document representations from discriminative projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 251–261. Association for Computational Linguistics, 2010.

Rahimi, Ali and Recht, Benjamin. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2007.

Saibaba, Arvind K and Kitanidis, Peter K. Randomized square-root free algorithms for generalized hermitian eigenvalue problems. *arXiv preprint arXiv:1307.6885*, 2013.

Vapnik, Vladimir N. *Statistical learning theory*. Wiley, 1998.

Vershynin, Roman. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Wan, Li, Zeiler, Matthew, Zhang, Sixin, Cun, Yann L, and Fergus, Rob. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1058–1066, 2013.

Wold, Svante and Sjostrom, Michael. Simca: a method for analyzing chemical data in terms of similarity and analogy. *Chemometrics: theory and application*, 52:243–282, 1977.