Fast Pixel/Part Selection with Sparse Eigenvectors

Baback Moghaddam
Jet Propulsion Laboratory
California Institute of Technology
baback@jpl.nasa.gov

Yair Weiss
The Hebrew University
Jerusalem, Israel
yweiss@cs.huji.ac.il

Shai Avidan
Adobe Systems Inc.
Boston, MA USA
avidan@adobe.com

Abstract

We extend the "Sparse LDA" algorithm of [7] with new sparsity bounds on 2-class separability and efficient partitioned matrix inverse techniques leading to 1000-fold speed-ups. This mitigates the $O(n^4)$ scaling that has limited this algorithm's applicability to vision problems and also prioritizes the less-myopic backward elimination stage by making it faster than forward selection. Experiments include "sparse eigenfaces" and gender classification on FERET data as well as pixel/part selection for OCR on MNIST data using Bayesian (GP) classification. SparseLDA is an attractive alternative to the more demanding Automatic Relevance Determination. State-of-the-art recognition is obtained while discarding the majority of pixels in all experiments. Our sparse models also show a better fit to data in terms of the "evidence" or marginal likelihood.

1. Introduction

Spectral techniques have become an integral tool for learning and optimization in computer vision. Examples include Normalized Cuts [14] and Spectral Clustering [15], in which a *global* eigenvector solution minimizes a continuous (convex) relaxation of an otherwise NP-hard combinatorial objective function. Some form of post-processing is then applied to such *approximate* solutions in order to yield the desired segmentation, clustering or partition. More traditional use of spectral methods is in subspace dimensionality reduction and various factor-analytic techniques (*e.g.,* PCA, LDA, *etc.*) which are almost ubiquitous in dealing with intrinsically high-dimensions and/or densely-sampled data (*e.g.,* imagery and other spatial random fields).

The added constraint of *sparsity* however, fundamentally changes the nature of spectral techniques. Global eigenvectors are seldom sparse and ad-hoc methods of forcing them to be sparse (with post-processing) can lead to significant suboptimality. From the point-of-view of dimensionality reduction, lack of (input) sparsity is counter-productive as a non-sparse recipe (from even a single eigenvector) means using a (linear) combination of *all* the variables. Moreover, finding a low-dimensional manifold that is compactly parameterized by such derived (mixed) features is not always straightforward or even advantageous.

Sparsity is closely related to variable selection and automatic relevance determination (ARD), problems of enduring interest to the machine learning and statistics community. Sparseness implies selection and is typically invoked by means of continuous optimization with an l_1 norm penalty (*e.g.,* Lasso) or "relevance priors." For example, Roth & Lange [13] use a *wrapper* feature selection method by incorporating LDA into the M-step of an EM algorithm via regression (Fisher scoring). By imposing standard ARD (diagonal) Gaussian priors with Gamma hyperpriors they achieve the hierarchical Bayesian equivalent of variable selection with a Student-*t* type "norm."

In the computer vision community there has been much interest in visual learning of parts-based representations, often in the form of sparse bases. In face recognition (FR), the standard eigenfaces algorithm (PCA) has often been criticized for its lack of sparseness and topology. Alternative subspace methods like Local Feature Analysis (LFA) [11] were meant to directly address these shortcomings. Meanwhile, Bartlett *et al.* [1] proposed Independent Component Analysis (ICA) for FR as its basis functions exhibit sparseness. Similarly, Lee & Seung [6] advocated using Nonnegative Matrix Factorization (NMF) for visual representations, as non-negativity was deemed more neurologically plausible. More recently, Zass & Shashua [16] proposed Nonnegative Tensor Factorization (NTF) for use in parts-based representations (including sparse PCA). While non-negativity is an important constraint in some applications (*e.g.,* portfolio optimization), its use in image representation is primarily motivated by its tendency to promote sparsity. In contrast, the subspectral algorithms addressed in this paper do *not* employ non-negativity, as sparsity is directly imposed as a hard constraint (as opposed to being induced indirectly). Also, the appeal to the non-negativity of visual representation seems less compelling given recent findings of neurons which encode negative or *subtractive* responses.

2. Background

In statistics, several new techniques have been proposed for sparse spectral decomposition. Specifically, Zou *et al.* [17] proposed a sparse PCA algorithm (called SPCA) using an "Elastic Net" framework for l_1-penalized regression on regular PCs. Subsequently, d'Aspremont *et al.* [3] relaxed the hard cardinality constraint with a simpler *convex* approximation using semi-definite programming (SDP) for a more "direct" formulation (called DSPCA). In contrast, an alternative *discrete* spectral framework was recently proposed by Moghaddam *et al.* [8], using variational bounds on the covariance "subspectrum" derived by the eigenvalue *Inclusion Principle*. This discrete algorithm yielded substantial performance gains using a *greedy* search (GSPCA) and was also faster than continuous methods.

We extended this sparse EVD framework to *supervised* learning in [7], using a sparse reformulation of the Courant-Fischer "Min-Max" theorem for deriving *generalized* spectral bounds, thus subsuming sparse PCA as a special case of sparse LDA. Our variable selection algorithm functions as a *filter* (as opposed to a *wrapper*), using only 2nd-order statistics of the data — *e.g.,* in (Fisher) Linear Discriminant Analysis (LDA). Sparse LDA (SLDA) maximizes a *generalized* eigenvalue (generalized Rayleigh quotient) in a cardinality-constrained subspace (variable subset). This gives an exact formulation of sparse generalized EVDs and also suggests a simple post-processing step (variational renormalization) for improving continuous solutions. In [7] we also proposed an exact *optimal* algorithm (ESLDA) using branch-and-bound. However, our focus here will be on optimizing our greedy algorithm (GSLDA) specifically for binary (2-class) sparse linear discriminants.

3. Sparse Generalized EVD

Fisher Linear Discriminant Analysis (LDA) can be cast as a *generalized* eigenvalue decomposition (EVD), where given a symmetric matrix pair $A, B \in \mathcal{S}_+^n$, corresponding to the *between-class* and *within-class* covariance matrices respectively, we maximize a class-separability criterion defined by a generalized Rayleigh quotient (GRQ) $R(x) = (x^T A\, x)/(x^T B\, x)$. The optimal solution is the eigenvector corresponding to the maximal eigenvalue of $B^{-1/2}AB^{-1/2}$. Without the sparsity constraint the GRQ obeys the global bounds $\lambda_1(A, B) \leq R(x) \leq \lambda_n(A, B)$. Note that λ's are ranked in *increasing* order, thus $\lambda_{\min} = \lambda_1$ and $\lambda_{\max} = \lambda_n$.

The *sparse* version of LDA (or any GRQ cast as a G-EVD) is obtained by adding a *cardinality-constraint* on x:

$$\underline{\text{Sparse LDA}} \; : \quad \begin{array}{ll} \mathbf{max} & \dfrac{x^T A\, x}{x^T B\, x} \\ \text{subject to} & \text{card}(x) = k \end{array} \tag{1}$$

where $\text{card}(x)$ denotes the l_0 norm. However, this objective function is non-convex and NP-hard. Note that the special case of $B = I$ defaults to sparse PCA, therefore any algorithm for sparse LDA will also solve sparse PCA.

Optimality conditions are based on the equality

$$\frac{x^T A\, x}{x^T B\, x} = \frac{z^T A_k\, z}{z^T B_k\, z} \tag{2}$$

where $z \in \mathcal{R}^k$ is the nonzero subvector of x and (A_k, B_k) are the $k \times k$ principal submatrices of (A, B) obtained by deleting the rows/columns corresponding to the zero indices of x. The reduced quadratic form in z is equivalent to a standard *unconstrained* GRQ and since this subproblem's maximum is $\lambda_k(A_k, B_k)$, this *must* also be the optimal R. This reveals the true combinatorial nature of SLDA (or SPCA) wherein solving for the optimal solution is inherently a discrete search for the k indices which maximize the $\lambda_{\max}$ of a *subproblem* (A_k, B_k) — *i.e.,* the *subspectrum*.

Indeed, continuous solutions are only useful in yielding a sparsity pattern with which to solve an *unconstrained* subproblem in (A_k, B_k). Otherwise, they are typically suboptimal and must be "variationally renormalized" using the above equality. In [8] we showed that the *ad-hoc* method of "simple thresholding" (ST) — setting the smallest loadings to zero and renormalizing to unit-norm — is greatly enhanced by this "fix." We use this improved ST on the global Fisher vector of LDA in the experiments in Section 6.

3.1. Generalized Spectral Bounds

We have seen that $\lambda_{\max}(A_k, B_k)$ play a key role in defining SLDA solutions. But due to the combinatorial number of subspectra we prefer a more concise characterization by the $\lambda_i(A, B)$ which are readily available. The global spectrum and all its subspectra are indeed related.

Theorem 1 *Generalized Inclusion Principle* [8]. Consider the symmetric pair $A, B \in \mathcal{S}^n$ with generalized spectrum $\lambda_i(A, B)$. Let (A_k, B_k) be a corresponding pair of $k \times k$ principal submatrices with $1 \leq k \leq n$, and generalized subspectrum $\lambda_i(A_k, B_k)$. Then, for all $1 \leq i \leq k$

$$\lambda_i(A, B) \;\; \leq \;\; \lambda_i(A_k, B_k) \;\; \leq \;\; \lambda_{i+n-k}(A, B) \tag{3}$$

In other words, the generalized eigenvalues of (A, B) form upper and lower bounds for the generalized eigenvalues of all the principal submatrices (A_k, B_k). Indeed, the subspectra of (A_m, B_m) and (A_{m+1}, B_{m+1}) interleave or *interlace* each other, with the eigenvalues of the larger matrix pair "bracketing" those of the smaller one. For *positive-definite* symmetric matrices (covariances), augmenting A_m to A_{m+1} (adding a new variable) will always *expand* the spectral range: reducing $\lambda_{\min}$ and increasing $\lambda_{\max}$. This *monotonicity* has important theoretical and practical consequences for combinatorial optimization.

Since SLDA seeks to *maximize* the GRQ, the relevant inequality in Eq.(3) is the one with $i = k$, thus yielding

$$\lambda_k(A, B) \;\leq\; \lambda_{\max}(A_k, B_k) \;\leq\; \lambda_n(A, B) \qquad (4)$$

This shows that the k-th smallest eigenvalue of (A, B) is a lower bound for the class-separability criterion of sparse LDA with cardinality k (see also Section 4).

Given this discrete search formulation, branch-and-bound techniques [10] are ideally suited for sparse LDA. In [7], the generalized inclusion bounds are used for exact search (ESLDA) to find globally optimal solutions, albeit for smaller problems ($n < 40$) since branch-and-bound can exhibit exponential worst-case complexity.

Greedy techniques like *backward elimination* can also exploit the monotonic nature of nested submatrices and their "bracketing" eigenvalues: start with the full index set $I = \{1, 2, \ldots, n\}$ and sequentially delete the variable j which yields the maximum $\lambda_{\max}(A_{\setminus j}, B_{\setminus j})$ until only k elements remain. For *small* cardinalities $k << n$, the high cost of backward search makes its forward counterpart *forward selection* more attractive (despite it being potentially "myopic"): start with the null index set $I = \{\}$ and sequentially add the variable j which yields the maximum $\lambda_{\max}(A_{+j}, B_{+j})$ until k elements are selected.

In [8, 7], we proposed a simple bi-directional greedy search: pick the better of 2 solutions found by a forward and (independent) backward pass. This simple strategy has led to remarkably good results (*e.g.,* out-performing a variety of continuous algorithms). The dual-pass search has the added benefit of giving near-optimal solutions for *all* cardinalities (at once), with a complexity that is less demanding than finding single k solutions (one at a time).

4. Sparsity Bounds for Binary SLDA

For general multiclass sparse LDA (full-rank A) the objective value of any solution of cardinality k is bounded from below by the k-smallest global eigenvalue $\lambda_k(A, B)$. This bound characterizes the worst-case scenario and can be used to choose the smallest k that guarantees a minimum required variance (class-separability). However, this bound becomes trivially zero for 2-class SLDA, since $\lambda_k(A, B) = 0$ for all $k < n$ due to the rank-1 A matrix.

Using the trace inequality for matrix products on subproblems of size k, we derive the following new bound on the objective value $\lambda_k(A_k, B_k) = a_k^T B_k^{-1} a_k$

$$\mathrm{Tr}(B_k^{-1} a_k a_k^T) \;\geq\; \frac{\mathrm{Tr}(a_k a_k^T)}{\lambda_k(B_k)} \;\geq\; \frac{S_k}{\lambda_{\max}(B)} \qquad (5)$$

where S_k is the sum of the k largest diagonal elements of A. The $r.h.s.$ of Eq(5) is a global (worst-case) lower bound (*i.e.* it applies to all possible solutions of cardinality k regardless of how sub-optimal they may be). To illustrate,

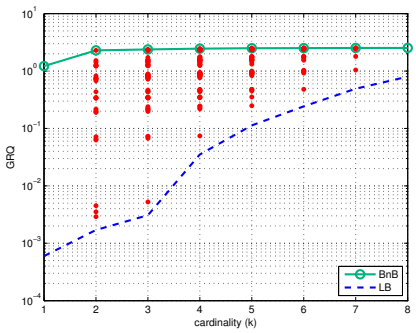


Figure 1. A sample lower bound on the binary SLDA objective.

Figure 1 plots this lower bound for a sample problem along with the global optima found by branch-and-bound search. Also shown (in red) are the locally optimal solutions for every possible sparsity pattern. For example, this lower bound indicates that in order for every subproblem found (no matter how suboptimal) to capture GRQ$(k) \geq 0.1$, the cardinality must be $k \geq 5$. While it is possible to get tighter (best-case) bounds by using the full spectrum $\lambda_i(B)$, the bound in Eq(5) is one of the simplest, as it only relies on the diagonal elements of A and $\lambda_{\max}(B)$.

In addition to these global (worst-case) bounds, some best-case bounds for greedy search can be derived from Theorem 1. For example, in [8] we show that SLDA obeys certain "nesting" bounds on backward search, where among all the n possible $(n-1)$-by-$(n-1)$ principal submatrices of the pair (A, B), obtained by deleting a single (say j-th) row and column, there is *at least* one whose objective value is no less than $\frac{n-1}{n}$ of $\lambda_{\max}(A, B)$

$$\max_j \;\lambda_{\max}(A_{\setminus j}, B_{\setminus j}) \;\geq\; \frac{n-1}{n}\,\lambda_{\max}(A, B) \qquad (6)$$

This nesting bound can be applied recursively in backward search mode to show, for example, that our greedy solutions are guaranteed to achieve no less than the fraction k/n of the initial (global) $\lambda_{\max}(A, B)$. In actual practice of course, one captures far more variance than what this linear bound indicates, due to the overly pessimistic assumptions implicit in the recursion of Eq(6). In addition to having guarantees for GSLDA itself, lower bounds on $\lambda_{\max}$ can be used to find *minimax* error bounds for hyperplane classifiers [5].

5. Efficient Eigenvalue Computation

Discrete algorithms for the *general* (full-rank) case of sparse PCA/LDA will require $O(k^3)$ EVDs for each subproblem (A_k, B_k). This is essentially unavoidable and leads to the usual difficulties with scaling of complexity: forward search has $O(n^3)$ (or less) whereas backward search has $O(n^4)$ (or more). The latter limits the use of the (usually) more accurate *backward* search for large n.

Nevertheless, greedy algorithms were still more efficient (and accurate) than continuous ones (see details in [8]).

Fortunately, for the special case of *binary* classification the required GSLDA computations can be made exceedingly efficient as the only finite eigenvalue $\lambda_{\max}(A_k, B_k)$ can be computed in closed-form as $a_k^T B_k^{-1} a_k$. This is due to the rank-1 A matrix in the GRQ numerator being a simple outer-product $A = aa^T$. Hence the computational complexity of 2-class GSLDA hinges on our ability to invert B_k submatrices "on-the-fly." A naive implementation, even with a Cholesky decomposition, is still grossly inefficient as B_k and $B_{k\pm1}$ differ by a single row/column. Therefore, partitioned matrix inverse techniques [4] and simple rank-1 updates for the required B_k^{-1} are highly recommended. These implementation details are given in the **Appendix**.

Moreover, by computing the *increments* of change in the GRQ (instead of final values), intermediate terms (matrix-vector byproducts) will cancel, leading to an essentially "loop-free" array computation over the available indices considered for inclusion/deletion. Consequently, these optimized algorithms offer a significant speed-up for 2-class GSLDA (*e.g.,* by several orders of magnitude). This opens up the possibility of applying subspectral optimization techniques to a wider range of problems such as pixel or object part selection in computer vision.

For example, a full (dual-pass) run of the algorithm in [7] for a matrix of size $n = 1024$, using "on-the-fly" Cholesky computation of $\lambda_{\max}(A_k, B_k)$, takes approximately 12 hours of computation (in Matlab 7.2 on a 3.2GHz P4) where 80% of the cputime is taken up by the *backward* pass. In stark contrast, our implementation using rank-1 updates on partitioned inverses requires only 2 minutes, where the backward pass now takes up only 40% of the total cputime (due to its simpler rank-1 updates, see Appendix). This represents a speed-up factor of 340. For even larger matrices ($n > 2000$) the cputimes were $\sim 10^3$ faster than those of the default GSLDA. For example, with $n = 2048$ the original algorithm required about 12 days of cputime whereas our optimized version required just 20 minutes, with backward search using only $1/3$ of the total cputime.

6. Experiments

Before presenting the experimental results with binary GSLDA, we first illustrate pixel/part selection using GSPCA, by computing "sparse eigenfaces" for 2D FR and sparse eigenshapes (segments) for 3D face modeling. Although sparse PCA means working with unlabeled data in an *unsupervised* setting, these examples will motivate the GSLDA results in the next section.

Figure 2-(left) shows the average face of a database of roughly 11,000 20-by-16 pixel images of frontal faces of 370 individuals obtained in a real access control setting (in a factory) on 2 separate days, in 2 different locations and under 2 different lighting conditions. Figure 2-(middle) shows our full mask which totals $n = 174$ pixels. We have downsampled the data specifically for this illustrative example.

Figure 2. Mean face, full mask and sparse mask with GSPCA.

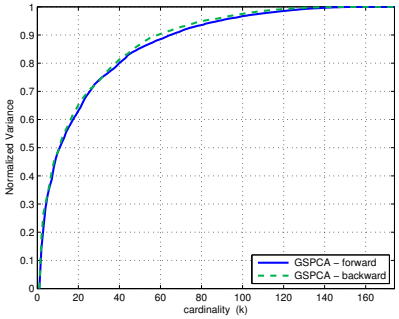


Figure 3. Variance curves for forward & backward GSPCA.

We compare standard eigenfaces (PCA using the full mask) to its sparse version using the subset of pixels found by sparse PCA. We compute the full covariance matrix of size $n = 174$ and apply the dual-pass GSPCA algorithm [8]. The resulting variance curves are shown in Figure 3 where the forward/backward passes find sparse solutions of comparable variance (true only at this low resolution). Note that by using only half the pixels ($k = 85$) there is no appreciable loss compared to the total variance of a full mask ($k = 174$). The selected pixels, corresponding to the sparsity pattern of the first GSPCA eigenvector of cardinality $k = 85$ is shown in Figure 2-(right). Note that the pixels in the eyes were not selected because at this low resolution there is almost no variability in the appearance of eyes (this however is not the case at higher resolutions).

Using the sparsity pattern found, we compute eigenfaces for both the full and sparse masks as shown in Figure 4 where we have kept the same sparsity mask for the higher-order sparse eigenfaces (for simplicity). Note the subtle differences between the pixels (parts) common to the different sets of bases. Sparse eigenvectors are encoding different (local) aspects of the data than the global ones. We now compare the verification performance of the two methods by using the same number of basis functions (100 in this case) in encoding each face. To calibrate the overall difficulty of the verification task (matching 7400 probes to 3700 galleries) we note that a leading commercial FR system achieved an EER of about 1% on this data but at 16 times higher resolution (80x64 pixel frame). Figure 5 shows the ROC curves obtained, where the sparse model using the

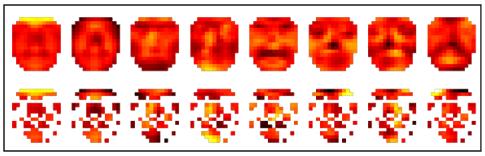


Figure 4. Top 8 eigenfaces with full and sparse masks.

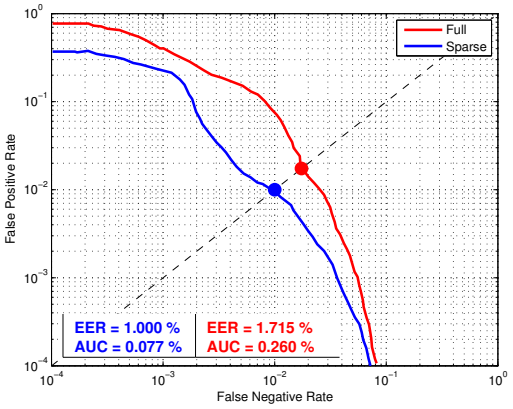


Figure 5. ROC for full and sparse eigenfaces.

same number of basis functions shows better FPR and FNR at every operating point. The EER has dropped from 1.7% down to 1.0% and the AUC from 0.26% down to 0.08%. This is encouraging not only in terms of verification performance, but also considering that the sparse model required half as many basis pixel values to be stored for projections. Alternatively, we could say that the 100 GSPCA bases were modeling only 85 pixels as opposed to 174, thereby increasing their explanatory power. Either way, it is noteworthy that nearly all pixels omitted by GSPCA correspond to near-uniform (redundant) parts of the face like the cheeks (or eyes in this low-resolution case).

It is also possible to use sparse eigenvectors for modeling 3D shape. In fact, the "Morphable Model" approach for 3D face modeling [2], often uses modular PCA segments for a coarse parts-based representation of shape (for greater flexibility). The full face mesh is (re)constructed by stitching together (blending) these individual part meshes (which can overlap). However, this segmentation is mostly a convenient ad-hoc partition into the usual eye-region, mouth-region, *etc.* In contrast, sparse PCA can discover shape parts automatically in a purely statistical (data-driven) manner. Figure 6 shows the vertex pattern of 3 sparse eigenvectors obtained using (forward) GSPCA on the XYZ mesh, in standard coarse-to-fine fashion (for reducing computation). Note that selected regions (shown textured) do covary, like protruding chins and noses or symmetric left/right cheeks, and are indeed jointly coupled face parts. These segments are quite different from the *a prior* partitions into eye/nose/mouth. This sparse analysis is more "correct" in letting the shape statistics dictate the 3D part segmentation.

Figure 6. 3D parts found using GSPCA on XYZ data.

6.1. Pixel Selection with GSLDA

We evaluated our optimized GSLDA variable selection method (or "hard-ARD") along with simple thresholding of the Fisher discriminant (the global eigenvector of LDA). We also compared these to the "soft-ARD" estimation of hyperparameters with marginal likelihood maximization. In all the experiments we used the GPML Matlab toolbox of Rasmussen & Williams [1] for both GP classification and regression (see [12] for details). All computations were carried out with Matlab 7.2 on a 3.2GHz Pentium 4. In Section 6.1.1 we give a detailed account of USPS digit classification using GSLDA and a similar treatment in Section 6.1.2 for gender classification on the FERET database. Both of these datasets have similar data dimensionality and number of training/testing cases.

6.1.1 USPS Digit Classification

We compared GSLDA hard-ARD to the Fisher thresholded pixel selection for classification of 16-by-16 USPS digits "3" *vs.* "5". We used a balanced partition of the USPS data into 767 training cases split 406/361 for the digits "3" and "5" and a test set of 773 cases split into 418/355, following the evaluation protocol in [12]. Using the class means in the training set (Figure 10(a,b)) and an estimate of the within-class covariance B, we applied GSLDA to find hard-ARD solutions for each cardinality k. The total cputime of the optimized dual-pass GSLDA was less than 2 seconds. Figure 7 shows the resulting forward/backward GSLDA objectives for all values of k. Note that backward search yields better solutions than the forward search and that the simple (global) technique of Fisher thresholding is uniformly inferior to GSLDA in terms of the captured variance.

For every selected pixel-set of size k, we trained corresponding GPCs using both Laplace and EP approximations of the latent posterior (with probit likelihoods), including separate hyperparameter estimation by maximizing marginal likelihood. The kernel function used was the squared-exponential with 2 hyperparameters: a signal variance term and a *common* lengthscale (*i.e.,* a non-ARD kernel) — see [12] for details of these computations. The main performance criteria used were *test error* and the test set's averaged log predictive probability or *test information*,

[1] http://www.gaussianprocess.org/gpml/code

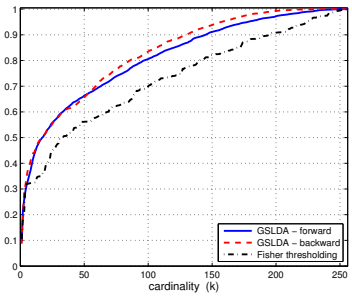


Figure 7. Hard-ARD for USPS digits 3-vs-5: GRQ objectives for greedy forward/backward search (GSLDA) and Fisher thresholding *vs.* cardinality of selected pixel-set.

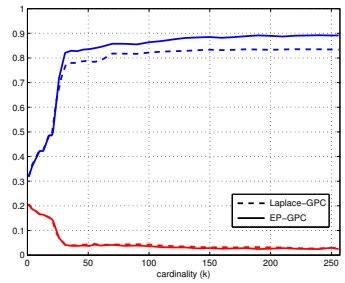


Figure 8. Laplace and EP GPC on USPS digits 3-vs-5: profiles of **test information** (blue) and **test error** (red) *vs.* the cardinality of hard-ARD pixel-sets found by GSLDA. All pixels shared a common length-scale hyperparameter in the covariance kernel.

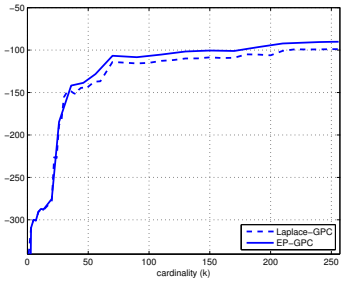


Figure 9. Laplace and EP GPC on USPS digits 3-vs-5: profiles of log **marginal likelihood** *vs.* the cardinality of the hard-ARD pixel sets found by GSLDA. Selected pixels had a common length-scale hyperparameter in the covariance kernel.

expressed in terms of the fraction of the available bit captured by the GPC for making predictions (*i.e.,* essentially the balance of entropy in model predictions *vs.* target labels, hence 1.0 bits being ideal). The results are shown in Figure 8 which show that we require much less than all $n = 256$ pixels to make accurate predictions, as pixel-sets of size $k = 150$ suffice in terms of test error and test information. In fact, it is mostly for the sparsest regimes ($k < 50$) that performance seriously degrades. This is confirmed by the log marginal likelihoods shown in Figure 9.

We next examine the alternative *soft*-ARD by estimating individual pixel lengthscales with the marginal likelihood.

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

Figure 10. USPS digits 3-vs-5: class means (a) & (b), soft-ARD relevance maps (inverse-lengthscale hyperparameters) for Laplace-GPC (c) and EP-GPC (d), Fisher loadings (e) and location of the optimal single-pixel found by GSLDA (f).

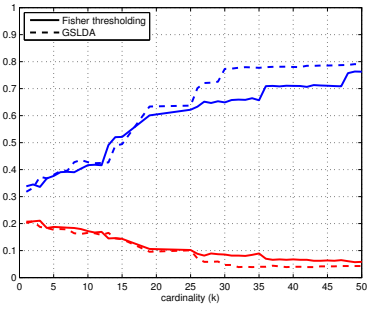


Figure 11. Laplace GPC on USPS digits 3-vs-5: comparing **test information** (blue) and **test error** (red) with pixel subsets found by Fisher thresholding (solid) and GSLDA (dashed) for $k \leq 50$.

The squared-exponential kernel now requires 257 hyperparameters (1 for each pixel and 1 for the overall signal variance/scale). This is a much more demanding computation, requiring a conjugate-gradient search in a 257-dimensional space (as opposed to 2-dimensional) and requires far more cputime. To appreciate the added burden, soft-ARD required approximately 4 minutes of cputime for the Laplace-GPC and 21 minutes for the EP-GPC (independently of k). We contrast this with the mere 2 seconds required by GSLDA for *all* k. There is also the added concern of having more local optima in the marginal likelihood with this many hyperparameters and an increased risk of over-fitting.

The saliency (relevance) of individual pixels can be visualized by displaying the inverse-lengthscales in image format as in Figure 10(c) and (d), for Laplace and EP GPCs, respectively. Although not easy to interpret, there does appear to be more relevance associated with pixels near the upper right-hand cusp of the digit "5" (where it differs markedly from "3"). By comparison, we show the corresponding loadings (absolute-value elements) of the Fisher discriminant (LDA eigenvector) in Figure 10(e) which bears a resemblance to the relevance maps in (c) and (d). Note that ranking the elements of this eigenvector was how we obtained the pixels-sets for Fisher thresholding in Figure 7.

Using the soft-ARD relevance maps in (c) and (d), we can also rank pixels according to inverse-lengthscales to form subsets of various size k. But this turns out not to yield the best performance (same as with Fisher thresholding, to be presented shortly). In fact, visual inspection of the relevance maps in Figure 10(c)-(e) might lead us to conclude that the upper-right cusp (gap) of the digit "5" is in

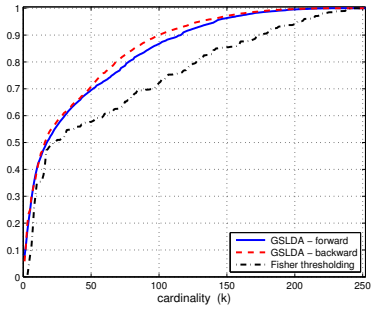


Figure 12. Hard-ARD for FERET faces M-vs-F: GRQ objective values for greedy forward/backward search (GSLDA) and Fisher thresholding *vs.* pixel-set size k.

fact the most salient object part. Indeed, all 3 maps seem to confirm this. However, this location does *not* correspond to the most discriminant part. In fact, the single most discriminant pixel, found by GSLDA, is shown in Figure 10(f) and corresponds to the upper-*left* cusp (gap) of the digit "3" instead (note that this is the *optimal* single pixel in terms of maximal GRQ). Indeed, it is through a combination of these two parts (cusps) that the best discrimination is obtained. Thus the soft-ARD relevance maps in (b) and (c) could (in this case) mislead us into focusing on the wrong pixels. This mis-specification is more problematic in the sparsest of regimes only, as shown in Figure 11 where we compare the test error and test information for a Laplace-GPC for $k < 50$ (similar results were obtained for an EP-GPC). Clearly, GSLDA subsets are doing better. However, beyond $k = 100$ there are sufficiently many different pixels to make both subsets perform satisfactorily (this is a common phenomenon with highly-correlated spatial data). In Figure 11 we also see that in the extreme case of using a *single* pixel, we do in fact do twice better than chance, but only by choosing the optimal pixel shown in Figure 10(f).

6.1.2 FERET Gender Classification

We next apply the same pixel selection protocol to FERET gender classification. We use 21-by-12 "thumbnails" in order to compare to a previous study which benchmarked different classifiers [9]. 1755 such thumbnails (1044 M and 711 F) were split in half (878/877) for train/test partitions.

Figure 12 shows the GSLDA results. Once again backward search performs better than forward and Fisher thresholding is inferior for all k. Total cputime of GSLDA was less than 2 seconds. The test error and test information in Figure 13 indicate that far fewer than $n = 252$ pixels suffice for making accurate predictions (even at this low resolution). Once again, only in the sparsest regimes ($k < 50$) does performance seriously degrade. This is confirmed by the log marginal likelihoods in Figure 14, where the best model fit actually favours the range $30 \leq k \leq 100$.

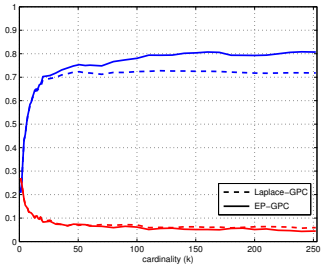


Figure 13. Laplace and EP GPC on FERET faces M-vs-F: profiles of **test information** (blue) and **test error** (red) *vs.* cardinality of hard-ARD pixel sets found by GSLDA. All pixels shared a common length-scale hyperparameter in the covariance kernel.

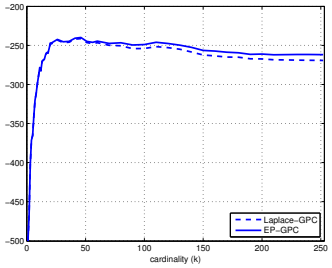
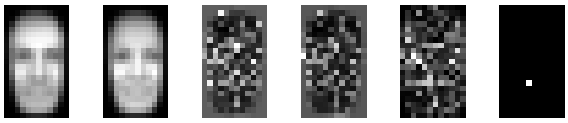


Figure 14. Laplace & EP GPC on FERET faces M-vs-F: plot of log **marginal likelihood** *vs.* size of hard-ARD pixel-sets of GSLDA. Selected pixels had common length-scale hyperparameters.



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

Figure 15. FERET M-vs-F: class means (a) & (b), soft-ARD relevance maps (inverse-lengthscale hyperparameters) for Laplace-GPC (c) and EP-GPC (d), Fisher loadings (e) and location of the optimal single-pixel found by GSLDA (f).

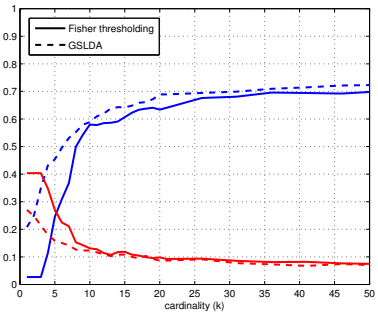


Figure 16. Laplace GPC on FERET faces M-vs-F: plots of the **test information** (blue) and **test error** (red) of pixel-sets found by Fisher thresholding (solid) and GSLDA (dashed) for $k \leq 50$.

The M/F class means in Figure 15(a,b) indicate the far greater subtlety of this discrimination task (compared to USPS). Figure 15 shows soft-ARD relevance maps in (c,d)

and Fisher loadings in (e). These maps are even harder to interpret. Their maxima do not correspond to the best single-pixel, shown in (f). They also select the wrong pixels for $k < 50$. Figure 16 shows a big deficit in test error for Fisher thresholding (similar or worst results were found for soft-ARD). For $k < 4$ Fisher subsets have only chance level performance (here 40% due to the 60/40 gender mix of the data). In the extreme single pixel case we once again do much better than chance. Not surprisingly, this pixel is in the upper lip where facial hair (mustache) readily distinguishes gender. Our best error rate (4.5% with EP) is comparable to [9] using SVMs with *all* 252 pixels (3.4%), since that study used an 80/20 split as opposed to our 50/50.

7. Discussion

By using GSLDA as a *filter* we decouple the sparsity estimation from the subsequent inference stage. In contrast, others embed sparsity estimation within the inference, which works best when sparsity already exists or can at least be induced by the prior. Our approach *enforces* sparsity as a hard constraint and leaves the inference stage (relatively) intact and much simpler. This is partly justified by the fact that a full Bayesian treatment of sparsity is still NP-hard (as the partition function has 2^n terms). In fact, even "exact" inference with sparsity-inducing priors like the Laplacian is only approximating a cardinality constraint, since the convex l_1 norm is being used as a *surrogate* for the l_0 norm.

The GSLDA algorithm of [7] is highly effective, capturing more variance than all continuous algorithms currently available. But the complexity of its backward search has up to now limited its range of applications. Our partitioned matrix improvements lead to $\sim 10^3$ speed-ups and a state-of-the-art algorithm for 2-class SLDA, which readily extends to sparse *nonlinear* discriminants using kernel methods.

Appendix

For forward greedy search, let s be the current subset of k indices and $t = s \cup i$ for a candidate $i \notin s$. Given the current inverse B_{ss}^{-1}, the new augmented inverse is

$$B_{tt}^{-1} = \begin{bmatrix} B_{ss}^{-1} + r_i v_i v_i^T & -r_i v_i \\ -r_i v_i^T & r_i \end{bmatrix}$$

where $v_i = B_{ss}^{-1} B_{si}$ with (si) indexing the s rows and i-th column of B and the scalar $r_i = 1/(B_{ii} - B_{si}^T v_i)$. The corresponding SLDA objective is $\lambda_{\max}(A_t, B_t) = a_t^T B_{tt}^{-1} a_t$. If we expand the expression for the *incremental* change $\Delta_i = \lambda_{\max}(A_t, B_t) - \lambda_{\max}(A_s, B_s)$ intermediate terms cancel, leading to a (loop-free) array computation for Δ.

For backward greedy search (going from the index set t down to s), by partitioning the current inverse as follows

$$B_{tt}^{-1} = \begin{bmatrix} P_{ss} & q_i \\ q_i^T & z_i \end{bmatrix}$$

a simpler rank-1 update results: $B_{ss}^{-1} = P_{ss} - q_i q_i^T / z_i$. Once again, by solving for the increments Δ_i, many redundant calculations can be avoided. This backward computation is now even more efficient than the forward one, since "growing" an inverse is harder than "shrinking" it.

Acknowledgments

Research conducted at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

[1] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. Face recognition with independent component analysis. *IEEE Transactions on Neural Networks*, 13(6), 2003.

[2] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model. *IEEE PAMI*, 25(9), 2003.

[3] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A Direct Formulation for Sparse PCA using Semidefinite Programming. In *Neural Information Processing Systems 17*, 2004.

[4] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Press, 1989.

[5] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. Minimax Probability Machine. In *Neural Information Processing Systems 14*. 2002.

[6] D. D. Lee and H. S. Seung. Learning Parts of Objects with Nonnegative Matrix Factorization. *Nature*, 401, 1999.

[7] B. Moghaddam, Y. Weiss, and S. Avidan. Generalized Spectral Bounds for Sparse LDA. In *International Conference on Machine Learning*. ICML'06, June 2006.

[8] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral Bounds for Sparse PCA: Exact & Greedy Algorithms. In *Neural Information Processing Systems 18*, 2006.

[9] B. Moghaddam and M.-H. Yang. Learning Gender with Support Faces. *IEEE PAMI*, 24(11), 2002.

[10] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley, New York, 1988.

[11] P. Penev and J. Attick. Local Feature Analysis: A General Statistical Theory for Object Representation. *Network: Computation in Neural Systems*, 7(3), 1996.

[12] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[13] V. Roth and T. Lange. Feature Selection in Clustering Problems. In *Neural Information Processing Systems 16*. 2004.

[14] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE PAMI*, 22(8), 2000.

[15] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *International Conference on Computer Vision*, 1999.

[16] R. Zass and A. Shashua. Nonnegative Sparse PCA. In *Neural Information Processing Systems 19*. 2007.

[17] H. Zou, T. Hastie, and R. Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15(2), 2003.