Subspace Scores for Feature Selection in Computer Vision

Cameron Musco
cnmusco@mit.edu

Christopher Musco
cpmusco@mit.edu

Abstract

Feature selection has become an essential tool in machine learning – by distilling data vectors to a small set of informative dimensions, it is possible to significantly accelerate learning algorithms and avoid overfitting. Feature selection is especially important in computer vision, where large image vectors are often combined with huge synthetically generated feature sets.

Inspired by recent theoretical work on dimensionality reduction for k-means clustering, we introduce an unsupervised feature selection method and evaluate its performance on fundamental vision tasks. Our approach is based on a new measure for feature importance, 'subspace scores', which we derive from statistical leverage scores. Sampling by subspace scores is provably effective for k-means clustering and low rank approximation – it can significantly reduce the number of features in a dataset before solving these problems and still get nearly optimal results [11]. We verify this work by applying subspace sampling to clustering handwritten digit and face datasets.

Additionally, we address a broader range of applications by proposing subspace scores as a general purpose feature selection tool for vision data, akin to Fisher [18] or Laplacian scores [39]. We provide a theoretical justification for subspace scores, in addition to experimental results. Evaluation on a facial recognition task is promising: subspace scoring significantly outperforms eigenface methods and a variety of standard unsupervised feature selection algorithms. In fact, it is nearly competitive with popular supervised methods for selection. These preliminary experimental results justify further exploration for vision applications.

1. Introduction

In modern machine learning tasks, the number of available data features is typically enormous – often larger than the number of available data examples. While the ability to collect and store rich data is essential for building accurate models, features are usually polluted by noise and redundancy, which can make it difficult to extract meaningful information [10]. Extraneous features slow learning algorithms and can lead to overfitting via the *curse of dimensionality* or increase the chance that certain optimization algorithms get stuck at local minima [16].

Feature overload is especially problematic in computer vision. Not only do images contain many pixels, each encapsulating several color values, but synthetic data construction is common in computer vision. Feature generation algorithms are used to extract highly non-linear structure from pixels using, for example, Fourier information, Gabor energy functions, SIFT features [31], or HOG features [14]. The potential number of features available for a given vision problem is essentially unlimited [4].

1.1. Dimensionality Reduction

To cope with feature overload, *dimensionality reduction* has become an important tool in any machine learning toolkit, including for vision applications. The goal is to significantly reduce the number of features in a dataset before running a learning algorithm. Ideally, eliminating features only leads to a minor reduction in the algorithm's effectiveness or, by avoiding overfitting and local minima, actually improves learning performance [10]. Dimensionality reduction breaks into two main categories [29]:

Feature Extraction A feature extraction algorithm generates a small set of *new* data features by transforming the original data. Principle Component Analysis (PCA) is a standard example of linear feature extraction, although non-linear methods are common as well. The bag-of-words model is especially popular in computer vision [13, 35].

Feature Selection A feature selection algorithm chooses and possibly reweights a small set of *original* data features based on some measure of importance, like feature variance, Laplacian score, or Fisher score. Either the highest ranked features are chosen or features are randomly sampled, with probability proportional to importance.

Both types of dimensionality reduction are popular and a variety of techniques have been studied theoretically and empirically. Theoretical analysis is often inspired by observed effectiveness in practice and, in turn, often inspires new algorithms that work well experimentally.

1.2. Applications to Clustering

One area that has seen an especially fruitful interchange between theory and practice is dimensionality reduction for

clustering problems. The goal is to reduce the dimension of data points significantly before clustering, hoping to obtain a good partition in a fraction of the time. Research has focused heavily on the ubiquitous k-means clustering objective, an unsupervised learning technique included in a recent list of 'Top 10 Algorithms in Data Mining' [38].

It has long been observed that k-means can be accelerated by projecting data points to a small set of top principal data components before clustering, without sacrificing quality. Typically, $O(k)$ principal components are used, which can be a substantial reduction since k, the number of target clusters, is usually small. Inspired by this observation, several theoretical results attempt to understand the connection between k-means and PCA, noting that PCA can be characterized as a relaxation of k-means clustering [40, 15].

This fact was applied rigorously to show that projecting to $O(k)$ principal components as a dimensionality reduction step is *guaranteed* to give a good k-means clustering [17, 19]. In recent work, we tighten these results and, using the same basic ideas, prove that several alternative reduction techniques are also useful for k-means [11]. In particular, inspired by preliminary results in [7], we introduce a feature selection algorithm that chooses a subset of original data features based on *subspace scores* ([11], Theorem 14). This new measure of statistical importance is based on leverage scores, an important concept from linear regression.

Subspace score sampling also applies to a general class of *constrained low rank approximation* problems, which includes k-means clustering, unconstrained PCA, and several other natural problems – provably good solutions are obtainable even after significant dimensionality reduction.

1.3. Goals

However, the theoretical results in [11] remain untested in practice, leading to several natural directions for future experimental research. This paper seeks to initiate an investigation through canonical computer vision applications – high dimensional image data is an ideal test bed for dimensionality reduction algorithms. We ask two main questions:

1. Is subspace scoring effective for k-means in practice?

PCA and other methods evaluated theoretically in [11] (e.g. Johnson-Lindenstrauss embedding [24]) have been tested extensively [15, 20]. PCA in particular works remarkably well, so we are interested in understanding whether subspace scoring can give comparable results. Feature selection methods are often preferred in practice over feature extraction because they maintain data interpretability and can preserve important data structure (e.g. sparsity). Selection also incurs a small space overhead, simply requiring pointers into existing data matrices. For computer vision, which often relies on synthetic features, it can eliminate the need to compute many features in the first place.

We give evidence towards an affirmative answer of Question 1 in Section 3 through an application to handwritten digit and face clustering. Subspace scoring shows promise as the first provably accurate and practically effective feature selection method for k-means clustering.

2. Given their applicability to k-means and a variety of low rank approximation problems, are subspace scores effective for general purpose feature selection in vision?

Again, we focus on feature selection since Principal Component Analysis and Johnson-Lindenstrauss methods have been evaluated as general dimensionality reduction tools. In the case of PCA, so called *eigenspace methods* are fundamental for objection recognition [33], motion tracking [5], image modeling [12], and face recognition [37].

Our work points to an affirmative answer to this second question as well. In Section 4.1, we give new mathematical interpretation that suggests subspace scores may be a valuable metric beyond their intended purpose for k-means clustering. Additionally, in Section 4.2 we address Question 2 experimentally via an application to facial recognition. When used as a dimensionality reduction technique before nearest neighbor classification, subspace score sampling significantly outperforms PCA (eigenfaces) and all other unsupervised feature selection methods tested. Subspace scores are only outclassed by a supervised Fisher score method and a supervised Laplacian score variant, which both use label information to choose features.

2. Background

Before evaluating subspace scores in Sections 3 and 4, it is helpful to give a brief introduction to the k-means clustering problem and relevant dimensionality reduction techniques, including our recent results in [11], which inspired this study of subspace scores for vision.

2.1. k-means Clustering

Data clustering is one of the most common approaches to *unsupervised learning*. Without access to label information, clustering algorithms use some distance metric to partition data points into groups that seem likely to share a common label. In computer vision, clustering is used for image segmentation, face detection, training bag-of-words classifiers, and a variety of other common tasks [36].

k-means is a particularly natural clustering *objective function*, although the name is sometimes used to reference a common heuristic, Lloyd's algorithm [30], for minimizing the objective. We use the former meaning.

The goal is to partition data into k clusters that minimize total intra-cluster variance. Suppose we wish to divide n vectors in $\mathbb{R}^d$, $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, into clusters, $\{C_1, \ldots, C_k\}$. Let $C(\mathbf{a}_j)$ denote the cluster $\mathbf{a}_j$ is assigned to. Let $\boldsymbol{\mu}_i$ be the centroid (i.e. average) of the vectors in C_i. Then, the

optimal k-means clustering is given by

$$\arg\min_{\{C_1,\ldots,C_k\}} \left(\sum_{j=1}^{n} \|\mathbf{a}_j - \boldsymbol{\mu}_{C(\mathbf{a}_j)}\|_2^2 \right).$$

In other words, we just compute the squared ℓ_2 distance from every data vector to its cluster centroid. Our goal is to minimize the sum of these distances. By rearranging our summation, we see that this is equivalent to computing the variance for each cluster and summing over all clusters:

$$\arg\min_{\{C_1,\ldots,C_k\}} \left(\sum_{i=1}^{k} \sum_{\mathbf{a}_j \in C_i} \|\mathbf{a}_j - \boldsymbol{\mu}_i\|_2^2 \right).$$

In general, this function is NP-hard to optimize [1]. Nevertheless, it can be solved effectively in practice using either heuristic algorithms (e.g. Lloyd's algorithm) or a variety of provably good approximation algorithms [30, 25, 2, 22].

2.2. Dimensionality Reduction for k-means

Any of these algorithms can be accelerated by preprocessing data to reduce the dimension of each point $\mathbf{a}_j$ from d to some $d' \ll d$. Of course this can be done in a huge variety of ways, so what do we mean for a dimensionality reduction method to be provably good for k-means? Denote our original data matrix, whose rows are the vectors $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, as $\mathbf{A} \in \mathbb{R}^{n \times d}$. Denote the data matrix containing our dimension reduced rows as $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times d'}$. For any clustering C, we will write the k-means cost for data matrix $\mathbf{A}$ and clustering $C = \{C_1, \ldots, C_k\}$ as $Cost(C, \mathbf{A})$.

Suppose C^* is the optimal clustering for $\mathbf{A}$ and $\tilde{C}^*$ is the optimal clustering for $\tilde{\mathbf{A}}$. One natural goal is to seek out dimensionality reduction methods that guarantee

$$Cost(\tilde{C}^*, \mathbf{A}) \leq \lambda \cdot Cost(C^*, \mathbf{A}),$$

for some approximation factor λ. In other words, if we find the optimal k-means clustering for our dimension reduced data, it will be close to optimal for our original data. Since k-means is rarely solved exactly, this guarantee is typically strengthened so that any *approximately* optimal clustering for $\tilde{\mathbf{A}}$ works as well. I.e. if we find some $\tilde{C}'$ such that $Cost(\tilde{C}', \tilde{\mathbf{A}}) \leq \alpha \cdot Cost(\tilde{C}^*, \tilde{\mathbf{A}})$ for some $\alpha \geq 1$, then

$$Cost(\tilde{C}', \mathbf{A}) \leq \alpha\lambda \cdot Cost(C', \mathbf{A}).$$

Achieving $\alpha = 1$ corresponds to exactly solving k-means on $\tilde{\mathbf{A}}$, recovering our original guarantee.

Several results achieve this sort of guarantee, starting with [17], which shows that projecting $\mathbf{A}$'s columns to their top k principle directions is sufficient for an approximation of $\lambda = 2$. This work was generalized in [19], which shows that taking $O(k/\epsilon^2)$ principle components is sufficient for $\lambda = 1 + \epsilon$. Bounds have also been shown for alternative dimensionality reduction methods, including feature selection [7, 8, 6], but always with an approximation of $\lambda > 2$.

Recently, we improve on the PCA results, showing that projecting to just $\lceil k/\epsilon \rceil$ principal components suffices for $\lambda = 1 + \epsilon$. We also give improved error analysis for Johnson-Lindenstrauss projections and feature selection – subspace score sampling needs to select just $O(k \log k/\epsilon^2)$ dimensions for a $\lambda = 1 + \epsilon$ approximation.

We should be careful to note that these bounds only provide an initial insight into the effectiveness of dimensionality reduction for k-means. As mentioned, k-means is often solved using a heuristic algorithm, which is vulnerable to local minima. It is not clear how dimensionality reduction effects the probability of reaching a global minimum. Objective function loss could be worse or better than predicted depending on the heuristic used.

Additionally, it is important to remember that optimizing the k-means objective is usually not the final goal of unsupervised learning – the objective is simply a coarse mathematical indicator of a partition's quality. The ultimate test for clustering is the quality of answer obtained for the problem k-means is being applied to, measured by classification rate, segmentation accuracy, etc.

Nevertheless, a concrete theoretical goal is valuable in roughly predicting the effectiveness of dimensionality reduction and, for feature selection, understanding what it means to be 'important' for clustering.

2.3. Subspace Scores

Variations on subspace scores have been studied in the theoretical computer science literature for a variety of problems including linear regression and matrix decomposition [32, 28]. For k-means clustering, we formulate them as a weighted sum of a feature's *rank-m leverage score*, ℓ, and its *residual score*, r [11]. Recall that we seek to sample columns from a data matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$. Let $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$ denote the singular value decomposition of $\mathbf{A}$. $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{d \times r}$ have orthogonal columns (the left and right singular vectors of $\mathbf{A}$) and $\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$ is a positive diagonal matrix containing the singular values of $\mathbf{A}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_d$, from top left to bottom right. If our data is centered, the right singular vectors $\mathbf{V}$ are equivalent to the *principal components* of our data. For example, if each row of $\mathbf{A}$ held pixel values for a face image, then $\mathbf{V}$'s columns would give us pixels values for the eigenfaces of $\mathbf{A}$. Now, for a chosen rank parameter m, let $\mathbf{V}^{(m)}$ be $\mathbf{V}$ with all but its first m columns (the top m principal components of $\mathbf{A}$) zero'd out.

For weighting parameter γ, the subspace score of the i^{th}

column of $\mathbf{A}$, $\mathbf{A}_i$, is denoted v_i and defined by:

$$\begin{aligned} v_i &= \ell_i + \gamma \cdot r_i \\ &= \|\mathbf{V}_i^{(m)\top}\|_2^2 + \gamma \cdot \|\mathbf{A}_i - \mathbf{A}_i \mathbf{V}^{(m)} \mathbf{V}^{(m)\top}\|_2^2. \end{aligned}$$

$\ell_i = \|\mathbf{V}_i^{(m)\top}\|_2^2$ is the rank-m leverage score of column $\mathbf{A}_i$, which measures how important the i^{th} feature is in composing the top m principal components of $\mathbf{A}$. $r_i = \|\mathbf{A}_i - \mathbf{A}_i \mathbf{V}^{(m)} \mathbf{V}^{(m)\top}\|_2^2$ is the residual score of column $\mathbf{A}_i$, which measures the variance of the i^{th} feature once the top principle components are controlled for (projected out). For provably good k-means approximation, we set $m = 2k$ and $\gamma = \frac{k}{\sum_{i=k}^{d} \sigma_i^2}$ [11]. However, in general m and γ should be considered free parameters. In Section 4.1 we give intuition for why balancing a feature's leverage score and residual score gives an ideal measure for feature selection.

Subspace scores are simple and relatively fast to compute. Algorithms for the singular value decomposition are widely available and can be accelerated when we do not require a full SVD of $\mathbf{A}$ – for subspace score computation we just need $\mathbf{A}$'s top m singular vectors. For example, we are able to quickly compute scores for all test problems using MATLAB's `svds()` function, an implementation of the Lanczos algorithm [26].

3. k-means Clustering Evaluation

Our first goal is to test subspace score feature selection for clustering, comparing to PCA based methods, which are popular in practice. We test on the USPS handwritten digit database [23], which was downloaded from [34]. This dataset contains 9298 16×16 pixel images of the ten numerical digits, with roughly equal counts of each digit. A sample set of images is included in Figure 1. We test clustering with both the original image pixels (256 features) as well as generated HOG descriptors (1764 features) obtain using MATLAB's `extractHOGFeatures` function.

Figure 1: Selected images from the USPS handwritten digit database.

For faces, we use the Extended Yale Face Database B, which contains images of 38 subjects under various position and lighting conditions [21, 27]. We only consider the face-on position for each subject and remove images with very poor illumination – i.e. average brightness much below the average for the full data set. This leaves us with 1978 images in total, approximately 50 for each subject.

All images are centered, cropped, and normalized for brightness. A sample of prepared images is included in Figure 2. Each image is 192×168 pixels, and so originally has 32256 features.

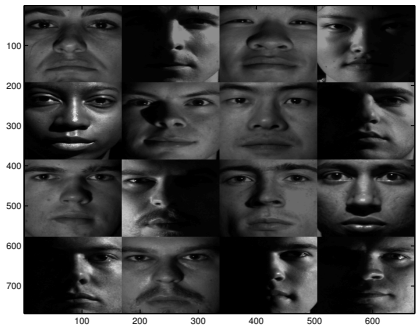


Figure 2: Selected set of centered, cropped, and normalized images from the Extended Yale Face Database B.

To compute subspace scores, we set m and γ as described in Section 2.3. We sample features (i.e. pixels) with probability proportional to their subspace scores, reweighting selected features according to the inverse of these probabilities. This ensures that the sampled image is equal to the original *in expectation*.

For comparison, PCA feature reduction is implemented by first computing the top singular vectors of our dataset and projecting each data point onto these vectors.

We use MATLAB's standard k-means implementation for clustering, which uses Lloyd's algorithm with the k-means++ initialization rule [2]. This approach can get stuck at a local minimum, but the initialization rule guarantees a solution within a $O(\log k)$ factor of optimal. Since practical implementations of exact or $(1+\epsilon)$ error approximation algorithms are not widely available, this algorithm is used nearly universally in practice. For each test, we run k-means++ with 5 different initializations and 300 iterations of Llyod's algorithm. In all tests the algorithm converged before the final iteration. For handwritten digits there are 10 natural classes so we set $k = 10$. For faces, we set $k = 38$, the number of subjects in the dataset.

3.1. Results

Our results are included in Figures 3, 4, and 5 for handwritten digits, handwritten digit HOG features, and faces. After computing a baseline k-means objective value using all data features, we test the value obtained by minimizing k-means using dimension reduced data in a variety of

smaller dimensions.

PCA feature reduction is extremely effective. For all datasets, fewer than 10 principal components give nearly optimal clusters. This result may seem surprising – in general, a $(1+\epsilon)$ approximately optimal clustering requires projection to $\lceil k/\epsilon \rceil$ principal components. However, as noted in [11], many fewer principal components are required if the principal components of our data matrix decay quickly. As shown in Figure 6, this is indeed the case for all datasets tested. Although consisting of many features, our data is "close" to low dimensional since it can be well approximated by just a few principal components.

Although it underperforms PCA, subspace score sampling is also quite effective. We see close to optimal performance after sampling approximately 100 features. While unimpressive for the standard handwritten digit dataset (which has just 256 features originally), this reduction is substantial for the HOG dataset (1764 original features) and face images (32256 original features).

The cost function for figures Figures 3, 4, and 5 is the standard k-means squared Euclidean distance cost. Using available labels, we were also able to compute a supervised objective value, penalizing points that appear in the same cluster but have different ground truth labels. PCA and subspace score sampling both performed well under this metric, indicating that these methods are effective for optimizing our underlying objective, in addition to the surrogate k-means object. Note that for handwritten digits, HOG features significantly improved performance when measured with ground truth labels. This seemed to come at the cost of a larger feature set. However, using feature reduction, we are able to eliminate this cost.

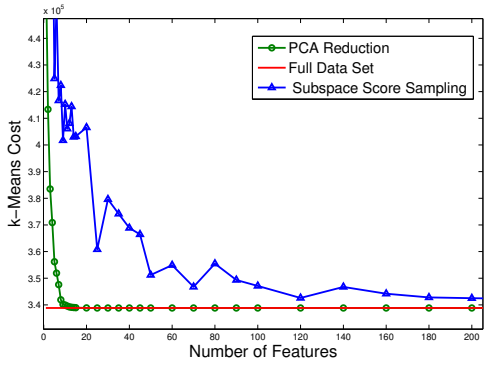


Figure 3: Effect of feature reduction on k-means clustering for handwritten digits. Original feature set: 256 pixels.

4. Subspace Scoring for Face Recognition

Since they give theoretically justified and empirically confirmed approximation results for k-means, it is natural to ask if subspace scores are more widely applicable to feature selection in machine vision. In this section, we provide theoretical and experimental evidence to suggest they are.

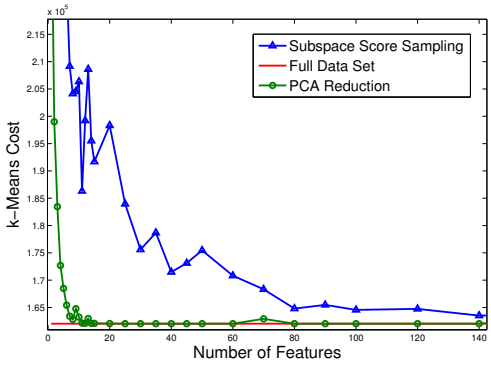


Figure 4: Effect of feature reduction on k-means clustering for handwritten digits. Original feature set: 1764 HOG features.

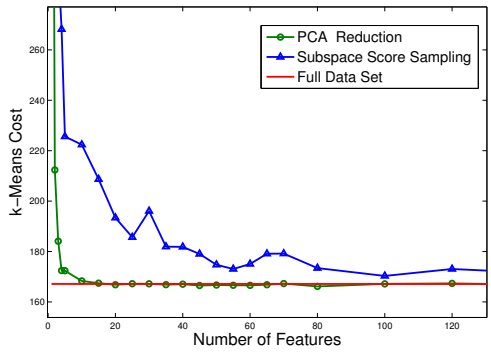


Figure 5: Effect of feature reduction on k-means clustering for face images. Original feature set: 32256 pixels.

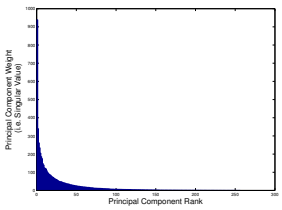
4.1. Mathematical Intuition

Recall that the subspace score for the i^{th} column (feature) of a data matrix $\mathbf{A}$ is computed as:

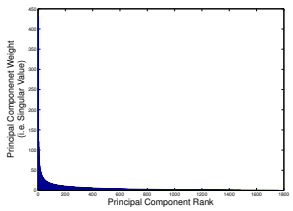
$$\begin{aligned} v_i &= \ell_i + \gamma \cdot r_i \\ &= \|\mathbf{V}_i^{(m)\top}\|_2^2 + \gamma \cdot \|\mathbf{A}_i - \mathbf{A}_i \mathbf{V}^{(m)} \mathbf{V}^{(m)\top}\|_2^2. \end{aligned}$$

The columns of $\mathbf{V}^{(m)}$ are the top m singular vectors of $\mathbf{A}$ – i.e. its top principal components for mean centered data. Thus, the i^{th} row of $\mathbf{V}^{(m)}$, $\mathbf{V}_i^{(m)\top}$ contains the value of the i^{th} feature in each of the top m principal components of $\mathbf{A}$. By computing the squared norm of this row, the i^{th} leverage score ℓ_i sums the 'importance' that a given feature plays in the top principal components of our data matrix.

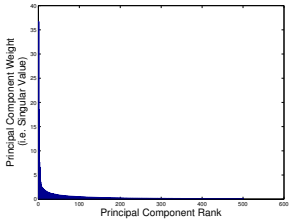
On the other hand, r_i measures the variance of a feature *outside* of the top m principal components. We begin by subtracting $\mathbf{A}_i \mathbf{V}^{(m)} \mathbf{V}^{(m)\top}$ from our feature vector $\mathbf{A}_i$, which computes the component of the i^{th} feature out-



(a) USPS digit images.



(b) USPS digit HOG features



(c) Yale face images

Figure 6: All datasets exhibit rapid spectral decay, indicating that they are well approximated by few prinicipal components. This helps explain the strong performance of PCA based feature extraction for k-means clustering.

side of the span of our top principal directions. Evaluating $\|\mathbf{A}_i - \mathbf{A}_i\mathbf{V}^{(m)}\mathbf{V}^{(m)\top}\|_2^2$ computes the variance of that component.

Generally, projecting onto the largest principal directions preserves *global patterns* in images (the rows of $\mathbf{A}$) – this largely explains why PCA is such an effective dimensionality reduction tool. The leverage score component of our subspace score ensures that we select pixels important to global variation. On the other hand, the residual score captures pixels that have high *local* variance. Even if these features do not participate in global patterns, they may be important in classification, clustering, and other tasks. γ balances the weight we place on global vs. local pixel importance, while m parameterizes the intrinsic dimensionality of our 'global variation'.

It is important to note that subspace scores go beyond computing a weighted sum of feature variance in the top principal directions and variance outside of the top directions. If this were the case, our feature score would be:

$$\|\mathbf{A}_i\mathbf{V}^{(m)}\mathbf{V}^{(m)\top}\|_2^2 + \gamma \cdot \|\mathbf{A}_i - \mathbf{A}_i\mathbf{V}^{(m)}\mathbf{V}^{(m)\top}\|_2^2$$
$$= \|\mathbf{\Sigma}\mathbf{V}_i^{(m)\top}\|_2^2 + \gamma \cdot \|\mathbf{A}_i - \mathbf{A}_i\mathbf{V}^{(m)}\mathbf{V}^{(m)\top}\|_2^2.$$

The equality follows from writing $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ via the singular value decomposition and noting that multiplying by $\mathbf{U}$ does not effect norm since it's an orthonormal matrix. If γ was set to 1, this would exactly equal $\|\mathbf{A}_i\|_2^2$, the variance of feature i.

Instead, subspace scores normalize the top principal components by eliminating $\mathbf{\Sigma}$ before computing $\ell_i = \|\mathbf{V}_i^{(m)\top}\|_2^2$. This observation leads to some additional intuition. Typically top principal directions are believed to capture true data variance, while lower principal components are suspected to be polluted by noise. If we knew our features contained only informative data, normalization would be essential. Otherwise, valuable discerning information could be washed out by very high variance in a few directions – this is especially an issue with quickly decaying principal components, like those shown in Figure 6.

On the other hand, in the presence of noise, we may not want to consider smaller principal directions, which we suspect may not be informative at all. Subspace scores balance this trade off by applying *selective normalization*. Setting m is an attempt to guess the cut off between data directions and noise directions. By not excluding everything below the cut off entirely, our r term adds robustness.

4.2. Experimental Evaluation

With intuition that subspace scores give a useful measure of feature importance, we test their effectiveness a well known application of dimensionality reduction in vision – facial recognition [37]. Typically, a set of training faces are projected onto their largest principal components, which are known as *eigenfaces*. To identify a new face, the face image is first projected onto these eigenfaces before classification – for example we might simply find the nearest neighbor of the dimension reduced vector in the training set.

Projection onto few top principal components can significantly accelerate image classification. It may also serve as an implicit denoising operation by dropping contributions from small data directions. However, our experiments do not indicate that this effect is significant – a nearest neighbor classifier that uses full face images always outperforms PCA reduced data, albeit at additional runtime cost. This observation matches results from prior work [3].

Instead of applying PCA, we can modify the eigenface method by using feature selection to choose a small subset of pixels for use in classification. At first glance, this method may seem naive, as selecting individual pixels could eliminate information present in correlations between neighboring pixels. However, given a proper importance measure, pixel selection can significantly outperform the eigenface method as well as classification using *all* image features [39]. Feature selection also avoids a projection step during classification, which is required by the eigenface method, so it helps further accelerate face recognition.

We test whether subspace scores give a good enough measure of feature importance to see these benefits.

4.3. Setup

We begin by establishing a baseline via the standard eigenface method as well full feature classification. Again using the Extended Yale Face Database B, we select 20%

of our data to serve as a training set, with the rest used for testing. We use an SVD to compute eigenfaces for the mean centered training set. Every test and training image is projected onto the top k eigenfaces before running nearest neighbor classification. Classification rates for different values of k, along with the rate obtained without dimensionality reduction are included in Figure 7.

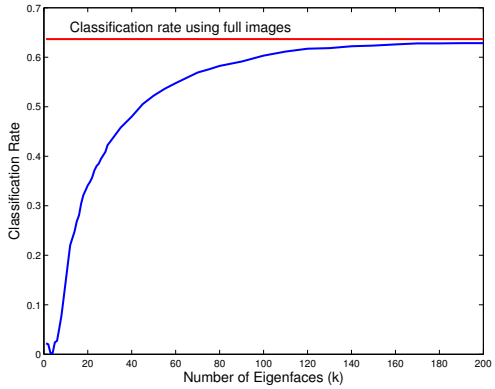


Figure 7: Classification rate using nearest neighbor classifier after projection onto top k eigenfaces.

As the number of eigenfaces increases, classification rate approaches performance for the full feature set (i.e. unmodified images), as expected. The best classification rate is not especially high – nearest neighbor classification is a simple approach and we apply it with little image preprocessing (brightness normalization, centering, and cropping). Nevertheless, this approach will be sufficient for comparing feature reduction methods.

4.4. Comparision of Feature Selection Methods

Next we compare a variety of feature selection methods, including subspace scoring, to the baseline. The tested methods are all based on scoring image pixels:

Uniform scores: All pixels are scored equally and sampled uniformly at random. This method provides a reference for other selection techniques – it seems unlikely to outperform nearest neighbor classification with complete images.

Data variance: After normalizing brightness across the training set, we set each pixel's score to its variance across the images. This is a simple, common approach to scoring.

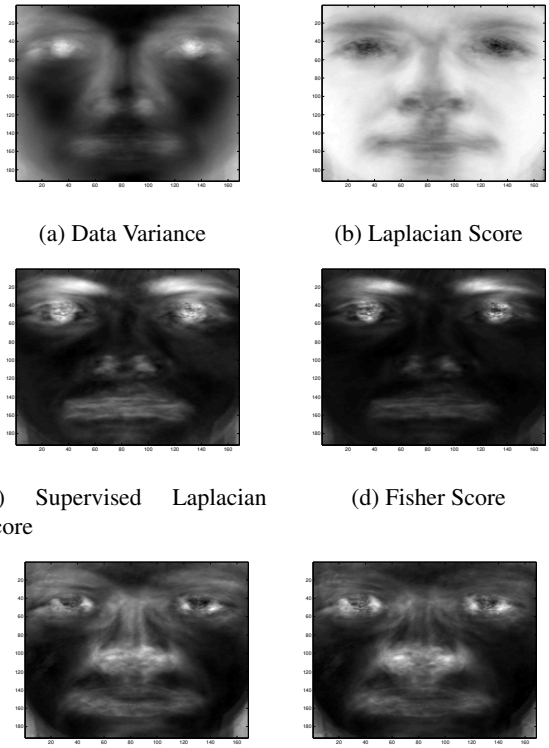
Laplacian scores [39]: A common feature selection technique based on the eigenvectors of a similarity graph for the training data. We use a 5-nearest neighbor graph based on Euclidean distance to compute scores, adapting code available at [9].

Supervised Laplacian scores: Here the similarity graph contains an edge between any two training images with the same label (i.e. are pictures of the same individual).

Fisher scores: Another common supervised technique [18]. Higher scores are given to features that have small within-class variance in comparison to their total variance.

We expect the last two feature selection methods to prove more powerful than subspace scores and other unsupervised approaches, which do not incorporate label information.

For intuition, all scores are visualized in Figure 8, with lighter pixels corresponding to higher scores. Despite being unsupervised, our subspace scores closely match the supervised Laplacian and Fisher scores, a promising observation. Interestingly, unsupervised Laplacian scores produce a very different image – pixels in important areas near the eyes and mouth are actually weighted down. Adjusting the underlying similarity matrix did not change the result substantially and, predictably, unsupervised Laplacian scores proved ineffective in our experiments.



(a) Data Variance (b) Laplacian Score

(c) Supervised Laplacian Score (d) Fisher Score

(e) Subspace Score, $m = 10$ (f) Subspace Score, $m = 40$

Figure 8: Different feature scoring methods. Lighter pixels indicate higher feature score.

4.5. Recognition Results

Face recognition results for sampling by importance score are included in Figure 9. We compute two different

sets of subspace scores, one with $m = 10$ and the other with $m = 40$. In both cases γ is set so that $\sum_i \ell_i$ is approximately equal to $\sum_i \gamma \cdot r_i$. Equalizing the total weight of our leverage scores and residual scores gave good performance in general. A more nuanced understanding of how to choose m and γ is an interesting question for future study. For both settings of m, subspace scores perform very well, outperforming the baseline method, uniform sampling, data variance, and unsupervised Laplacian scores.

In Figure 10, we also include results for selecting top scoring features instead of sampling with probability proportional to score. As more pixels are selected, performance typically increases to a point, after which it converges back to the baseline rate. For all methods, we obtained higher peak performance using pixel selection rather than sampling. This is somewhat in opposition to theoretical results, where sampling is always required, and understanding why selection is better in practice is also an interesting question.

Overall, subspace scores seem a promising choice for feature selection. They appear to capture much of the same information as supervised Fisher and Laplacian scores, without relying on labeling information. While labels were available for our application, in semi-supervised or unsupervised applications, this may not be the case.

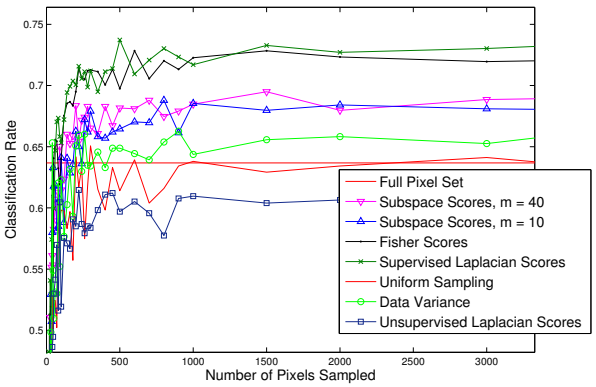


Figure 9: Classification rate for different selection methods. Pixels are sampled with rate proportional to feature score.

5. Conclusion

Our preliminary investigation justifies further investigation of subspace scores. To the best of our knowledge, these scores are the first measure of importance to successfully interpolate between the data normalization approach of leverage scores, which has been essential to work on linear regression, and standard measures of data variance.

As future work, we would like to confirm the effectiveness of subspace scoring on a much wider range of vision tasks, ideally where unsupervised selection is the only option. In doing so, we hope to gain a better understanding of how to appropriately set parameters m and γ and how sampling performs in comparison to highest score selection.

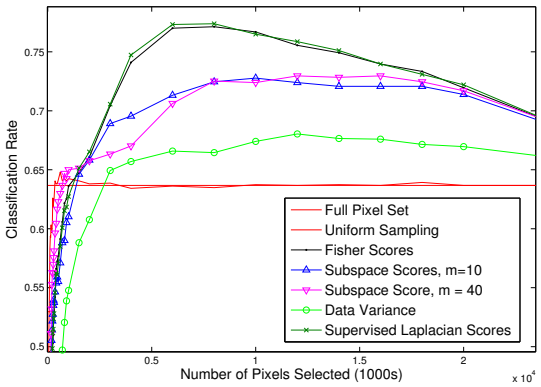


Figure 10: Classification rate for different selection methods. Pixels are chosen in order of decreasing feature score.

Finally, given the perspective described in Section 4.1, we might hope to explore alternative *selective normalization* functions for computing data variance. While the function underlying subspace scores comes directly from theoretical analysis of k-means, smoother approaches may be better for practical applications. For example, instead of normalizing all top principal components completely, a more gradual normalization factor may be preferable.

References

[1] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

[2] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

[3] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.

[4] J. Bins and B. Draper. Feature selection from huge feature sets. In *Proceedings of the 8th International Conference on Computer Vision (ICCV)*, volume 2, pages 159–165, 2001.

[5] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[6] C. Boutsidis and M. Magdon-Ismail. Deterministic feature selection for k-means clustering. *IEEE Transactions on Information Theory*, 59(9):6099–6110, 2013.

[7] C. Boutsidis, M. W. Mahoney, and P. Drineas. Unsupervised feature selection for the k-means clustering problem. In *Ad-*

vances in Neural Information Processing Systems 22 (NIPS), pages 153–161, 2009.

[8] C. Boutsidis, A. Zouzias, and P. Drineas. Random projections for k-means clustering. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 298–306, 2010.

[9] D. Cai. Matlab codes and datasets for feature learning. `http://www.cad.zju.edu.cn/home/dengcai/Data/data.html`, November 2014.

[10] M. Carreira-Perpinan. *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction*. University of Sheffield, 2001.

[11] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu. Dimensionality reduction for k-means clustering and low rank approximation. *Computing Research Repository (CoRR)*, abs/1410.6801, 2014. arXiv:1410.6801.

[12] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[13] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *Workshop on statistical learning in computer vision, ECCV*, 1(1-22):1–2, 2004.

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005.

[15] C. Ding and X. He. K-means clustering via principal component analysis. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

[16] C. Ding, X. He, H. Zha, and H. Simon. Adaptive dimension reduction for clustering high dimensional data. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM)*, pages 147–154, 2002.

[17] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004.

[18] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2 edition, 2001.

[19] D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. In *Proceedings of the 24th Annual Symposium on Discrete Algorithms (SODA)*, pages 1434–1453, 2013.

[20] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 186–193, 2003.

[21] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[22] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. *Discrete and Computational Geometry*, 37(1):3–19, 2007. Preliminary version in the 21st Annual Symposium on Computational Geometry (SCG).

[23] J. J. Hull. A database for handwritten text recognition research. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(5):550–554, 1994.

[24] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference on modern analysis and probability*, volume 26 of *Contemporary Mathematics*, pages 189–206. 1984.

[25] A. Kumar, Y. Sabharwal, and S. Sen. A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proceedings of the 45th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 454–462, 2004.

[26] C. Lanczos. An iterative method for the solution of the eigenvalue problem of linear differential and integral. *J. Res. Nat'l Bur. Std.*, 42:225–282, 1950.

[27] K.-C. Lee. The extended yale face database b. 'Cropped images' dataset at `http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html`.

[28] M. Li, G. L. Miller, and R. Peng. Iterative row sampling. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 127–136. IEEE, 2013.

[29] H. Liu and H. Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[30] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[31] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 4th IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999.

[32] M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[33] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International journal of computer vision*, 14(1):5–24, 1995.

[34] C. Rasmussen and C. Williams. Usps handwritten digit data. `http://www.gaussianprocess.org/gpml/data/`, November 2014.

[35] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 370–377, 2005.

[36] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., 1st edition, 2010.

[37] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[38] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z.-H. Zhou, M. Steinbach, D. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[39] P. N. Xiaofei He, Deng Cai. Laplacian score for feature selection. pages 507–514, 2005.

[40] H. Zha, X. He, C. H. Q. Ding, M. Gu, and H. D. Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 1057–1064, 2001.