An eigenvalue-problem formulation for non-parametric mutual information maximisation for linear dimensionality reduction

Raymond Liu[1], **Duncan F. Gillies**[1]
[1]Department of Computing, Imperial College of Science Technology and Medicine, London, United Kingdom

Abstract— *Well-known dimensionality reduction (feature extraction) techniques, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), are formulated as eigenvalue-problems, where the required features are eigenvectors of some objective matrix. Eigenvalue-problems are theoretically elegant, and have advantages over iterative algorithms. In contrast to iterative algorithms, they can discover globally optimal features in one go, thus reducing computation times and avoiding local optima. Here we propose an eigenvalue-problem formulation for linear dimensionality reduction based on maximising the mutual information between the class variable and the extracted features. Mutual information takes into account all moments of the input data while PCA and LDA only account for the first two moments. Our experiments show that our proposed method achieves better, more discriminative projections than PCA and LDA, and gives better classification results for datasets in which each class is well-represented.*

Keywords: Feature extraction, dimensionality reduction, mutual information, eigenvalue-problem, pattern recognition.

Acknowledgement

This research is supported by the Departmental Teaching Award of Department of Computing, Imperial College London. This award is funded by EPSRC, UK.

1. Introduction

Feature extraction, also known as dimensionality reduction, has become a standard pre-processing step in classification and regression tasks where the data has high input dimensionality or has irrelevant input features (variables). An example of such tasks is face recognition where the input dimensionality is equal to the number of pixels per image. The number of variables is therefore large and many of them give no discriminating information. It is now well-known that high-dimensional data suffer from the curse of dimensionality ([1] section 1.4).

Principal Component Analysis (PCA, [1] section 12.1) and Linear Discriminant Analysis (LDA, [1] section 4.1.4, also known as Fisher's Discriminant Analysis — FDA) are two of the most popular feature extraction techniques. One limitation of LDA is that it involves inverting the within-class covariance matrix which is often singular for high dimensional data, but this problem has been overcome by PCA+LDA ([2]). The most noteworthy limitation of both PCA and LDA in our investigation however, is that they both only consider the first two moments of the training data. For example, LDA implicitly assumes that each class in the training data has a unimodal Gaussian distribution, and therefore performs badly for data where the classes don't have this distribution.

In this paper we introduce a new feature extraction algorithm based on mutual information, which accounts for all moments of the training data. As we will see, there have been recent developments in information-theoretic feature extraction ([3], [4], [5], [6]), but all the methodologies introduced here are based on an iterative approach, and thus do not share the theoretical elegance of PCA and LDA. Moreover, many iterative methods cannot guarantee that globally optimal features will be found, and the number of iterations in the optimisation could potentially be large. Our proposed method in this paper is formulated as an eigenvalue-problem, just like PCA and LDA, and so shares all the advantages therein.

2. Related work

In [6], Bollacker and Ghosh present an early, speculative investigation of information-theoretic feature extraction. Two methodologies were introduced: one is an iterative approach and the other is an eigenvalue-problem formulation but which has no theoretical justification. Torkkola's iterative approach in [3] makes use of an alternative form of entropy and mutual information proposed by Renyi ([7], [3]). A particular form of Renyi's mutual information is the quadratic mutual information (QMI), which has very convenient theoretical properties with Gaussian mixture distributions, as we will see. Since then there has been a string of methodologies that make use of these properties of QMI. In [5], the authors apply QMI to variational graph embedding, and formulates the problem of feature extraction as one of optimisation which can be solved by an EM-style algorithm ([1] chapter 9). In [4], the authors use an alternative criterion to mutual information, called Info-Margin, but which is based on Renyi's quadratic entropy and mutual information. The algorithm is again iterative.

A successful feature *selection* (as opposed to *extraction*, [8]) algorithm based on an approximation of Shannon mutual information is introduced in [9]

3. Approximate quadratic mutual information discriminant analysis

The QMI (quadratic mutual information) ([7], [3]) between two continuous random variables X and Y is given by

$$I_Q(X;Y) := \int_X \int_Y [p_{X,Y}(x,y) - p_X(x)p_Y(y)]^2 dxdy.$$

If one of the random variables is discrete, then we replace the corresponding integral with a sum.

Now let us consider extracting a single feature, represented by the unit vector $\mathbf{w}$, from the D-dimensional data input space $\mathcal{S} = \mathbb{R}^D$. Let $\mathbf{X}$ be a random vector variable in $\mathcal{S}$, and write $Y := \mathbf{w}^{\mathrm{T}}\mathbf{X}$. Thus Y is the projection of $\mathbf{X}$ in the direction $\mathbf{w}$. Similarly, we write $y = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ for realisations $\mathbf{x}$ and y of random variables $\mathbf{X}$ and Y respectively, and $y_n = \mathbf{w}^{\mathrm{T}}\mathbf{x}_n$ for each of the input training data points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathcal{S}$. We want to maximise the quadratic mutual information between the extracted feature and the class variable. That is, we want to maximise

$$I_Q(Y;C) = \sum_{c=1}^{K} \int_{\mathbb{R}} [p_{Y|C}(y|c)p_C(c) - p_Y(y)p_C(c)]^2 dy, \tag{1}$$

where we note that $p_{Y,C}(y,c) = p_{Y|C}(y|c)p_C(c)$. K is the total number of classes. Here we use the class prior probabilities $p_C(c) := \frac{N_c}{N}$ where each N_c is the nunber of training examples in class c, and N is the total number of training examples. Let Ω_c denote the index set of class c, thus $N_c = |\Omega_c|$. We use kernel density estimation with Gaussian kernels to estimate $p_Y(y)$ and $p_{Y|C}(y|c)$ ([10], [3]).

$$p_Y(y) = \frac{1}{N}\sum_{n=1}^{N} G(y - y_n, \frac{\sigma^2}{2}),$$
$$p_{Y|C}(y|c) = \frac{1}{N}\sum_{n\in\Omega_c} G(y - y_n, \frac{\sigma^2}{2}),$$

where $G(\cdot,\cdot)$ is the Gaussian kernel defined by

$$G(x,h) := \frac{1}{\sqrt{2\pi h}} \exp\Big(-\frac{x^2}{2h}\Big).$$

The factor of $\frac{1}{2}$ as in $\frac{\sigma^2}{2}$ above is just for notational convenience as we will see. The combination of using quadratic mutual information and Gaussian kernel density estimation provides a beautifully elegant analytical solution for the integral in Equation (1), thanks to the convenient property of Gaussian densities that the convolution of two Gaussians is another Gaussian.

$$\int_{\mathbb{R}}\int_{\mathbb{R}} G(y-\mu, \frac{\sigma^2}{2})G(z-\lambda, \frac{\sigma^2}{2})\mathrm{d}y\mathrm{d}z = G(\mu-\lambda, \sigma^2).$$

Using this, and multiplying out the expression for $I_Q(Y;C)$ above, substituting for the probability density functions, we obtain

$$I_Q(Y;C) = \frac{1}{N^2}\Big[\sum_{c=1}^{K}\sum_{n\in\Omega_c}\sum_{m\in\Omega_c} G_{nm} + \Big(\sum_{c=1}^{K}\frac{N_c^2}{N^2}\Big)\sum_{n=1}^{N}\sum_{m=1}^{N} G_{nm} - 2\sum_{c=1}^{K}\frac{N_c}{N}\sum_{n\in\Omega_c}\sum_{m=1}^{N} G_{nm}\Big],$$

where we have used the short-hand notation $G_{nm} := G(y_n - y_m, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\Big(\frac{(y_n-y_m)^2}{2\sigma^2}\Big)$. Although the above sums look like triple-sums, they're actually double-sums over the training examples because of class indexing, and so are $\mathcal{O}(N^2)$. Recall that $y_n = \mathbf{w}^{\mathrm{T}}\mathbf{x}_n$ where $\mathbf{w}$ is the feature (projection) that we want to extract (that maximises its mutual information with the class variable). Note that every G_{nm} in the above expression involves $\mathbf{w}$, and we are trying to maximise the above in $\mathbf{w}$. But we do not want to perform the maximisation iteratively, we want to formulate this as an eigenvalue problem instead. Each G_{nm} can be written, in terms of the $\mathbf{x}_n$ and $\mathbf{w}$, as

$$G_{nm} = \frac{1}{\sigma\sqrt{2\pi}}\exp\Big[-\frac{\mathbf{w}^{\mathrm{T}}(\mathbf{x}_n - \mathbf{x}_m)(\mathbf{x}_n - \mathbf{x}_m)^{\mathrm{T}}\mathbf{w}}{2\sigma^2}\Big]. \tag{2}$$

Now here is the key point. We approximate G_{nm} above by $\mathbf{w}^{\mathrm{T}}E_{nm}\mathbf{w}$, where each E_{nm} is defined by

$$E_{nm} := \frac{1}{\sigma\sqrt{2\pi}}\exp\Big[-\frac{(\mathbf{x}_n - \mathbf{x}_m)(\mathbf{x}_n - \mathbf{x}_m)^{\mathrm{T}}}{2\sigma^2}\Big]. \tag{3}$$

Note that E_{nm} is a matrix exponential (and therefore a matrix itself), because each $(\mathbf{x}_n - \mathbf{x}_m)(\mathbf{x}_n - \mathbf{x}_m)^{\mathrm{T}}$ is a matrix. Note also that the essential difference between G_{nm} and $\mathbf{w}^{\mathrm{T}}E_{nm}\mathbf{w}$ is that the $\mathbf{w}^{\mathrm{T}}$ and $\mathbf{w}$ have been moved outside the exponential. We will see later that this approximation is a very close one, and more importantly that for any $\mathbf{w}$, each $\mathbf{w}^{\mathrm{T}}E_{nm}\mathbf{w}$ is a tight upper bound of G_{nm}, and for some suitable constant C, each $(\mathbf{w}^{\mathrm{T}}E_{nm}\mathbf{w} - C)$ is a tight lower bound for G_{nm}. This justifies the soundness of maximising or minimising the approximation $\tilde{I}_Q(Y;C)$ to the QMI $I_Q(Y;C)$, where $\tilde{I}_Q(Y;C)$ is obtained from $I_Q(Y;C)$ by replacing every G_{nm} by $\mathbf{w}^{\mathrm{T}}E_{nm}\mathbf{w}$. For now we call this method *Approximate mutual information discriminant analysis* (AQMIDA).

Now if we define the matrix E by

$$E := \frac{1}{N^2}\Big[\sum_{c=1}^{K}\sum_{n\in\Omega_c}\sum_{m\in\Omega_c} E_{nm} + \Big(\sum_{c=1}^{K}\frac{N_c^2}{N^2}\Big)\sum_{n=1}^{N}\sum_{m=1}^{N} E_{nm} - 2\sum_{c=1}^{K}\frac{N_c}{N}\sum_{n\in\Omega_c}\sum_{m=1}^{N} E_{nm}\Big], \tag{4}$$

we see that our approximation $\tilde{I}_Q(Y;C)$ is simply given by

$$\tilde{I}_Q(Y;C) = \mathbf{w}^{\mathrm{T}}E\mathbf{w}, \tag{5}$$

and we wish to maximise this in $\mathbf{w}$. Finally we recall from elementary linear algebra that the maximising $\mathbf{w}$ is given by the largest eigenvector (the eigenvector with the largest eigenvalue) of the matrix E. If we want to extract more than one feature, say M features where $M > 1$, then we simply use the M largest eigenvectors of E. We have now formulated the problem of feature extraction based on QMI as an eigenvalue problem, that is to say, the same formulation as PCA and LDA, but using the criterion of maximum mutual information.

We still need to estimate σ^2, our bandwidth parameter in the Gaussian kernels. The variance of the data along different directions in the feature space will in general be different, but we circumvent this problem by whitening the data a priori using PCA. That is, before performing our feature extraction algorithm, we project the input data into normalised PCA space, after which the data will have the same sample variance (equal to 1) in all directions. For this paper our bandwidth parameter σ^2 is estimated by Silverman's rule of thumb ([11]). $\sigma = \left(\frac{4\hat{\sigma}^5}{3N}\right)^{\frac{1}{5}}$, where $\hat{\sigma}$ is the sample standard deviation. Exactly how the bandwidth affects the quality of the extracted features is a subject of future research.

3.1 Closeness of approximation

Recall that our central idea is to approximate the actual QMI $I_Q(Y;C)$ (Equation (1)) by the approximate QMI $\tilde{I}_Q(Y;C)$ (Equations (5) and (4)). $I_Q(Y;C)$ and $\tilde{I}_Q(Y;C)$ only differ in their summands because the structure of the sum is the same. Each summand in $I_Q(Y;C)$ is G_{nm}, and from Equation (2) we can see that by letting

$$\lambda := \big[\frac{\mathbf{w}^{\mathrm{T}}(\mathbf{x}_n - \mathbf{x}_m)}{\|\mathbf{x}_n - \mathbf{x}_m\|}\big]^2, \tag{6}$$

we can write

$$G_{nm} = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{\|\mathbf{x}_n - \mathbf{x}_m\|^2 \lambda}{2\sigma^2}). \tag{7}$$

Using Equation (3) and the definition (Taylor expansion) of the matrix exponential, we can easily show that each E_{nm} can be written as

$$E_{nm} = \frac{1}{\sigma\sqrt{2\pi}}\big[I - \frac{(1 - e^{-\frac{\|\mathbf{x}_n - \mathbf{x}_m\|^2}{2\sigma^2}})}{\|\mathbf{x}_n - \mathbf{x}_m\|^2}(\mathbf{x}_n - \mathbf{x}_m)(\mathbf{x}_n - \mathbf{x}_m)^{\mathrm{T}}\big],$$

where I is the identity matrix, thus each summand $\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w}$ of $\tilde{I}_Q(Y;C)$ can be written as

$$\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w} = \frac{1}{\sigma\sqrt{2\pi}}[1 - (1 - e^{-\frac{\|\mathbf{x}_n - \mathbf{x}_m\|^2}{2\sigma^2}})\lambda]. \tag{8}$$

Notice that by Equation (6) we have $0 \leq \lambda \leq 1$, and that $\sqrt{\lambda}$ is the projection of $\mathbf{w}$ in the direction of $\frac{\mathbf{x}_n - \mathbf{x}_m}{\|\mathbf{x}_n - \mathbf{x}_m\|}$.

Figure 1 illustrates the approximation of G_{nm} by $\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w}$. Here we regard $\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w}$ and G_{nm} as functions of λ (Equations (7) and (8)). Recall that we want to maximise $I_Q(Y;C)$ with respect to $\mathbf{w}$, and by Equation (6), λ is a quadratic function of $\mathbf{w}$. We can see from Figure 1 that it is a close approximation, and that each $\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w}$ is a tight upper bound of each G_{nm}. Indeed, $\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w}$ and G_{nm} agree when $\lambda = 0$ or $\lambda = 1$. Furthermore, we can see clearly that $(\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w} - 1)$ is a lower bound for G_{nm}, and that for some constant $C < 1$, $(\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w} - C)$ is a tight lower bound. Therefore, since the approximation $\tilde{I}_Q(Y;C)$ to the actual QMI $I_Q(Y;C)$ is a summand-wise approximation $\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w}$ of G_{nm}, we see that maximising $\tilde{I}_Q(Y;C)$ is equivalent to maximising a (close) lower bound for $I_Q(Y;C)$. This is the theoretical justification for our eigenvalue-problem approach.

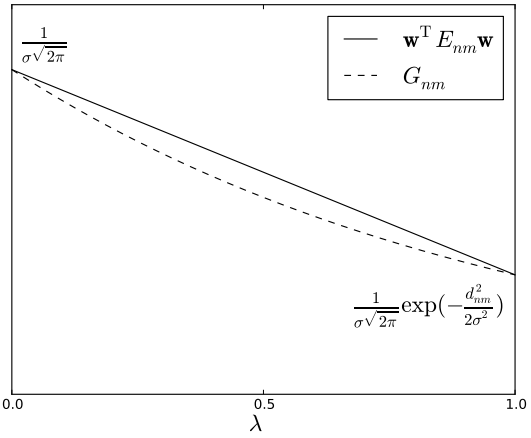


Fig. 1: Approximation of each G_{nm} by each $\mathbf{w}^{\mathrm{T}} E_{nm} \mathbf{w}$. Here we regard λ as the variable (abcissa) on the horizontal axis.

3.2 Complexity

Recall that N is the total number of training examples and D is the input dimensionality. We show that for $D > N$ (high-dimensional data), the complexity of our algorithm is of the same order as PCA.

Let X denote the $N \times D$ design matrix, whose nth row is the nth training example $\mathbf{x}_n^{\mathrm{T}}$. Without loss of generality, assume that the training examples are centered, that is,

$$\sum_{n=1}^{N} \mathbf{x}_n = \mathbf{0}.$$

PCA seeks to find eigenvectors of the sample covariance matrix $\frac{1}{N} X^{\mathrm{T}} X$. For high-dimensional data ($D > N$), it is more efficient to solve the equivalent problem of finding the eigenvectors of the matrix $\frac{1}{N} X X^{\mathrm{T}}$ ([1] Section 12.1.4). Construction of the matrix takes $\mathcal{O}(N^2 D)$ time, while the eigenvalue-problem itself takes $\mathcal{O}(N^3)$ time, so altogether the complexity of PCA, assuming $N < D$, is $\mathcal{O}(N^2 D + N^3) = \mathcal{O}(N^2 D)$. Similarly we can show that if $N > D$, the complexity of PCA is $\mathcal{O}(N D^2)$.

Our algorithm requires PCA as a pre-processing step, after which the training examples will have dimension at most $(N-1)$. We will use the same notation $\mathbf{x}_n$ for the nth data point in the PCA space, on the understanding that it now has dimensionality $(N-1)$ and no longer D. We seek to find eigenvectors of the matrix E as defined in Equation (4). Let c_n be the true class label of training example $\mathbf{x}_n$, and let $\mathbb{I}[\cdot]$ be the indicator function. We can rearrange Equation (4) so that it takes the following form.

$$E = \frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \rho_{nm} (\mathbf{x}_n - \mathbf{x}_m)(\mathbf{x}_n - \mathbf{x}_m)^{\mathrm{T}}, \quad (9)$$

where

$$\rho_{nm} = \Big(\frac{N_{c_n} + N_{c_m}}{N} - \sum_{c=1}^{K} \frac{N_c^2}{N^2} - \mathbb{I}[c_n = c_m] \Big) \tau_{nm},$$

where we use the short-hand notation

$$\tau_{nm} := \frac{1 - e^{-\frac{\|\mathbf{x}_n - \mathbf{x}_m\|^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}\|\mathbf{x}_n - \mathbf{x}_m\|^2}.$$

The precise form of ρ_{nm} (and τ_{nm}) does not matter here. Just note that ρ_{nm} is symmetric in n and m, and that it can be computed in $\mathcal{O}(N)$ time because $\|\mathbf{x}_n - \mathbf{x}_m\|^2$ can be computed in $\mathcal{O}(N)$ time (because $\mathbf{x}_n$ and $\mathbf{x}_m$ are $(N-1)$-dimensional vectors). Note further that using the symmetry of ρ_{nm} and Equation (9), we have

$$\begin{aligned} E &= \frac{2}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \rho_{nm} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} - \frac{2}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \rho_{nm} \mathbf{x}_n \mathbf{x}_m^{\mathrm{T}} \\ &= \frac{2}{N^2} \sum_{n=1}^{N} \Big(\sum_{m=1}^{N} \rho_{nm} \Big) \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} - \frac{2}{N^2} \sum_{n=1}^{N} \mathbf{x}_n \Big(\sum_{m=1}^{N} \rho_{nm} \mathbf{x}_m^{\mathrm{T}} \Big). \end{aligned}$$

We can now see, with the help of the parentheses above, that the matrix E can be computed in $\mathcal{O}(N^3)$ time. We leave it to the reader to check that this is true, noting that each $\mathbf{x}_n$ lives in the PCA space and so is (at most) $(N-1)$-dimensional.

The complexity of the eigenvalue-problem itself is again $\mathcal{O}(N^3)$ because E is an (at most) $(N-1) \times (N-1)$ matrix. Therefore, the total complexity of our algorithm, assuming $D > N$, is $\mathcal{O}(N^2 D + N^3 + N^3) = \mathcal{O}(N^2 D)$, which is the same as that of PCA.

Now if $D < N$, then a similar analysis shows that the complexity of our algorithm is $\mathcal{O}(N^2 D + N D^2)$, which is slightly higher than that of PCA (which is $\mathcal{O}(N D^2)$) due to the N^2 term. However there is no trivial way of eliminating or reducing the N^2 term because it comes from having to compute the double-sum over the training set. As Torkkola claimed in [3], "Since a kernel density estimate results in a sum of kernels over samples, any kind of divergence measure between two densities necessarily requires $\mathcal{O}(N^2)$ operations."

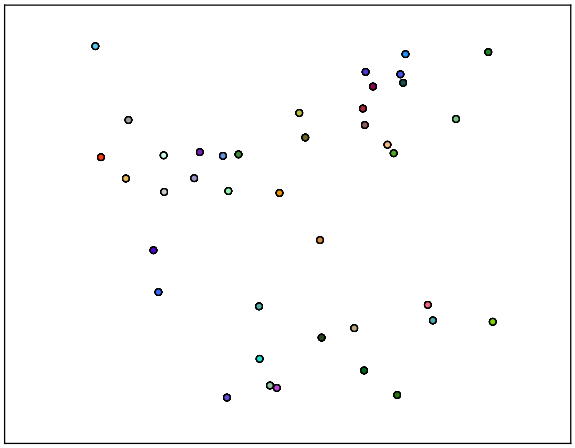
4. Experiments and evaluation

We will first illustrate the performance of our proposed method with 2D projections of the AT&T Database of Faces (formerly the ORL Database of Faces), and evaluate our method in terms of cross-validation classification accuracy using the Letter dataset which can be obtained from the UCI Machine Learning Repository. Both evaluations will be done in comparison to PCA and LDA, since our method has the same problem formulation.

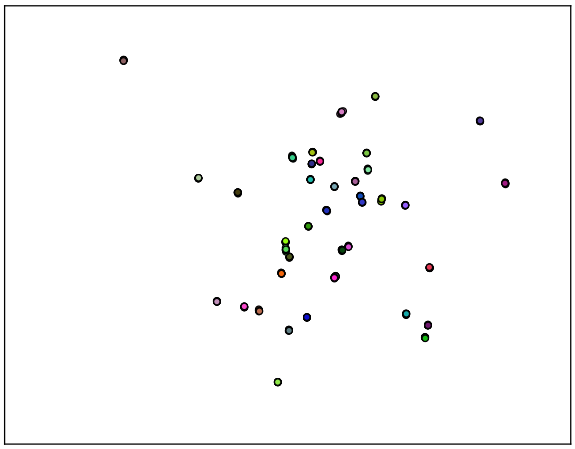
For convenience we will name our proposed algorithm *AQMIDA*. Recall that here we write LDA to mean PCA+LDA.

The AT&T Database of Faces contains 400 training examples in total, with 40 classes (individuals) and 10 training examples for each class. Figure 2 shows the 2D projections of the face dataset for AQMIDA, LDA, and PCA. We can see that unsurprisingly, the LDA projection is much better than the PCA projection, because the examples in the same class are much closer together and we can clearly see the class separation. However, the AQMIDA projection is even better than the LDA projection (more discriminating), exhibiting better class separation and more compact within-class clusters. In fact, in Figure 2a, each class (of the 40 classes) looks like a singleton point because of the resolution of the image, when in fact it is actually 10 data points that are very close together. Figures 3a and 3b elucidate the projections more clearly. They are zoomed-in versions of the projections of LDA and AQMIDA respectively, where the resolution of both canvases is the same, and they show a typical class of 10 data points in the projection. We can see that the class is spread out in the LDA projection, but still looks like a singleton point in the AQMIDA projection. For a comparison of the quality of projection of AQMIDA with two other information-theoretic feature extraction methods, the reader is invited to study Figure 6 of [4]. This shows 2D projections of the same dataset but using the Info-margin technique of Qiu and Wu ([4]) and Torkkola's method ([3]). A comparison of our Figure 2a and Figure 6 of [4] shows the superiority of the quality of the AQMIDA projection over the other methods. Moreover, AQMIDA is an eigenvalue-problem which requires no iteration.

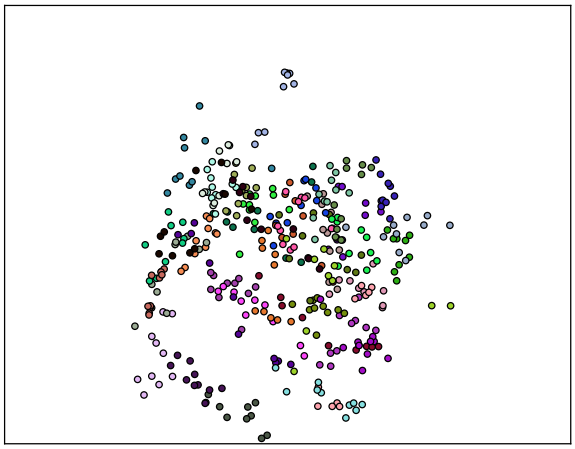
The Letter dataset contains 20,000 examples with 26 classes, and the input dimensionality is 16. The classes are roughly evenly distributed. Because of the large sample size, we did our experiments on a randomly selected 800 examples, where the 26 classes are represented as evenly as possible. 5-fold cross-validation was carried out on the 800 randomly selected examples, and the whole process was repeated 10 times. Figure 4 shows the error rates of AQMIDA (solid line), LDA (dashed line), and PCA (dash-dotted line). The horizontal axis is the number of extracted features (the dimensionality of the feature space), from 1 to 15. It is clear from Figure 4 that AQMIDA gives better classification accuracy than LDA, which in turn gives



(a) AQMIDA

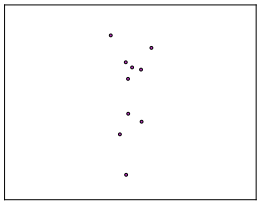


(b) LDA

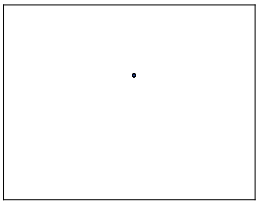


(c) PCA

Fig. 2: 2D projections of the AT&T Face dataset.



(a) LDA



(b) AQMIDA

Fig. 3: 2D projections of the AT&T Face dataset, zoomed-in on a typical class. The resolution of the canvases is the same.

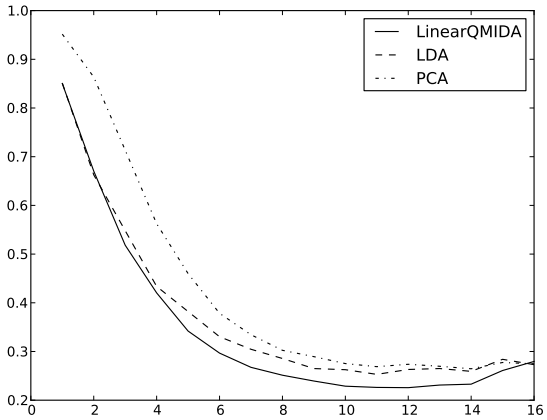


Fig. 4: Error rates for the Letter dataset

better accuracy than PCA. Perhaps this is because AQMIDA accounts for all moments of the input data and therefore is suitable for data whose class distributions are non-Gaussian or even multi-modal, whereas LDA and PCA only account for the first two moments of the input data and therefore are only applicable to unimodal Gaussian data. More precise explanations are a subject of further research.

5. Conclusion and further research

In this paper we introduced a new information-theoretic feature extraction method, AQMIDA, that is formulated as an eigenvalue problem just like LDA and PCA. An eigenvalue-problem approach, in contrast with iterative algorithms, has the following advantages. It is theoretically elegant; it does not require iteration, and therefore avoids potentially high computational cost and obliviates the issue of convergence of solutions; and finally, globally optimal features are guaranteed, in contrast with iterative algorithms that might get stuck in local optima. We have demonstrated the quality of features extracted by AQMIDA via visual projections and classification accuracy using real-world data, where we have shown that AQMIDA gives superior, more discriminative projections, and lends itself to better classification accuracy.

5.1 Further research

The idea of AQMIDA is a novel one and so there is ample room for further research. First of all, it will be informative to investigate the impact of bandwidth estimation on the quality of the extracted features. In particular, how sensitive the extracted features are to a perturbation in the bandwidth.

From a more theoretical point of view, our eigenvalue-problem formulation of AQMIDA made it necessary to make an approximation to the actual QMI between a feature and the class variable. Although the approximation is close, it would provide remarkable theoretical insight into the area of feature extraction (dimensionality reduction) and data visualisation to investigate whether there is a way of formulating an eigenvalue-problem that maximises the actual QMI and not an approximation. If this is possible, then we would have a theoretically globally optimal solution (due to the nature of eigenvalue-problems) to the problem of maximising QMI, and we would be able to investigate more profound properties of information-theoretic feature extraction.

One undesirable property of information-theoretic feature extraction is over-fitting. It is known that LDA can over-fit a given dataset, and achieve worse classification performance than PCA, when the training set is small and is not sufficiently representative of each class ([12]). Current empirical studies show that information-theoretic feature extraction, including AQMIDA, can too suffer from over-fitting, and occasionally more severely than LDA. More interestingly, it seems that AQMIDA suffers from over-fitting exactly when LDA does, and more severely than LDA, and unsurprisingly gives better, more discriminative projections than LDA. So, when LDA out-performs PCA in terms of classification accuracy, so does AQMIDA out-perform LDA (Figure 4); when LDA gives lower classification accuracy than PCA but obtains more discriminative projections than PCA, so does AQMIDA give lower classification accuracy than LDA but obtains more discriminative projections than LDA. More experiments need to be done to gain deeper insight and make more reliable conclusions. Intuitively we would expect that adjusting the bandwidth parameter (mentioned above) could alleviate over-fitting. The dynamics and subtleties herein make an interesting area of future research.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[2] J. Yang and J.-y. Yang, "Why can LDA be performed in PCA transformed space?" *Pattern Recognition*, vol. 36, no. 2, pp. 563–566, Feb. 2003.

[3] K. Torkkola, "Feature extraction by non parametric mutual information maximization," *J. Mach. Learn. Res.*, vol. 3, pp. 1415–1438, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944981

[4] X. Qiu and L. Wu, "Info-margin maximization for feature extraction," *Pattern Recognition Letters*, vol. 30, no. 16, pp. 1516 – 1522, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865509002141

[5] S.-H. Yang, H. Zha, S. Zhou, and B.-G. Hu, "Variational graph embedding for globally and locally consistent feature extraction," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009.

[6] K. Bollacker and J. Ghosh, "Linear feature extractors based on mutual information," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 2, aug 1996, pp. 720 –724 vol.2.

[7] A. Renyi, "On measures of entropy and information," vol. 1. University of California Press, 1961, pp. 547–561.

[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, March 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=944919.944968

[9] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1226–1238, 2005.

[10] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. pp. 1065–1076, 1962. [Online]. Available: http://www.jstor.org/stable/2237880

[11] B. W. Silverman, *Density estimation for statistics and data analysis*. Chapman and Hall, 1998.

[12] A. Martinez and A. Kak, "Pca versus lda," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 2, pp. 228 –233, feb 2001.