Overfitting in linear feature extraction for classification of high-dimensional image data

Raymond Liu\textsuperscript{a,}\textsuperscript{*}, Duncan F. Gillies\textsuperscript{a}

\textsuperscript{a}Department of Computing, Imperial College London, London, UK

Abstract

Overfitting has been widely studied in the context of classification and regression. In this paper, we study the overfitting in the context of dimensionality reduction. We show that the conventional wisdom of improving classification performance by maximising inter-class discrimination is not valid for high-dimensional datasets, and can lead to severe overfitting. In particular, we prove the theoretical existence of perfectly discriminative subspace projections, and show that for datasets with very high input dimensionality, inter-class discrimination should be reduced rather than maximised. This naturally leads to a simple dimensionality reduction technique, which we call Soft Discriminant Maps, which we use to show a direct relationship between the classification performance and the level of inter-class discrimination of feature extractors. Moreover, Soft Discriminant Maps consistently exhibit better classification performance than other comparable techniques.

Keywords: Dimensionality reduction, feature extraction, classification, high-dimensional datasets, overfitting

1. Introduction

In the amply active research area of dimensionality reduction, and in particular feature extraction \cite{1}, we aim to find feature subspaces whose dimension-
Figure 1: 2D projections of the AT&T Dataset of Faces using LDA. (a) Projection computed using the whole dataset. (b) The whole dataset is projected onto the 2D feature space computed using just 80% of the data, where the rest 20% of the data points are marked by grey circles.

alities are typically much smaller than that of the input spaces, and in which the data is efficiently represented and well-represented. The (loosely defined) term efficiently represented means that the dimensionality of the data points is sufficiently small so that classification, regression or other manipulation of the data can be done quickly. For example, a face detection and recognition system on a hand-held device such as a smartphone or a camera needs to be able to execute in a matter of milliseconds or even nanoseconds. Of more interest in this paper is the (again loosely defined) term well-represented. It means that little or no useful information is lost during the dimensionality reduction process, and redundant information is pruned out. It is well known that too much redundant information can severely decrease classification accuracy, and this effect is called the peaking phenomenon [1, 2].

1.1. Overfitting and the curse of dimensionality in the context of dimensionality reduction

It would be reasonable to hold the intuition that the more information a learning machine has, the more accurate it would be at making predictions. However the curse of dimensionality [3, 4] teaches us that this view is too sim-
ple. Information comes in different forms, and in the context of the curse of dimensionality we focus on two such forms. The first is information about each data point, and mathematically this translates to input dimensionality. For example a million-pixel image carries more information than a 256-pixel thumbnail. The second is information about the underlying distribution, and this is given by the abundance of training data points. For example, a dataset with 100 data points per class is more informative than a dataset with only 10 data points per class. We will call the former specific information, and the latter global information. One interpretation of the curse of dimensionality is this: too much specific information is bad, and the more global information the better.

Overfitting is closely related to the curse of dimensionality and the problem of small sample size \[5, 6\]. It occurs not just when the input dimensionality is high, but when it is high compared to the number of training data points, also called the sample size. We will illustrate this with a real-world example. Here for the sake of illustration, we use linear discriminant analysis (LDA, also known as Fisher’s discriminant) \[2, \text{sec.4.1.4}\] as our feature extraction algorithm. LDA seeks features along which the variance of each class is minimised but the distances between classes are maximised. Consider the AT&T dataset of faces, formerly the ORL face dataset. It has 40 classes, 10 data points per class, and an input dimensionality of 10,304. More concretely, this means that there are 40 persons, 10 (greyscale) face images per person, and each face image has 10,304 pixels. We will extract 2 features from the 10,304 input features using LDA, and project the data onto them. In figure 1a we trained LDA using the whole data set, and projected the whole dataset onto its 2 best features. We can see the inter-class discrimination very clearly. In fact, each class is so compact that it appears as a singleton point at this image resolution, and well-separated from the other classes. In this sense, the LDA features fit this data very well. In fact too well, as we see in figure 1b. In figure 1b we split the data into an 80% training set and a 20% testing set. LDA is trained using just the training set to produce 2 features, but both the training set and the testing set are
projected onto the features. The projected testing data points are marked with grey circles, and the projected training data points are without grey circles. Each class has 8 training data points that are tightly packed together much like figure [4a] but also 2 testing data points that are far from the 8 training data points. Note again that the 8 training data points in each class are so tightly packed together that they appear as a single data point at this resolution. See for example the class on top of figure [4b] marked by green stars. This illustrates that the feature extractor, LDA in this case, overfits the training data, and fits the testing data poorly. Indeed, if overfitting did not occur, the 2 testing data points in each class (marked with grey circles) would be close to their corresponding 8 training data points. In this sense, the feature extractor is said to give poor generalisation performance. For many feature extractors, the higher the input dimensionality of a dataset, the more likely this kind of overfitting will occur. This is because the search space, in which the optimal feature lives, increases exponentially, from $\mathbb{R}^2$ to $\mathbb{R}^3$ to $\ldots$ to $\mathbb{R}^D$, $D \gg 1$. Intuitively, we can imagine that when the input dimensionality is large, there are ‘too many features to choose from’ and while there is only one optimal feature, most ‘others’ will lead to severe overfitting due to small sample size.

Overfitting can be alleviated if we have an abundance of training data points that are highly representative of the true data distribution. However, for many real-world applications, this can be unrealistic in practice. The AT&T dataset of faces, for example, has 400 training data points, which is quite typical of a real-world scenario. Jain, Duin and Mao [1, sec.3] suggested that as a rough guideline, we should have at least 10 times as many training data points as input dimensions. Now given that the input dimensionality of the AT&T dataset of faces (for example) is 10,304, such abundance of training data points is unrealistic for real-world applications. Hence the need for dimensionality reduction.

1.2. Our contributions and related work

It is important to distinguish between two contexts in which overfitting occurs: classification on the one hand, and dimensionality reduction on the other.
This paper focuses on the latter. Trunk [7] and Bishop [2, sec.1.4] provide a mathematical example and intuitive, visual illustrations of the curse of dimensionality. These illustrations are taken from the context of classification. A more detailed discussion of the curse of dimensionality with respect to limited sample size can be found in [5, 3, 4].

[8, 9] give general and intuitive introductions to the problem of overfitting. Specific examples of overfitting in machine learning can be found in [10, 11, 12]. These perspectives of overfitting are in the context of classification and regression, and this paper studies overfitting in dimensionality reduction, and in particular linear feature extraction. [13] discusses overfitting in Fisher discriminant analysis and proposes an semisupervised extension to alleviate overfitting. In this paper we explore more generally the theoretical reasons and mechanisms behind overfitting in high-dimensional feature extraction, and propose a new feature extraction technique, Soft Discriminant Maps (SDM), as a tool to explore and control the level of overfitting. As a by-product, we will see why PCA almost consistently out-performs LDA and EMI [14] on real-world datasets. Through our understanding of overfitting, we then show experimentally that we can fine-tune SDMs to out-perform even PCA on real-world datasets.

Our contributions can be summarised as follows.

- Overfitting is studied in the context of dimensionality reduction as opposed to classification or regression which have been widely studied in the literature.
- We advocate, illustrate and provide theoretical grounding for the idea of increasing within-class spread, as opposed to decreasing within-class spread which is often a consequence of many major contemporary methodologies in the literature through the practice of maximising inter-class discrimination. To the best of our knowledge, this idea is a novel one.
- We prove the triviality of minimising within-class spread in very high-dimensional input spaces (lemmas 1 and 2), and as a corollary, the existence of perfectly discriminative projections.
We propose a novel feature extraction technique, called Soft Discriminant Map (SDM), that alleviates overfitting while preserving discriminative information. A detailed account of some similar techniques in the literature is provided in section 3.2.

1.3. Possible confusion of terms: feature detection and feature extraction

The term feature extraction in the statistical pattern recognition literature [1] can be easily confused with the term feature detection in the computer vision literature [15, 16, 17]. The word feature, as in feature detection in the computer vision literature, is a relatively loosely defined term that means a shape, or a line, or a contour [15, 16], or an eye (in the context of face detection) [17]; this meaning of the word feature is similar to the natural human understanding of the word. On the other hand, the word feature, as in feature extraction or feature selection in the statistical pattern recognition literature [1, 18], means a line (a dimension) in the vector space in which a data point (e.g. a finger print image or a gene expression [19, 20, 21]) lives. In this paper we will use the word feature in the statistical pattern recognition sense, in line with [1], to mean a dimension in a vector space; we do not discuss feature detection in the computer vision sense, which is outside of the scope of this paper.

1.4. Notation

Please refer to table 1 for the notational convention used throughout this paper. In general, we use bold face for vectors and matrices, and matrices are denoted in upper case.

1.5. Dimensionality reduction techniques

Dimensionality reduction is a term comprising feature extraction and feature selection. Modern dimensionality reduction techniques abound in many variations and areas of application. In this section we briefly introduce some of the well-known techniques and their categories.
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>Input dimensionality of the data.</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of training data points.</td>
</tr>
<tr>
<td>$K$</td>
<td>Number of classes.</td>
</tr>
<tr>
<td>$I_c$</td>
<td>Index set of class $c$.</td>
</tr>
<tr>
<td>$N_c$</td>
<td>Number of training data points in class $c$.</td>
</tr>
<tr>
<td>$c_n$</td>
<td>Class label of the $n$th data point.</td>
</tr>
<tr>
<td>$x_n$</td>
<td>Vector, the $n$th data point.</td>
</tr>
<tr>
<td>$\bar{x}_c$</td>
<td>Vector, the mean of class $c$.</td>
</tr>
<tr>
<td>$w$</td>
<td>Usually a unit vector representing a feature.</td>
</tr>
<tr>
<td>$y_n$</td>
<td>The $n$th data point projected onto a feature subspace.</td>
</tr>
<tr>
<td>$W$</td>
<td>Matrix, where each column is a feature.</td>
</tr>
</tbody>
</table>

**Table 1: Notation**

1.5.1. Feature selection and feature extraction

A feature is a dimension in space, and can be represented by a unit vector. Feature selection (variable selection) is the process of selecting a subset of the original input features [18, 22]. In contrast, feature extraction constructs new features using functions of the input features [1]. The kind of feature extraction we focus on in this paper is linear feature extraction, in which the extracted features are linear combinations of the input features. Linear feature extraction includes the well-known techniques LDA and PCA (section 1.5.4), and our proposed SDM technique (section 3). Feature selection is useful in applications such as gene selection [19, 20, 21] and text classification [23, 24].

1.5.2. Linear and non-linear feature extraction

Feature extraction can be linear or non-linear. There are two senses of the term non-linear in the dimensionality reduction literature. One is that the features themselves cannot be represented as straight lines in the input space; an example of this is Kernel PCA [2, sec.12.3]. The other sense of the term “non-linear” is more in interpretation rather than in mathematical formulation,
where the algorithm is based on a non-linear concept but the features are nevertheless obtained through eigendecomposition of some objective matrix directly constructed from the data points, and therefore represent straight lines (linear) in the input space; an example of this is Isomap [25], where the algorithm is based on the non-linear geodesic distances between data points, but the solutions are linear. In this paper we focus on linear feature extraction, where we understand “linear” in the mathematical sense.

1.5.3. Information-theoretic dimensionality reduction

An relatively new approach to dimensionality reduction is to use information theory, where we maximise some estimate of the mutual information between the data points and the class labels [26, 27, 14, 22]. The main motivation for information-theoretic dimensionality reduction is that distance-based (and variance-based) dimensionality reduction techniques are liable to fail on datasets with multimodal class-conditional distributions [14]. Mutual information can potentially measure all moments of the data distribution, while variance-based methods can only measure up to the second moment (the covariance).

Torkkola [27] introduced an iterative algorithm for maximising an estimate of Kapur’s quadratic mutual information [28], and Liu and Gillies [14] introduced an estimate of mutual information that permits closed-form optimisation by eigendecomposition, called EMI (Eigenvalue-based Mutual Information). Both have been shown to give good classification performance for some datasets, however those datasets are relatively low-dimensional (at most about 30 input dimensions). For high dimensional datasets, Torkkola’s algorithm becomes computationally intensive and unstable due to the iterative nature of the algorithm, and EMI underperforms compared to PCA, for reasons that will become clear in section 2.

1.5.4. PCA and LDA

Perhaps the two best known feature extraction algorithms are Principal Component Analysis and Linear Discriminant Analysis [2]. There are numerous
variations of LDA, aimed at solving certain problems associated with LDA, such as the small sample size problem. In this section we clarify the type of LDA that we use.

LDA for multiple classes was originally formulated as the eigendecomposition of the matrix $S_w^{-1}S_b$ [2, sec. 4.1.4], where the within-class and between-class covariance matrices are given by

$$S_w := \frac{1}{N} \sum_{c=1}^{K} \sum_{n \in I_c} (x_n - \bar{x}_c)(x_n - \bar{x}_c)^T,$$

(1)

$$S_b := \sum_{c=1}^{K} \frac{N_c}{N} (\bar{x}_c - \bar{x})(\bar{x}_c - \bar{x})^T.$$

(2)

However, the rank of the within-class covariance matrix $S_w$ is $N - K$, and when $D > N - K$ this matrix is singular. To solve this problem, we use PCA+LDA [29], whereby PCA is applied to the data before LDA, and instead of using the matrix $S_w^{-1}S_b$, we use $S_t^{-1}S_b$, where $S_t$ is the total covariance matrix, $S_t = S_w + S_b$. The total covariance matrix has rank $N - 1$, and is always invertible in the principal component subspace (the space produced by applying PCA to the data), since the principal component subspace has dimensionality at most $N - 1$. It can be shown [29] that no information is lost during the application of PCA before LDA. In this paper, when we speak of LDA, we always mean PCA+LDA as described above. We will be using PCA and LDA to illustrate overfitting in high-dimensional input spaces, and as benchmarks for comparison with our newly proposed techniques, Soft Discriminant Maps (section 3).

2. High input dimensionality

In this section we study the effects of very high input dimensionality on some exemplary feature extraction algorithms, and introduce a measure of inter-class discrimination (separation) that is suitable for high-dimensional datasets.

In summary, we make two main observations. First, for high-dimensional datasets, a feature extraction algorithm with low inter-class discrimination can
Figure 2: Classification error rates on the AT&T Dataset of Faces.

Figure 3: 2D projections of the AT&T Dataset of Faces. (a) PCA; (b) EMI; (c) LDA. The classes in the EMI and LDA projections are so compact that they look like singleton data points at this resolution.
often out-perform feature extraction algorithms with much higher inter-class discrimination, despite conventional intuition. Second, in very high-dimensional input spaces, low-dimensional subspaces can always be found that exhibit perfect or near-perfect inter-class discrimination. The latter observation implies that naively aiming for maximum inter-class discrimination is trivial (and can be counter-productive).

2.1. Relationship between within-class spread and classification accuracy

First consider the AT&T Dataset of Faces, which we have already shown in figure [1]. Figure [2] shows the error rates achieved by PCA [2, sec.12.1], LDA, and EMI [13]. In all experiments in this paper, we use the nearest-neighbour classifier and the 2-fold cross-validation method, averaged over 12 repeats; the error bars on the graphs mark 1 standard deviation away from the average error rates. We see that PCA achieves considerably lower error rates than LDA and EMI. However, from figure [3] we can see that PCA clearly exhibits much lower inter-class discrimination than LDA and EMI. This is contrary to the conventional wisdom of increasing the inter-class discrimination to enhance classification performance [27, 30, 31].

In essence, this is due to overfitting, as described in section [1.1]. In the LDA and EMI projections, each class of the training data is extremely compact, and so these projections are too specific to the training data; in the PCA projection, the classes of the training data are more spread-out, and this projection embeds the true distribution of the data more accurately. However, from figure [3] we may quite reasonably suspect that perhaps the classes in the PCA projection are too spread-out, in that we can see overlapping between them; and it turns out that for this dataset, we would be right. We will see in section [3] that we can adjust the compactness of the classes in feature projections, and achieve better classification accuracy than even PCA.

It is worthwhile to remark that the above observation, that PCA outperforms more sophisticated techniques such as LDA and EMI on a real-world dataset, agrees with the observations made by Van Der Maarten et al. [32]. In their com-
parative review of the relatively new dimensionality reduction techniques (Multidimensional Scaling (MDS), Isomap, Maximum Variance Unfolding (MVU), Diffusion Maps, Multilayer Autoencoders, Locally Linear Embedding (LLE), Laplacian Eigenmaps, Hessian LLE, Local Tangent Space Analysis (LTSA), Locally Linear Coordination (LLC), and Manifold Charting) as compared with PCA and Kernel PCA [33], the authors have found that the new techniques “perform well on selected artificial tasks, but do not outperform the traditional PCA on real-world tasks.”

2.2. Class-conditional distribution and inter-class discrimination in high-dimensional input spaces

We show that maximising the compactness (minimising the spread) of each class in a feature subspace while preserving class separation is trivial when the input dimensionality is high. As a corollary this will also show that we only need to consider unimodal classes, thus obviating the need for information-theoretic feature extraction. It means that for high-dimensional datasets, we should not naively and solely focus on maximising inter-class discrimination, and that superior classification performance often cannot be explained purely based on superior inter-class discrimination, but in fact often the opposite.

Suppose we have an unsupervised dataset (that is, not separated into classes) with $D$ input dimensions and $L$ distinct clusters (not to be confused with classes) whose centres (modes, means or medians) are at $\mu_1, \ldots, \mu_L$. We speak of centres instead of modes, means or medians, for the sake of generality. Suppose we aim to find a $P$-dimensional subspace in which the centres of the clusters coincide. Then we want to solve

\[ \mu_1^T w = \mu_2^T w = \ldots = \mu_L^T w \quad (3) \]

in $w$. Any solution $w$ is a line along which the centres coincide. There may be infinitely many solutions to (3) in general, but the number of linearly independent solutions will be the maximum dimensionality of a subspace in which the centres $\mu_1, \ldots, \mu_L$ coincide. Any subspace with dimensionality higher than that
will necessarily embed a *multicentral* dataset. In general there are \( D - (L - 1) \) linearly independent solutions to equation \([3]\), since the equation imposes \( L - 1 \) constraints on \( w \), assuming that \( \mu_1, \ldots, \mu_L \) are linearly independent. Thus we have the following observation.

**Lemma 1.** Suppose a \( D \)-dimensional dataset is separated into \( L \) distinct clusters whose centres (means, medians or modes) are linearly independent from each other. There exists a \( P \)-dimensional projection such that the centres coincide, if and only if \( P \leq D - L + 1 \).

We can generalise this for supervised datasets (that is, one with two or more classes). Suppose each class \( c, 1 \leq c \leq K \), has \( L_c \) clusters, centred at \( \mu_1^{(c)}, \ldots, \mu_{L_c}^{(c)} \). Suppose we aim to find a \( P \)-dimensional subspace in which the projection of the dataset is such that each class appears to have one single cluster. We assume that each cluster falls entirely within one of the \( K \) classes, in other words no two classes share a cluster. Then we have \( K \) simultaneous sets of equations to solve in \( w \).

\[
w^T \mu_1^{(c)} = w^T \mu_2^{(c)} = \ldots = w^T \mu_{L_c}^{(c)} \quad 1 \leq c \leq K.
\]  

\[(4)\]

For each class \( c \), equation \([4]\) imposes \( L_c - 1 \) constraints on \( w \). So the total number of constraints is \( \sum_{c=1}^{K} (L_c - 1) = L - K \), where we let \( L := \sum_{c=1}^{K} L_c \), the total number of clusters. Therefore there are \( D - (L - K) \) linearly independent solutions for \( w \). Note that we implicitly assume that every class is non-empty, so each class has at least one cluster, and so \( L \geq K \). The following observation summarises this.

**Lemma 2.** Suppose a \( D \)-dimensional supervised dataset is categorised into \( K \) classes and separated into \( L \) distinct clusters whose centres (means, medians or modes) are linearly independent from each other, where \( L \geq K \). Assume each cluster falls entirely within one of the classes. There exists a \( P \)-dimensional projection of the dataset such that the centres in each class coincide, if and only if \( P \leq D - L + K \).
A trivial case of lemma 2 is when each class is already a single cluster in the input space, that is, \( L = K \). Then lemma 2 says that a subspace has unicentral classes if and only if the subspace has dimensionality at most \( D - L + K = D \), obviously.

A more interesting and more important case of lemma 2 is when we regard each data point \( x_n \) as its own singleton cluster, that is, \( L = N \). Then lemma 2 says that we can find a subspace in which the data points in each class coincide, if and only if the dimensionality of the subspace is at most \( D - N + K \). When this happens, we say that the subspace is a perfect fit, or that it is perfectly fitting for the training dataset. For the AT&T Dataset of Faces, even if we pre-process the data using PCA (which reduces the dimensionality from \( D = 10,304 \) to \( N - 1 = 399 \)), the dimensionality of a perfectly fitting feature subspace can still be up to \((N - 1) - N + K = K - 1 = 39\). So in a typical feature extraction task, we can always extract perfectly fitting subspaces, or at least subspaces in which each class is tightly packed as one unimodal cluster, and this is exemplified by LDA in figure 1a.

Another important implication of Lemma 2 is the following. Claims pertaining to explaining the superior classification performance of a feature extraction algorithm in terms of its superior inter-class discrimination [30], should be taken with care. For low-dimensional data such as the Pima Indians Diabetes dataset, superior inter-class discrimination does indeed lead to superior classification performance [14]. However, for high-dimensional data such as the AT&T Dataset of Faces, we have seen that we can always find perfectly fitting subspaces and therefore achieve perfect discrimination (in a sense), but that these are prone to overfitting (figure 1) and will give low classification accuracy (figure 2). Therefore, when overfitting is shown to occur, and when a feature extraction algorithm is shown to give superior classification accuracy, this superiority cannot be explained purely by its superior discriminative power compared with another feature extraction algorithm — there must be another explanation. An example of such a claim can be found in [30], where a feature extraction algorithm called Infomargin is introduced. The authors have shown that Infomargin can give
better inter-class discrimination than Torkkola’s QMI maximisation [27], and claim that because of this, Infomargin gives superior classification accuracy on several datasets, including the AT&T Dataset of Faces.

2.3. A measure of inter-class discrimination

We have seen that too much inter-class discrimination, as in the cases of LDA and EMI, can cause overfitting. On the other hand, we must have some inter-class discrimination, or else the classes will completely overlap and will be indistinguishable from each other. In investigating the relationship between overfitting and the level of inter-class discrimination, we introduce a measure of inter-class discrimination, closely related to LDA. It will be useful in our assessment of Soft Discriminant Maps which we will introduce in section 3.

We will call the following the $\Delta$-measure of a supervised dataset $\{y_n, c_n\}_{n=1}^N$ in a feature space.

$$\Delta(\{y_n, c_n\}_{n=1}^N) := \sqrt{\frac{\sum_{c=1}^K N_c \|\bar{y}_c - \bar{y}\|^2}{\sum_{c=1}^K \sum_{n \in I_c} \|y_n - \bar{y}_c\|^2}}.$$  \hspace{1cm} (5)

This is the square root of the ratio between the average between-class variance and the average within-class variance.

Note that in a two-class problem where the aim is to extract one feature, what is often known as the Fisher criterion [21, p.188] between two classes is
in fact $\Delta^2$ in our notation in equation \[5\]. For a multi-class problem, the Fisher criterion can be defined in a number of ways \[34, p.446\]. One of them is $\text{tr}[((W^T S_w W)^{-1}(W^T S_b W))]$ and another is $\frac{\text{tr}(W^T S_b W)}{\text{tr}(W^T S_w W)}$, where $S_b$ and $S_w$ are the between-class and within-class covariance matrices. The former is the criterion used by LDA, and the latter is equal to $\Delta^2$. We call equation \[5\] the $\Delta$-measure and not the Fisher measure, in order to avoid any confusions.

The value of the $\Delta$-measure is high when the average within-class variance is small compared to the average between-class variance. It is an intuitively evident measure of inter-class discrimination, but only under the condition that the classes are unimodal. However, lemma 2 implies that the condition of unimodal classes can always be satisfied for high-dimensional datasets. This justifies the use of the $\Delta$-measure as a measure of inter-class discrimination for the high-dimensional datasets that we study in this paper.

Figure 4 shows the $\Delta$-measure in the subspaces computed by PCA, LDA and EMI on the AT&T Dataset of Faces. An interesting observation is that the order of LDA, EMI and PCA with respect to the $\Delta$-measure is the reverse of their order with respect to classification accuracy, as shown in figure 2.

3. **Soft discrimination**

3.1. **Soft Discriminant Map (SDM)**

We propose a family of feature extractors, parameterised by a single scalar parameter $\beta$. This technique strikes a balance between discrimination and overlap between classes, and consequently circumvents the problem of overfitting. We simply consider the eigendecomposition of the following objective matrix.

$$M_\beta := S_b - \beta S_w, \quad \beta \in \mathbb{R}.$$  \[6\]

The parameter $\beta$ controls the level of inter-class discrimination. Large $\beta$ means small within-class variance and large between-class variance. Recall that LDA maximises the ratio between the between-class variance and the within-class variance, and notice that equation \[6\] is the weighted difference between them.
Figure 5: 2D projections of the AT&T Dataset of Faces computed by SDMs in (a) $\beta = 10$, (b) $\beta = 50$, (c) $\beta = 10,000$, (d) $\beta = 10,000,000$, LDA in (e), and PCA in (f).

Figure 6: Values of the $\Delta$-measure for the feature subspaces computed by SDMs for various values of $\beta$, on the AT&T Dataset of Faces.
Figure 7: Classification error rates on the AT&T Dataset of Faces.

Figure 8: 2D projection of the AT&T Dataset of Faces computed by SDM ($\beta = 50$), where the feature subspace is computed using 80% of the data. The rest 20% are marked by grey circles.
The inter-class discrimination in LDA is too “harsh”, and that is why it can cause overfitting. The proposed technique (which considers the largest eigenvectors of $M_\beta$ in equation (6)) permits a “softer” discrimination, hence the name *soft discriminant maps* (SDM).

Figure 5 shows the 2D projections computed by SDMs on the AT&T Dataset of Faces, for various values of $\beta$. For easy comparison, PCA and LDA are also included. We see how the value of $\beta$ affects the level of inter-class discrimination. Figure 6 reveals their levels of discrimination. It shows the values of the $\Delta$-measure (equation (5)) in their respective feature subspaces. Figure 7 shows their classification error rates. We see that SDMs give better classification performance than PCA, and far better than LDA. The key point here is that the reason for SDM’s superior performance over PCA is because it gives better inter-class discrimination, while its superior performance over LDA is because it gives lower inter-class discrimination (less harsh), thus avoiding overfitting. Figure 8 shows the level of overfitting in a 2D projection computed by SDM, analogous to figure 1b for LDA in section 1.1. Compared with figure 1b, we can see that the testing data points are relatively close to the corresponding training data points in the same class.

### 3.2. Other similar algorithms

In [35], the authors presented a unifying framework for many dimensionality reduction techniques including Isomap, Locally Linear Embedding, Laplacian Eigenmaps, Hessian LLE, and Local Tangent Space Alignment. The framework is called *Patch Alignment*, and has inspired a new technique which the authors call *Discriminative Locality Alignment* (DLA). DLA is based on the eigendecomposition of an objective matrix that is very similar to our SDM objective matrix in equation (6). The difference is that instead of using the whole within-class covariance matrix $S_w$, a matrix is built by considering each data point and only its nearest neighbours within its class. In contrast, using the whole within-class covariance matrix $S_w$ is equivalent to accounting for each data point and all of the other data points in the same class. Similarly, instead
of using the between-class covariance matrix $S_b$, a matrix is built by considering each data point and only its nearest neighbours outside of its class. In this way, the authors claim that local geometrical information is accounted for. The authors have demonstrated the superior classification performance of DLA over PCA, LDA and other competing algorithms on several face datasets, and have explained the superiority of DLA by the fact that it accounts for both discriminative information and local structure of the data. By its similarity to SDM however, we conjecture that the superior classification performance of DLA can also be explained in terms of overfitting, specifically that it does not suffer as much from overfitting as does LDA, and accounts for discriminative information while PCA does not, since PCA is an unsupervised technique.

Another algorithm similar to SDM is proposed in [31]. In fact, the algorithm proposed there is precisely a special case of SDM where $\beta = 1$. In other words the algorithm is an eigendecomposition of $S_b - S_w$. The authors called this the Maximum Margin Criterion. It is worth pointing out two flaws with the theory involved.

First, the name Maximum Margin Criterion is misleading. Eigenvectors of the matrix $S_b - S_w$ do not maximise the margin between classes, as can be seen in figure 5. Indeed, LDA feature subspaces and perfectly fitting projections (see lemma [2] in section 2.2) both exhibit larger margins between classes. In fact, it is because SDMs do not maximise the margin between classes that they can avoid overfitting. Therefore, we claim that the superior performance of the “maximum margin criterion” algorithm proposed in [31], when it does occur, is explained by the fact that the algorithm does not maximise the margin between classes and therefore avoids overfitting, while accounting for discriminative information at the same time.

Second, some of the authors’ claims pertaining to equivalences under certain conditions between their maximum margin criterion and LDA do not hold in general. As a particular example, the authors claim on page 161 that when the sample size is small, “[Maximum Margin Criterion] is actually equivalent to LDA + PCA”. The justification for this was claimed to be that “in order
to maximize $\text{tr}(W^T(S_b - S_w)W)$, the columns of $W$ have to be in the null space of $S_w$. This latter statement is not true. The reader may check this with the following example dataset. Consider a 2D dataset with 2 classes, each with 2 data points, which are $\{(-1,21), (-1,20)\}$ and $\{(1,-21), (1,-20)\}$ respectively. The null space of $S_w$ in this case is the line spanned by the vector $(1,0)^T$, since in this direction the within-class variance vanishes. However, the difference between the between-class variance and the within-class variance in this direction, $w^T(S_b - S_w)w$, is equal to 1, and is clearly less than that in the direction of the vector $(0,1)^T$, where the within-class variance is not 0 but equal to $\frac{1}{4}$. In general, and intuitively speaking, in order to maximise $w^T(S_b - S_w)w$, it is not necessary that the vector $w$ be in a direction where $w^TS_ww = 0$, because in this direction, $w^TS_bw$ may be small. In maximising $w^T(S_b - S_w)w$, a large $w^TS_bw$ is preferred, sometimes at the expense of a positive (non-zero) $w^TS_ww$.

3.3. Pre-processing by PCA

Prior to an execution of SDM, we recommend that the data be projected onto its maximal PCA space (the space spanned by all of the principal components), which has dimensionality $\min\{D, N-1\}$ (for $N$ linearly independent data points). For high-dimensional datasets where $D \geq N$, this is a way of reducing computational cost, since the construction of the matrix $S_w$ is in general $O(D^2(N-K))$, and that of $S_b$ is in general $O(D^2K)$.

It is natural to wonder whether any information has been lost through projecting the $D$-dimensional dataset onto the smaller $(N-1)$-dimensional PCA space. However it can be easily shown that no information is lost with respect to $S_w$ and $S_b$ through the application of PCA \cite{PCA}. Let $\Phi$ be the $(N-1)$-dimensional PCA space of a high-dimensional dataset with $D \geq N$, and let $\Phi^\perp$ be its orthogonal complement. Any $w \in \mathbb{R}^D$ can be written as $w = u + u_\perp$ where $u \in \Phi$ and $u_\perp \in \Phi^\perp$. By construction we have $u^T_Siu_\perp = 0$. Since the matrices $S_w$, $S_b$ and $S_i$ are all positive semi-definite, and since $S_w + S_b = S_i$, we see that $u^T_Swu_\perp = 0$ and $u^T_Sbu_\perp = 0$. Therefore we see that $w^T_Sww = u^T_Swu$. 

21
and $w^T S_b w = u^T S_b u$. This implies that the SDM objective $w^T (S_b - \beta S_w) w$ is the same whether we work in the input space or the PCA space. The same derivation can be found in [29] where the authors used the argument to justify the use of LDA in the PCA space.

Note that we do not necessarily recommend whitening of the data before applying SDM, but simply projecting the data onto the PCA space. Whitening is the combination of projecting the data onto the PCA space and in addition, scaling the projected data along each principal component so that the variance in all directions is one. Whitening would change the data somewhat, by scaling it (by different amounts in different directions). The user of SDM may choose to whiten the data before applying SDM, but in the experiments presented in this paper we do not. In our auxiliary experiments (not included in this paper), we have found that whitening does not produce a qualitative difference in the experimental results.

3.4. **Controlling the level of inter-class discrimination**

The essence of SDM is to be able to control the level of inter-class discrimination so as to avoid overfitting while retaining enough discriminative information in the data. Figures 5 and 6 are a schematic illustration of how the parameter $\beta$ affects the inter-class discrimination. However it would be interesting to see a more precise relationship between $\beta$ and inter-class discrimination.
SDM can be formulated as the following optimisation problem.

\[
\text{Maximise } w^T S_b w \quad \text{subject to } \quad w^T S_w w = \alpha \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \text{and } \quad w^T w = 1,
\]

where \( \alpha \) is a constant. In other words, equation (7) says “find a unit vector \( w \) along which the between-class covariance is maximised as long as the between-class covariance is equal to a given constant number”. The Lagrangian is

\[
w^T S_b w - \beta(w^T S_w w - \alpha) - \lambda(w^T w - 1).
\]

By setting the derivative with respect to \( w \) to zero we obtain

\[
(S_b - \beta S_w) w = \lambda w.
\]

As expected, equation (9) says that \( w \) is an eigenvector of \( S_b - \beta S_w \) with eigenvalue \( \lambda \). Now we would like to investigate the relationship between \( \beta \) and the inter-class discrimination along a feature, so we want to express \( \beta \) in terms of a feature \( w \) and matrices \( S_w \) and \( S_b \). Let \( \mathbf{1} \) be the vector whose every element is 1. \( \mathbf{1} := (1, \ldots, 1)^T \). We can produce two simultaneous equations from equation (9) by pre-multiplying it by \( w^T \) and \( \mathbf{1}^T \) respectively, and substitute the two equations together to eliminate \( \lambda \). We then obtain

\[
\beta = \frac{\mathbf{1}^T((ww^T - \mathbf{1})S_b]w}{\mathbf{1}^T[(ww^T - \mathbf{1})S_w]w},
\]

where \( \mathbf{1} \) is the identity matrix. For a one-dimensional feature subspace spanned
by \( \mathbf{w} \), the \( \Delta \)-measure can be written as

\[
\Delta(\mathbf{w}) = \sqrt{\frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}}
\]  

(11)

A comparison between equations (10) and (11) suggests that \( \beta \) is related directly to \( \Delta \). Note however the relationship between \( \beta \) and \( \Delta \) is stochastic and not exact, in the sense that it depends on the data which is random. For the AT&T Dataset of Faces, figure 9 shows the relationship between \( \beta \) and \( \Delta \) in 5-dimensional feature subspaces. Feature subspaces with other dimensionalities show the same result.

We see that \( \beta \) and \( \Delta \) seem to be linearly related to each other, and moreover the deviation from the optimal line is very small. This linear relationship is observed for other high-dimensional image datasets too, see section 4.

3.5. Highly representative datasets

We ought to emphasize that there exists high-dimensional datasets on which LDA does not suffer from overfitting as severely as in other datasets, as shown in figure 11. Indeed, for the JAFFE face dataset [36], figure 10 shows that LDA gives better classification accuracy than PCA. The reason for the robustness against overfitting is that the training dataset is representative of the true data distribution. Geometrically, this means that all non-zero regions of the (unknown) true data distribution contain suitable numbers of training data points,
and in general this is always a good property of a dataset. From figure 10 we can also see that SDMs out-perform both PCA and LDA. This could be because that although LDA does not suffer as much from overfitting for the JAFFE dataset as it does for the AT&T dataset of faces, it is still more affected by overfitting than SDMs (for appropriate values of $\beta$). Based on this and experiments on other datasets (section 4), we recommend that for high-dimensional datasets, even when LDA does not suffer very much from overfitting, it may still be a good idea to use SDMs. In practice, owing to the low computational complexity of PCA, LDA, and SDM, we can always plot graphs using cross-validation on the training dataset to decide which technique is best for a particular dataset.

4. Further experimental results and discussions

All experiments in this paper were implemented in Python, using the 3rd-party numerical library Numpy. All graphs and diagrams were plotted using the 3rd-party Python plotting library MatPlotLib.

4.1. Experimental results

In figures 7 and 10 we have shown the classification performance of SDMs on the AT&T Dataset of Faces and the JAFFE face dataset respectively. In this section we present some more experiments on various datasets. The datasets used and their characteristics are summarised in table 2. All evaluations are averages of 12 repeats of 2-fold cross-validation of the nearest neighbour classifier. Moreover, datasets are always pre-processed by PCA, but not whitened, before the application of SDMs. We use the nearest neighbour classifier because it is mathematically and intuitively simple, and we use a simple classifier because our focus is the evaluation of dimensionality reduction algorithms and not classifiers. The error bars on all graphs mark one standard deviation away from the average error rate.

Classification results for the four Essex datasets [37] are shown in figure 12. Figure 13 shows 2D projections of the Essex95 dataset computed by SDMs.
Figure 12: Classification error rates for the four Essex face datasets. (a) Essex94, (b) Essex95, (c) Essex96, (d) Essex Grimace.
for various values of $\beta$. The 2D projections of the other Essex datasets are completely analogous to figure 13 and so are not shown here.

Next we study the four FVC datasets (Fingerprint Verification Competition) \cite{38}. Note that this is a subset of the full database. Only this subset is available from the website \cite{38}, and not the full database. Figure 14 shows the classification results. Note that there are classification algorithms and suites (feature extractor plus classifier) far better designed for the specific task of recognising fingerprints based on these training datasets. Our focus here is not to design the best classifier specifically for this data, but rather to illustrate the breadth of applications and performance of SDMs as a feature extractor. Figure 15 shows the 2D projections of the FVC1 dataset. Again, the 2D projections for the other three FVC datasets are completely analogous and so only FVC1 is shown here.

Classification results and 2D projections of the Sheffield face dataset (formerly UMIST) \cite{39} are shown in figures 16 and 17. In the context of face recognition, this is an interesting dataset because each class (corresponding to
Figure 14: Classification error rates for the four FVC fingerprint datasets. (a) FVC1, (b) FVC2, (c) FVC3, (d) FVC4.

Figure 15: 2D projections of the FVC1 dataset computed by SDMs for various values of $\beta$. (a) PCA, (b) SDM $\beta = 10$, (c) SDM $\beta = 100$, (d) SDM $\beta = 1000$. 
Figure 16: Classification error rates for the Sheffield dataset.

Figure 17: 2D projections of the Sheffield dataset, computed by SDMs for various values of $\beta$. (a) PCA, (b) SDM $\beta = 10$, (c) SDM $\beta = 100$, (d) SDM $\beta = 1000$. 
<table>
<thead>
<tr>
<th>Dataset name</th>
<th>N</th>
<th>K</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT&amp;T</td>
<td>400</td>
<td>40</td>
<td>10304</td>
</tr>
<tr>
<td>Essex94</td>
<td>3040</td>
<td>152</td>
<td>36000</td>
</tr>
<tr>
<td>Essex95</td>
<td>1440</td>
<td>72</td>
<td>36000</td>
</tr>
<tr>
<td>Essex96</td>
<td>3018</td>
<td>151</td>
<td>38416</td>
</tr>
<tr>
<td>Essex Grimace</td>
<td>360</td>
<td>18</td>
<td>36000</td>
</tr>
<tr>
<td>JAFFE</td>
<td>213</td>
<td>10</td>
<td>65536</td>
</tr>
<tr>
<td>FVC1</td>
<td>80</td>
<td>10</td>
<td>90000</td>
</tr>
<tr>
<td>FVC2</td>
<td>80</td>
<td>10</td>
<td>93184</td>
</tr>
<tr>
<td>FVC3</td>
<td>80</td>
<td>10</td>
<td>214144</td>
</tr>
<tr>
<td>FVC4</td>
<td>80</td>
<td>10</td>
<td>76800</td>
</tr>
<tr>
<td>Sheffield</td>
<td>565</td>
<td>20</td>
<td>10304</td>
</tr>
</tbody>
</table>

Table 2: High-dimensional datasets and their characteristics. \( N = \) total number of data points; \( K = \) number of classes; \( D = \) input dimensionality. The FVC datasets are fingerprints, the rest are faces.

Each individual contains several angles of view of an individual’s face, varying gradually from profile to frontal. This rotation in our physical world would translate to a one-dimensional manifold (a curve) in the input space, where the one dimension corresponds to the one degree of physical freedom in the rotation of the head. We can verify this by looking at figure 17a, where we can see the intuitively satisfying pattern that the shape of each class seems to be of a one-dimensional curve.

Note that LDA can extract at most \( K - 1 \) features due to its formulation, and so in the FVC datasets for example, where \( K = 10 \), the error rates of LDA were only shown for feature subspaces of dimensionality up to 9.

4.2. Discussion of experimental results

4.2.1. Error rates

With the exception of the JAFFE dataset (figure 10) and a couple of special but minor cases in the FVC2 dataset (figure 14b), LDA exhibits the worst
classification accuracy. As discussed in sections 1.1 and 2, this is due to the fact that the level of inter-class discrimination is too high in that the classes are too compact in the feature space, causing overfitting.

We also observe that, again with the exception of the JAFFE dataset and some special cases in the FVC dataset (figures 14b, 14c, and 14d), the classification performance of PCA is better than LDA but worse than our proposed SDMs. The superiority of PCA in these cases over LDA is explained by the fact that PCA does not overfit the input data, in other words PCA exhibits a lower inter-class discrimination; the inferiority of PCA to SDMs is explained by the fact that the inter-class discrimination in PCA is in fact too low, and therefore suffers from overlapping of classes (ambiguity in the class membership of data points). This is not surprising, since PCA is an unsupervised technique and does not account for discriminative information in the data.

Importantly, we observe that without exception, we obtain the best classification accuracy with our proposed SDMs. For some low number of dimensions in some FVC datasets, SDMs are observed to be suboptimal, but considering a wider range of number of dimensions, SDMs still give the best performance. This is due to the fact that SDMs give controllably lower levels of inter-class discrimination than LDA (but not too low, as does PCA) while retaining discriminative information in the data.

The amount by which the classification error rates of SDMs are lower than PCA and LDA varies between datasets. This depends on the representativeness of the input data and the underlying data distribution. For the Essex Grimace dataset for example, when the dimensionality of the feature subspace approaches 20, there is very little difference in classification error between LDA, PCA and SDMs. Interestingly, the author of the Essex datasets, Spacek [37], said that “[Essex]96 and grimace are the most difficult, though for two different reasons (variation of background and scale, versus extreme variation of expressions)”. In our analysis and understanding, the factors that make a face dataset “difficult”, such as variation of background, scale and facial expressions, actually make it easier for highly discriminative techniques such as LDA, because the input data
is made more representative of the true data distribution by the aforementioned variations, and so LDA suffers less from overfitting. This explains why LDA exhibits a good classification performance on the Essex Grimace dataset, though not better than SDMs.

The JAFFE dataset is somewhat unusual, and we gave an analysis of it in section 3.5.

4.2.2. Optimal values of $\beta$

The best values of $\beta$ (the free parameter in our proposed SDM technique) can be different for different datasets. Moreover, the sensitivity of the classification accuracy to the value of $\beta$ varies between datasets. For example, from figure 12 we can see that the Essex94 dataset is relatively insensitive to the value of $\beta$, whereas the Essex95 dataset is observably more sensitive, in that there is a greater variation in classification accuracy for the same range of $\beta$ (between $\beta = 10$ and $\beta = 1000$). Figure 12b suggests that the best value for $\beta$ may be around 10. It is interesting to note that all of the datasets used in this paper, except Essex95, are somewhat insensitive to the value of $\beta$.

4.3. Relationship between $\beta$ and inter-class discrimination

Figure 18 shows the near-linear relationship between $\beta$ and the $\Delta$-measure for some of the datasets we studied, analogous to figure 9 for the AT&T dataset of faces. Corresponding plots for the other dataset we have studied show com-
pletely analogous patterns to those shown in figure 18 and so are not included here.

5. Conclusions

Our conclusions can be summarised by the questions below.

- **When does overfitting occur in feature extraction?** It occurs when the input dimensionality is high and when an algorithm puts too much emphasis on maximising inter-class discrimination. This is the case when we apply LDA or EMI to high-dimensional image datasets. LDA maximises the Fisher criterion while EMI maximises mutual information, both are measures of inter-class discrimination.

- **Does LDA always lead to overfitting?** No. On some low-dimensional datasets, both LDA and EMI are observed to have better performance than PCA [27, 14]. But on high-dimensional datasets ($D > 1000$, say), overfitting becomes drastically more likely.

- **Why does PCA have superior performance on many real-world datasets, despite being simpler (less sophisticated) in formulation?** Because PCA does not over-emphasise maximising inter-class discrimination, and so embed the true distribution of the input data more faithfully when the input dimensionality is high.

- **Is inter-class discrimination bad?** No, we need a certain level of inter-class discrimination to separate the classes, but too much of it will lead to overfitting.

- **How do we control the level of inter-class discrimination in order to alleviate or avoid overfitting?** Using Soft Discriminant Maps, we can control the level of inter-class discrimination and consistently achieve better performance than even PCA.
• Why do Soft Discriminant Maps perform better than PCA on real-world image datasets? Because for these datasets, the level of inter-class discrimination provided by PCA is too low (whereas in LDA and similar techniques, it is too high). SDMs provide some inter-class discrimination, while not so much as to lead to overfitting.

From our conclusions we recommend that researchers and practitioners steer their attention away from the intuitively satisfying practice of maximising inter-class discrimination, when the input dimensionality is high. Indeed, lemmas 1 and 2 show that having perfectly compact classes, perfectly separated from each other, is trivial. Often, the opposite needs to be done in the high-dimensional setting, i.e. we need to decrease the level of inter-class discrimination. If in doubt, use SDMs with various values for the parameter \( \beta \) to observe and control the effect of overfitting.

6. Future research — the maximum-spread-zero-overlap principle

The strength of SDM is rooted in soft inter-class discrimination, the idea that feature subspaces in which the classes are more spread-out, rather than compact, are more robust against overfitting. Feature subspaces that maximise compactness of classes and distance between classes, such as those computed by LDA or EMI, reflect specific learning hypotheses [40, sec.2.3.1], and often too specific. The intuition of compact classes is as follows, “any new data point must be within this very small vicinity of this class, in order to be classified positive for (belonging to) this class, otherwise it will be classified negative for this class”. This criterion is clearly very specific, and we can intuitively see how this may well lead to overfitting. We need more general criteria for deciding, for each new (testing) data point, its membership of a class. These more general criteria, or learning hypotheses, correspond to less compact and more spread-out classes. We have seen that SDMs can provide feature subspaces that embed more spread-out classes (for appropriate values of \( \beta \)). However, we have also noted in PCA that when classes are too spread-out, overlapping between classes
may occur and classification performance may again be damaged. Thus, it is natural to wonder whether the optimal level of spread of each class may be one such that the classes are as spread-out as possible without overlapping. This is the 
maximum-spread-zero-overlap principle (MSZO). Note that this is the opposite of the conventional wisdom of maximising between-class distance and minimising within-class spread, or maximising inter-class discrimination.

We now describe an algorithm aimed at achieving the MSZO principle. It is closely related to Support Vector Machines (SVMs) in formulation 7. First consider a two-class problem, where the two classes are labelled by 1 and −1 respectively. Suppose for the moment that the dataset is separable, that is to say that the two classes can be separated by a (D − 1)-dimensional hyperplane. Suppose each training data point \( x_n \) is labelled by \( c_n \) where \( c_n = 1 \) or \( -1 \). This is the same setup as in SVMs. For each \( n \), let us denote \( z_n := w^T x_n + b \), for some feature \( w \) and bias parameter \( b \). Since we assume that the dataset is separable, there exists \( w \) and \( b \) such that \( z_n \geq 0 \) whenever \( c_n = 1 \) and \( z_n \leq 0 \) whenever \( c_n = -1 \). So for such \( w \) and \( b \), we have \( c_n z_n \geq 0 \) for all \( n \). This is the zero-overlap condition. So we have the following optimisation problem.

\[
\begin{align*}
\text{Maximise} \quad & w^T S w \\
\text{subject to} \quad & w^T w = 1 \\
& c_n z_n \geq 0.
\end{align*}
\]

This is the precise statement of the MSZO problem. We can compare it with the underlying optimisation problem in SVMs.

\[
\begin{align*}
\text{Minimise} \quad & w^T w \\
\text{subject to} \quad & c_n z_n \geq 1.
\end{align*}
\]

To solve MSZO, we can employ numerical algorithms in non-linear programming similar to those used in SVMs. Note that in its current form, the MSZO problem is not quite a quadratic programming problem since the constraint \( w^T w = 1 \) is not linear in \( w \). We refer the reader to [41] for an extensive discussion of constrained optimisation.

The problem stated in (12) is under the assumption that the classes are separable. For non-separable datasets, we can project the data points onto a higher-dimensional space though a mapping \( \phi : x \mapsto \phi(x) \), in which the classes are
separable. One example of such a mapping is \( \phi(x) := (k(x, x_1), \ldots, k(x, x_N))^T \), where \( k(\cdot, \cdot) \) is a kernel. Note however that for high-dimensional datasets, lemma 2 implies that the classes are always separable in the input space. However if \( D > N \), it may still be a good idea to use a kernel mapping described above, or pre-process the data using PCA, in order to improve computational efficiency and to avoid overfitting. We can extend this technique to multiple classes via one-versus-the-rest or similar approaches.

References


