Maximum Entropy Linear Manifold for Learning Discriminative Low-Dimensional Representation

Wojciech Marian Czarnecki[1(✉)], Rafal Jozefowicz[2], and Jacek Tabor[1]

[1] Faculty of Mathematics and Computer Science,
Jagiellonian University, Krakow, Poland
{wojciech.czarnecki,jacek.tabor}@uj.edu.pl
[2] Google, New York, USA
rafjoz@gmail.com

Abstract. Representation learning is currently a very hot topic in modern machine learning, mostly due to the great success of the deep learning methods. In particular low-dimensional representation which discriminates classes can not only enhance the classification procedure, but also make it faster, while contrary to the high-dimensional embeddings can be efficiently used for visual based exploratory data analysis.

In this paper we propose Maximum Entropy Linear Manifold (MELM), a multidimensional generalization of Multithreshold Entropy Linear Classifier model which is able to find a low-dimensional linear data projection maximizing discriminativeness of projected classes. As a result we obtain a linear embedding which can be used for classification, class aware dimensionality reduction and data visualization. MELM provides highly discriminative 2D projections of the data which can be used as a method for constructing robust classifiers.

We provide both empirical evaluation as well as some interesting theoretical properties of our objective function such us scale and affine transformation invariance, connections with PCA and bounding of the expected balanced accuracy error.

Keywords: Dense representation learning · Data visualization · Entropy · Supervised dimensionality reduction

1 Introduction

Correct representation of the data, consistent with the problem and used classification method, is crucial for the efficiency of the machine learning models. In practice it is a very hard task to find suitable embedding of many real-life objects in $\mathbb{R}^d$ space used by most of the algorithms. In particular for natural language processing [12], cheminformatics or even image recognition tasks it is still an open problem. As a result there is a growing interest in methods of representation learning [8], suited for finding better embedding of our data, which may be further used for classification, clustering or other analysis purposes. Recent

A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 52–67, 2015.
DOI: 10.1007/978-3-319-23528-8_4

years brought many success stories, such as dictionary learning [13] or deep learning [9]. Many of them look for a sparse [7], highly dimensional embedding which simplify linear separation at a cost of making visual analysis nontrivial. A dual approach is to look for low-dimensional linear embedding, which has advantage of easy visualiation, interpretation and manipulation at a cost of much weaker (in terms of models complexity) space of transformations.

In this work we focus on the scenario where we are given labeled dataset in $\mathbb{R}^d$ and we are looking for such low-dimensional linear embedding which allows to easily distinguish each of the classes. In other words we are looking for a highly discriminative, low-dimensional representation of the given data.

Our basic idea follows from the observation [15] that the density estimation is credible only in the low-dimensional spaces. Consequently, we first project the data onto an arbitrary k-dimensional affine submanifold $\mathcal{V}$ (where k is fixed), and search for the $\mathcal{V}$ for which the estimated densities of the projected classes are orthogonal to each other, where the Cauchy-Schwarz Divergence is applied as a measure of discriminativeness of the projection, see Fig. 1 for an example of such projection preserving classes' separation. The work presented in this paper is a natural extension of our earlier results [6], where we considered the one-dimensional case. However, we would like to emphasize that the used approach needed a nontrivial modification. In the one-dimensional case we could identify subspaces with elements of the unit sphere in a natural way. For higher dimensional subspaces such an identification is no longer possible.

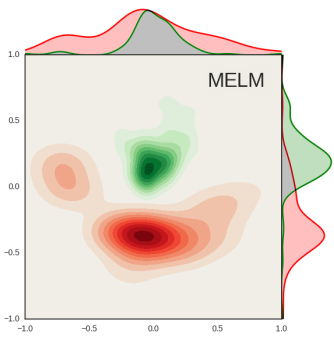

Fig. 1. Visualizatoin of *sonar* dataset using Maximum Entropy Linear Manifold with $k = 2$.

To the authors best knowledge the presented idea is novel, and has not been earlier considered as a method of classification and data visualization. As one of its benefits is the fact that it does not depend on affine rescaling of the data, which is a rare feature of the common classification tools. What is also interesting, we show that as its simple limiting one-class case we obtain the classical PCA projection. Moreover, from the theoretical standpoint the Cauchy-Schwarz Divergence factor can be decomposed into the fitting term, bounding the expected balanced misclassification error, and regularizing term, simplifying the resulting model. We compute its value and derivative so one can use first-order optimization to find a solution even though the true optimization should be performed on a Steifel manifold. Empirical tests show that such a method not only in some cases improves the classification score over learning from raw data but, more importantly, consistently finds highly discriminative representation which can be easily visualized. In particular, we show that resulting projections' discriminativeness is much higher than many popular linear methods, even recently

proposed GEM model [11]. For the sake of completness we also include the full source code of proposed method in the supplementary material.

2 General Idea

In order to visualize dataset in $\mathbb{R}^d$ we need to project it onto $\mathbb{R}^k$ for very small k (typically 2 or 3). One can use either linear transformation or some complex embedding, however choosing the second option in general leads to hard interpretability of the results. Linear projections have a tempting characteristics of being both easy to understand (from both theoretical perspective and practical implications of the obtained results) as well as they are highly robust in further application of this transformation.

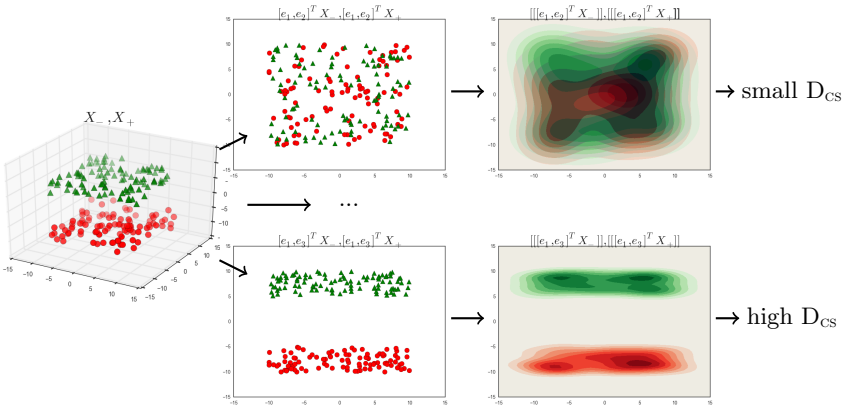

Fig. 2. Visualization of the MELM idea. For given dataset $\mathrm{X}_-, \mathrm{X}_+$ we search through various linear projections V and analyze how divergent are their density estimations in order to select the most discriminative.

In this work we focus on such class of projections so in practise we are looking for some matrix $\mathrm{V} \in \mathbb{R}^{d \times k}$, such that for a given dataset $\mathrm{X} \in \mathbb{R}^{d \times N}$ projection $\mathrm{V}^T\mathrm{X}$ preserves as much of the important information about X as possible (sometimes additionally under additional constraints). The choice of the definition of *information measure* IM together with the set of constraints φ_i defines a particular reduction method.

$$\begin{aligned} \underset{\mathrm{V} \in \mathbb{R}^{d \times k}}{\text{maximize}} \quad & \mathrm{IM}(\mathrm{V}^T\mathrm{X}; \mathrm{X}, \mathrm{Y}) \\ \text{subject to} \quad & \varphi_i(\mathrm{V}),\ i = 1, \ldots, m. \end{aligned}$$

There are many transformations which can achieve such results. For example, the well known Principal Component Analysis defines important information as data scattering so it looks for V which preserves as much of the X variance as

possible and requires V to be orthogonal. In information bottleneck method one defines this measure as amount of mutual information between X and some additional Y (such as set of labels) which has to be preserved. Similar approaches are adapted in recently proposed Generalized Eigenvectors for Discriminative Features (GEM) where one tries to preserve the signal to noise ratio between samples from different classes. In case of Maximum Entropy Linear Manifold (MELM), introduced in this paper, important information is defined as the discriminativness of the samples from different classes with orthonormal V. In other words we work with labeled samples (in general, binary labeled) and wish to preserve the ability to distinguish one class (X_-) from another (X_+). In more formal terms, our optimization problem is to

$$\begin{aligned} \underset{\mathrm{V}\in\mathbb{R}^{d\times k}}{\text{maximize}} \quad & \mathrm{D}_{\mathrm{CS}}([\![\mathrm{V}^T\mathrm{X}_-]\!],[\![\mathrm{V}^T\mathrm{X}_+]\!]) \\ \text{subject to} \quad & \mathrm{V}^T\mathrm{V} = I, \end{aligned}$$

where $\mathrm{D}_{\mathrm{CS}}(\cdot,\cdot)$ denotes the Cauchy-Schwarz Divergence, the measure of how divergent are given probability distributions; $[\![\cdot]\!]$ denotes some density estimator which, given samples, returns a probability distribution. The general idea is also visualized on Fig. 2.

3 Theory

We first discuss the one class case which has mainly introductory character as it shows the simplified version of our main idea.

Suppose that we have unlabeled data $\mathrm{X} \subset \mathbb{R}^d$ and that we want to reduce the dimension of the data (for example to visualize it, reduce outliers, etc.) to $k < d$. One of the possible approaches is to use information theory and search for such k-dimensional subspace $\mathcal{V} \subset \mathbb{R}^d$ for which the orthogonal projection of X onto $\mathcal{V}$ preserves as much information about X as possible.

One can clearly choose various measures of information. In our case, due to computational simplicity, we have decided to use Renyi's quadratic entropy, which for the density f on $\mathbb{R}^k$ is given by

$$\mathrm{H}_2(f) = -\log \int_{\mathbb{R}^k} f^2(\mathrm{x})d\mathrm{x}.$$

One can equivalently use information potential [14], which is given as the L^2 norm of the density $\mathrm{ip}(f) = \int_{\mathbb{R}^k} f^2(\mathrm{x})d\mathrm{x}$. We need an easy observation that one can compute the Renyi's quadratic entropy for the normal density $\mathcal{N}(m, \Sigma)$ in $\mathbb{R}^k$ [4]:

$$\mathrm{H}_2(\mathcal{N}(m,\Sigma)) = \tfrac{k}{2}\log(4\pi) + \tfrac{1}{2}\log(\det\Sigma). \tag{1}$$

However, in order to compute the Renyi's quadratic entropy of the discrete data we first need to apply some density estimation technique. By joining all the above mentioned steps together we are able to pose the basic optimization problem we are interested in.

Optimization problem 1. *Suppose that we are given data* X*, and* k *which denotes the dimension reduction. Find the orthonormal base* V *of the* k*-dimensional subspace*[1] $\mathcal{V}$ *for which the value of* $\mathrm{H}_2(\llbracket \mathrm{V}^T\mathrm{X} \rrbracket)$ *is maximal, where* $\llbracket \cdot \rrbracket$ *denotes a given fixed method of density estimation.*

If we have data X with mean m and covariance Σ in $\mathbb{R}^d$ and k orthonormal vectors $\mathrm{V} = [\mathrm{V}_1, \ldots, \mathrm{V}_k]$ then the we can ask what will be the mean and covariance of the orthogonal projection of X onto the space spanned by V. It is easy to show that it is given by $\mathrm{V}^T m$ and $\mathrm{V}^T \Sigma \mathrm{V}$. In other words, if we consider data in the base given by orthonormal extension of V to the whole $\mathbb{R}^d$, the covariance of the projected data corresponds to the left upper $k \times k$ block submatrix of the original covariance.

We are going to show that if we apply the simplest density estimation of the underlying density for projected data given by the maximal likelihood estimator over the family of normal densities[2] then our optimization problem is equivalent to taking first k elements of the base given by PCA.

Theorem 1. *Let* $\mathrm{X} \subset \mathbb{R}^d$ *be a given dataset with mean* m *and covariance* Σ *and let* $\llbracket \cdot \rrbracket_{\mathcal{N}}$ *denote the density estimation which returns the maximum likelihood estimator over Gaussian densities. Then*

$$\max\{\mathrm{H}_2(\llbracket \mathrm{V}^T\mathrm{X} \rrbracket_{\mathcal{N}}) : \mathrm{V} \in \mathbb{R}^{d\times k}, \mathrm{V}^T\mathrm{V} = I\}$$

is realized for the first k *orthonormal vectors given by the PCA and*

$$\min\{\mathrm{H}_2(\llbracket \mathrm{V}^T\mathrm{X} \rrbracket_{\mathcal{N}}) : \mathrm{V} \in \mathbb{R}^{d\times k}, \mathrm{V}^T\mathrm{V} = I\}$$

is realized for the last k *orthonormal vectors defined by the PCA.*

Proof. By the comments before and (1) we have

$$\mathrm{H}_2(\llbracket \mathrm{V}^T\mathrm{X} \rrbracket_{\mathcal{N}}) = \tfrac{k}{2}\log(4\pi) + \tfrac{1}{2}\log(\det(\mathrm{V}^T\Sigma\mathrm{V})).$$

In other words we search for these V for which the value of $\det(\mathrm{V}^T\Sigma\mathrm{V})$ is maximized. Now by Cauchy interlacing theory [2] eigenvalues of $\mathrm{V}^T\Sigma\mathrm{V}$ (ordered decreasingly) are bounded above by the eigenvalues of Σ. Consequently, the maximum is obtained in the case when V denotes the orthonormal eigenvectors of Σ corresponding to the biggest eigenvalues of Σ. However, this is exactly the first k elements of the orthonormal base constructed by the PCA. Proof of the second part is fully analogous.

As a result we obtain some general intuition that maximization of the Renyi's quadratic entropy leads to the selection of highly spread data, while its minimization selects projection where image is very condensed.

[1] We identify those vectors with a linear space spanned over them.
[2] That is for $A \subset \mathcal{V}$ we put $\llbracket A \rrbracket_{\mathcal{N}} = \mathcal{N}(m_A, \mathrm{cov}_A) : \mathcal{V} \to \mathbb{R}_+$.

Let us now proceed to the binary labeled data. Recall that $\mathrm{D_{CS}}$ can be equivalently expressed in terms of Renyi's quadratic entropy ($\mathrm{H_2}$) and Renyi's quadratic cross entropy ($\mathrm{H_2^\times}$):

$$\begin{aligned}\mathrm{D_{CS}}(\mathrm{V}) &= \log \int [\![\mathrm{V}^T\mathrm{X}_+]\!]^2 + \log \int [\![\mathrm{V}^T\mathrm{X}_-]\!]^2 - 2\log \int [\![\mathrm{V}^T\mathrm{X}_+]\!][\![\mathrm{V}^T\mathrm{X}_-]\!] \\ &= -\mathrm{H_2}([\![\mathrm{V}^T\mathrm{X}_-]\!]) - \mathrm{H_2}([\![\mathrm{V}^T\mathrm{X}_+]\!]) + 2\mathrm{H_2^\times}([\![\mathrm{V}^T\mathrm{X}_+]\!], [\![\mathrm{V}^T\mathrm{X}_-]\!]).\end{aligned}$$

Let us recall that our optimization aim is to find a sequence V consisting of k orthonormal vectors for which $\mathrm{D_{CS}}(\mathrm{V})$ is maximized.

Observation 1. *Assume that the density estimator $[\![\cdot]\!]$ does not change under the affine change of the coordinate system*[3]*. One can show, by an easy modification of the theorem by Czarnecki and Tabor [6, Theorem4.1], that the maximum of $\mathrm{D_{CS}}(\cdot)$ is independent of the affine change of data. Namely, for an arbitrary affine invertible map M we have:*

$$\begin{aligned}&\max\{\mathrm{D_{CS}}(\mathrm{V};\mathrm{X}_+,\mathrm{X}_-) : \mathrm{V} \textit{ orthonormal}\} \\ =&\max\{\mathrm{D_{CS}}(\mathrm{V};\mathrm{X}_+,\mathrm{X}_-) : \mathrm{V} \textit{ linearly independent}\} \\ =&\max\{\mathrm{D_{CS}}(\mathrm{V};M\mathrm{X}_+,M\mathrm{X}_-) : \mathrm{V} \textit{ orthonormal}\}.\end{aligned}$$

The above feature, although typical in the density estimation, is rather uncommon in modern classification tools.

Similarly to the one-dimensional case, when $\mathrm{V} \in \mathbb{R}^d$, we can decompose the objective function into fitting and regularizing terms:

$$\mathrm{D_{CS}}(\mathrm{V}) = \underbrace{2\mathrm{H_2^\times}([\![\mathrm{V}^T\mathrm{X}_+]\!], [\![\mathrm{V}^T\mathrm{X}_-]\!])}_{\text{fitting term}} - \underbrace{(\mathrm{H_2}([\![\mathrm{V}^T\mathrm{X}_-]\!]) + \mathrm{H_2}([\![\mathrm{V}^T\mathrm{X}_+]\!]))}_{\text{regularizing term}}.$$

Regularizing term has a slightly different meaning than in most of the machine learning models. Here it controls number of disjoint regions which appear after performing density based classification in the projected space. For one dimensional case it is a number of thresholds in the multithreshold linear classifier, for $k = 2$ it is the number of disjoint curves defining decision boundary, and so on. Renyi's quadratic entropy is minimized when each class is as condensed as possible (as we show in Theorem 1), intuitively resulting in a small number of disjoint regions.

It is worth noting that, despite similarities, it is not the common classification objective which can be written as an additive loss function and a regularization term

$$L(\mathrm{V}) = \sum_{i=1}^{N} \ell(\mathrm{V}^T\mathrm{x}_i, \mathrm{y}_i, \mathrm{x}_i) + \Omega(\mathrm{V}),$$

as the error depends on the relations between each pair of points instead of each point independently. One can easily prove that there are no ℓ, Ω for which

[3] This happens in particular for the kernel density estimation we apply in the paper.

$\mathrm{D}_{\mathrm{CS}}(v) = L(\mathrm{V}; \ell, \Omega)$. Such choice of the objective function might lead to the lack of connections with optimization of any reasonable accuracy related metric, as those are based on the point-wise loss functions. However it appears that D_{CS} bounds the expected balanced accuracy[4] similarly to how hinge loss bounds 0/1 loss. This can be formalized in the following way.

Theorem 2. *Negative log-likelihood of balanced misclassification in k-dimensional linear projection of any non-separable densities $f_{\pm}$ onto* V *is bounded by half of the Renyi's quadratic cross entropy of these projections.*

Proof. Likelihood of balanced misclassification over a k-dimensional hypercube after projection through V equals

$$\int_{[0,1]^k} \min\{(\mathrm{V}^T f_+)(\mathrm{x}), (\mathrm{V}^T f_-)(\mathrm{x})\} d\mathrm{x}.$$

Using analogous reasoning to the one presented by Czarnecki [5], using Cauchy and other basic inequalities, one can show that

$$-\log \int_{[0,1]^k} \min\{(\mathrm{V}^T f_+)(\mathrm{x}), (\mathrm{V}^T f_-)(\mathrm{x})\} d\mathrm{x} \geq \tfrac{1}{2} \mathrm{H}_2^{\times}(\mathrm{V}^T f_+, \mathrm{V}^T f_-).$$

□

As a result we might expect that maximizing of the D_{CS} leads to the selection of the projection which on one hand maximizes the balanced accuracy over the training set (minimizes empirical error) and on the other fights with overfitting by minimizing the number of disjoint classification regions (minimizes model complexity).

4 Closed form Solution for Objective and its Gradient

Let us now investigate more practical aspects of proposed approach. We show the exact formulas of both D_{CS} and its gradient as functions of finite, labeled samples (binary datasets) so one can easily plug it in to any first-order optimization software.

Let $\mathrm{X}_+, \mathrm{X}_-$ be fixed subsets of $\mathbb{R}^d$. Let $\mathcal{V}$ denote the k-dimensional subspace generated by $\mathrm{V} = [\mathrm{V}_1, \ldots, \mathrm{V}_k] \in \mathbb{R}^{d \times k}$ (we consider only the case when the sequence V is linearly independent). We project sets $\mathrm{X}_{\pm}$ orthogonally on V, and compute the Cauchy-Schwarz Divergence of the kernel density estimations (using Silverman's rule) of the resulting projections:

$$\mathrm{G}^{-1}(\mathrm{V})[\![\mathrm{V}^T \mathrm{X}_+]\!] \text{ and } \mathrm{G}^{-1}(\mathrm{V})[\![\mathrm{V}^T \mathrm{X}_-]\!],$$

where $\mathrm{G}(\mathrm{V}) = \mathrm{V}^T\mathrm{V}$ denotes the grassmanian. We search for such V for which the Cauchy-Schwarz Divergence is maximal. Recall that the scalar product in the space of matrices is given by $\langle \mathrm{V}_1, \mathrm{V}_2 \rangle = \mathrm{tr}(\mathrm{V}_1^T \mathrm{V}_2)$.

[4] Accuracy without class priors BAC(TP, FP, TN, FN) $= \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP+FN}} + \frac{\mathrm{TN}}{\mathrm{TN+FP}}\right)$.

There are basically two possible approaches one can apply: either search for the solution in the set of orthonormal V which generate $\mathcal{V}$, or allow all V with a penalty function. The first method is possible[5], but does not allow use of most of the existing numerical libraries as the space we work in is highly nonlinear. This is the reason why we use the second approach which we describe below.

Since, as we have observed in the previous section, the result does not depend on the affine transformation of data, we can restrict to the analogous formula for the sets

$$\mathrm{V}^T\mathrm{X}_+ \text{ and } \mathrm{V}^T\mathrm{X}_-,$$

where V consists of linearly independent vectors. Consequently, we need to compute the gradient of the function

$$\begin{aligned}\mathrm{D}_{\mathrm{CS}}(\mathrm{V}) &= \mathrm{D}_{\mathrm{CS}}([\![\mathrm{V}^T\mathrm{X}_+]\!], [\![\mathrm{V}^T\mathrm{X}_-]\!]) \\ &= \log \int [\![\mathrm{V}^T\mathrm{X}_+]\!]^2 + \log \int [\![\mathrm{V}^T\mathrm{X}_-]\!]^2 - 2\log \int [\![\mathrm{V}^T\mathrm{X}_+]\!][\![\mathrm{V}^T\mathrm{X}_-]\!],\end{aligned}$$

where we consider the space consisting only of linearly independent vectors. Since as the base of the space V we can always take orthonormal vectors, the maximum is realized for orthonormal sequence, and therefore we can add a penalty term for being non-orthonormal sequence, which helps avoiding numerical instabilities:

$$\mathrm{D}_{\mathrm{CS}}(\mathrm{V}) - \|\mathrm{V}^T\mathrm{V} - I\|^2,$$

where as we recall the sequence V is orthonormal iff $\mathrm{V}^T\mathrm{V} = I$. We denote above augmented D_{CS} by the *maximum entropy linear manifold* objective function

$$\mathrm{MELM}(\mathrm{V}) = \mathrm{D}_{\mathrm{CS}}(\mathrm{V}) - \|\mathrm{V}^T\mathrm{V} - I\|^2. \tag{2}$$

Besides $\mathrm{MELM}(\cdot)$ value we need the formula for its gradient $\nabla\mathrm{MELM}(\cdot)$. For the second term we obviously have

$$\nabla\|\mathrm{V}^T\mathrm{V} - I\|^2 = 4\mathrm{V}\mathrm{V}^T\mathrm{V} - 4\mathrm{V}.$$

We consider the first term. Let us first provide the formula for the computation of the product of kernel density estimations of two sets.

Assume that we are given set $A \subset \mathcal{V}$ (in our case A will be the projection of $\mathrm{X}_\pm$ onto V), where $\mathcal{V}$ is k-dimensional. Then the formula for the kernel density estimation with Gaussian kernel, is given by [15]:

$$[\![A]\!] = \frac{1}{|A|}\sum_{a\in A}\mathcal{N}(a, \Sigma_A),$$

where $\Sigma_A = (h_A^\gamma)^2\mathrm{cov}_A$ and (for γ being a scaling hyperparameter [6]) $h_A^\gamma = \gamma(\frac{4}{k+2})^{1/(k+4)}|A|^{-1/(k+4)}$.

[5] And has advantage of having smaller number of parameters.

Now we need the formula for $\int [\![A]\!][\![B]\!]$, which is calculated [6] with the use of

$$\int \mathcal{N}(a, \Sigma_A)\mathcal{N}(b, \Sigma_B) = \mathcal{N}(a-b, \Sigma_A + \Sigma_B)(0).$$

Then we get

$$\begin{aligned}\int [\![A]\!][\![B]\!] &= \frac{1}{|A||B|} \sum_{w \in A-B} \mathcal{N}(w, \Sigma_A + \Sigma_B)(0) \\ &= \frac{1}{(2\pi)^{k/2} \det{}^{1/2}(\Sigma_{AB})|A||B|} \sum_{w \in A-B} \exp(-\tfrac{1}{2}\|w\|^2_{\Sigma_{AB}}),\end{aligned}$$

where $A - B = \{a - b : a \in A, b \in B\}$ and Σ_{AB} is defined by

$$\begin{aligned}\Sigma_{AB} &= (h^{\gamma}_A)^2 \mathrm{cov}_A + (h^{\gamma}_B)^2 \mathrm{cov}_B \\ &= \gamma^2 (\tfrac{4}{k+2})^{2/(k+4)} (|A|^{-2/(k+4)} \mathrm{cov}_A + |B|^{-2/(k+4)} \mathrm{cov}_B).\end{aligned}$$

For a sequence $\mathrm{V} = [\mathrm{V}_1, \ldots, \mathrm{V}_k] \in \mathbb{R}^{d \times k}$ of linearly independent vectors we put

$$\Sigma_{AB}(\mathrm{V}) = \mathrm{V}^T \Sigma_{AB} \mathrm{V} \text{ and } S_{AB}(\mathrm{V}) = \Sigma_{AB}(\mathrm{V})^{-1}.$$

Observe that $\Sigma_{AB}(\mathrm{V})$ and $S_{AB}(\mathrm{V})$ are square symmetric matrices which represent the properties of the projection of the data onto the space spanned over V. We put

$$\phi_{AB}(\mathrm{V}) = \frac{1}{(2\pi)^{k/2} \det{}^{1/2}(\Sigma_{AB}(\mathrm{V}))|A||B|},$$

thus

$$\nabla \phi_{AB}(\mathrm{V}) = -\phi_{AB}(\mathrm{V}) \cdot \Sigma_{AB} \cdot \mathrm{V} \cdot S_{AB}(\mathrm{V}).$$

Consequently to compute the final formula, we need the gradient of the function $\mathrm{V} \to \det(\Sigma_{AB}(\mathrm{V}))$, which as one can easily verify, is given by the formula

$$\nabla \det(\Sigma_{AB}(\mathrm{V})) = 2 \det(\mathrm{V}^T \Sigma_{AB} \mathrm{V}) \cdot \Sigma_{AB} \mathrm{V} (\mathrm{V}^T \Sigma_{AB} \mathrm{V})^{-1}. \tag{3}$$

One can also easily check that for

$$\psi^w_{AB}(\mathrm{V}) = \exp(-\tfrac{1}{2}\|\mathrm{V}^T w\|^2_{\Sigma_{AB}(\mathrm{V})}),$$

where w arbitrarily fixed, we get

$$\nabla \psi^w_{AB}(\mathrm{V}) = -\psi^w_{AB}(\mathrm{V}) \cdot (w w^T \mathrm{V} S_{AB}(\mathrm{V}) - \Sigma_{AB} \mathrm{V} S_{AB}(\mathrm{V}) \mathrm{V}^T w w^T \mathrm{V} S_{AB}(\mathrm{V})).$$

To present the final form for the gradient of $\mathrm{D}_{\mathrm{CS}}(\mathrm{V})$ we need the gradient of the cross information potential

$$\mathrm{ip}^{\times}_{AB}(\mathrm{V}) = \phi_{AB}(\mathrm{V}) \sum_{w \in A-B} \psi^w_{AB}(\mathrm{V}),$$

$$\nabla \mathrm{ip}^{\times}_{AB}(\mathrm{V}) = \phi_{AB}(\mathrm{V}) \sum_{w \in A-B} \nabla \psi^w_{AB}(\mathrm{V}) + \left(\sum_{w \in A-B} \psi^w_{AB}(\mathrm{V}) \right) \cdot \nabla \phi_{AB}(\mathrm{V}).$$

Since

$$\mathrm{D_{CS}}(\mathrm{V}) = \log(\mathrm{ip}^{\times}_{\mathrm{X_+X_+}}(\mathrm{V})) + \log(\mathrm{ip}^{\times}_{\mathrm{X_-X_-}}(\mathrm{V})) - 2\log(\mathrm{ip}^{\times}_{\mathrm{X_+X_-}}(\mathrm{V})),$$

we finally get

$$\begin{aligned}\nabla\mathrm{D_{CS}}(\mathrm{V}) = &\frac{1}{\mathrm{ip}^{\times}_{\mathrm{X_+X_+}}(\mathrm{V})}\nabla\mathrm{ip}^{\times}_{\mathrm{X_+X_+}}(\mathrm{V}) + \frac{1}{\mathrm{ip}^{\times}_{\mathrm{X_-X_-}}(\mathrm{V})}\nabla\mathrm{ip}^{\times}_{\mathrm{X_-X_-}}(\mathrm{V}) \\ &- 2\frac{1}{\mathrm{ip}^{\times}_{\mathrm{X_+X_-}}(\mathrm{V})}\nabla\mathrm{ip}^{\times}_{\mathrm{X_+X_-}}(\mathrm{V}).\end{aligned}$$

Given

$$\begin{aligned}\mathrm{MELM}(\mathrm{V}) &= \mathrm{D_{CS}}(\mathrm{V}) - \|\mathrm{V}^T\mathrm{V} - I\|^2, \\ \nabla\mathrm{MELM}(\mathrm{V}) &= \nabla\mathrm{D_{CS}}(\mathrm{V}) - (4\mathrm{V}\mathrm{V}^T\mathrm{V} - 4\mathrm{V}),\end{aligned}$$

one can run any first-order optimization method to find vectors V spanning k-dimensional subspace $\mathcal{V}$ representing low-dimensional, discriminative manifold of the input space.

As one can notice from the above equations, the computational complexity of both function evaluation and its gradient are quadratic in terms of training set size. For big datasets this can a serious bottleneck. One of the possible solutions is to use approximation schemes for the computation of the Cauchy-Schwarz diviergence, which are known to significantly reduce the computational time without sacrificing the accuracy [10]. One other option is to use analogous of stochastic gradient descent where we define function value on a random sample of $\mathcal{O}(\sqrt{N})$ points (resampled in each iteration) from each class, leading to linear complexity and given that training set is big enough, one can get theoretical guarantees on the quality of approximation [15]. Finally, it is possible to first build a Gaussian Mixture Model (GMM) of each class distribution [17] and perform optimization on such density estimator. Computational complexity would be reduced to constant time per-iteration (due to fixing number of components during clustering) trading speed for accuracy.

5 Experiments

We use ten binary classification datasets from UCI repository [1] and libSVM repository [3], which are briefly summarized in Table 1. These are moderate size problems.

Code was written in Python with the use of scikit-learn, numpy and scipy. Besides MELM we use 8 other linear dimensionality reduction techniques, namely: Principal Component Analysis (PCA), class PCA (cPCA[6]), two ellipsoid PCA (2ePCA[7]), per class PCA (pPCA[8]), Independent Component Analysis (ICA), Factor Analysis (FA), Nonnegative Matrix Factorization (NMF[9]),

[6] cPCA uses sum of each classes covariances, weighted by classes sizes, instead of whole data covariance.

[7] 2ePCA is cPCA without weights, so it is a balanced counterpart.

[8] pPCA uses as V_i the first principal component of ith class.

[9] In order to use NMF we first transform dataset so it does not contain negative values.

Disriminative Learning using Generalized Eigenvectors (GEM [11]). PCA, ICA, NMF and FA are implemented in scikit-learn, cPCA, pPCA and 2ePCA were coded by authors and for GEM we use publically available code[10]. Implementation of MELM as a model compatible with scikit-learn classifiers and transformers is available both in supplementary materials and online[11].

Table 1. Summary of used datasets. N denote number of points, d dimensionality, $|\mathrm{X}_l|$ number of samples with l label, $\bar{m}$ mean density (number of nonzero elements) and d_l^t denotes number of dimensions which we have to include during PCA to keep t of label l variance.

dataset	N	d	$\lvert\mathrm{X}_-\rvert$	$\lvert\mathrm{X}_+\rvert$	$\hat{m}$	$d^{.95}$	$d_-^{.95}$	$d_+^{.95}$
australian	690	14	383	307	0.80	1	2	1
breast-cancer	683	10	444	239	1.00	1	1	1
diabetes	768	8	268	500	0.88	2	2	3
fourclass	862	2	555	307	1.00	2	2	2
german.numer	1000	24	700	300	0.75	3	3	3
heart	270	13	150	120	0.75	3	3	3
ionosphere	351	34	126	225	0.88	24	26	7
liver-disorders	345	6	145	200	1.00	3	3	3
sonar	208	60	111	97	1.00	28	24	24
splice	1000	60	483	517	1.00	55	52	54

In order to estimate how hard to optimize is the MELM objective function we plot in Fig. 3 histograms of $\mathrm{D_{CS}}$ values obtained during 500 random starts for each of the dataset. First, one can easily notice that $\mathrm{D_{CS}}$ have multiple local extrema (see for example *heart* or *liver-disorders* histograms). It also appears that in some of the considered datasets it is not easy to obtain maximum by the use of completely random starting point (see *ionosphere* and *australian* datasets), which suggests that one should probably use some more advanced initialization techniques.

To further investigate how hard it is to find a good solution when selecting maximum of $\mathrm{D_{CS}}$ we estimate the expected value of $\mathrm{D_{CS}}$ after s random starts from matrices $V^{(1)}, \ldots, V^{(s)}$

$$\mathbb{E}[\max_{\mathrm{V}=\mathrm{V}^{(1)},\ldots,\mathrm{V}^{(s)}} \mathrm{D_{CS}}(\text{L-BFGS}(\text{MELM}|\mathrm{V}))].$$

As one can see on Fig. 4 for 8 out of 10 considered datasets one can expect to find the maximum (with 5% error) after just 16 random starts. Obviously this cannot be used as a general heuristics as it is heavily dependent on the dataset size, dimensionality as well as its discriminativness. However, this experiment

[10] forked at http://gist.github.com/lejlot/3ab46c7a249d4f375536
[11] http://github.com/gmum/melm

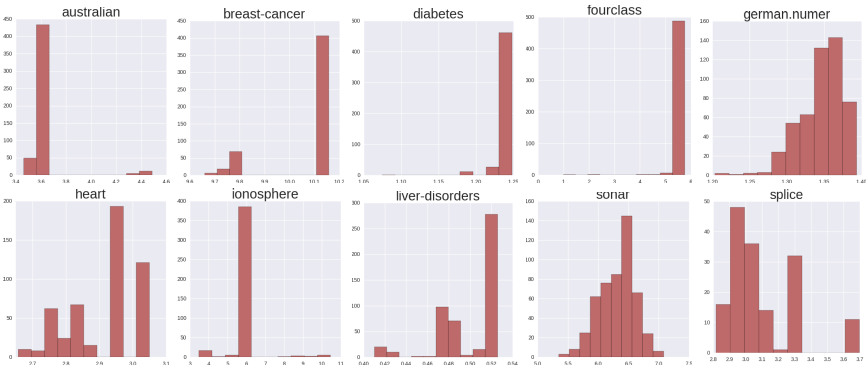

Fig. 3. Histograms of D_{CS} values obtained for each dataset during 500 random starts using L-BFGS.

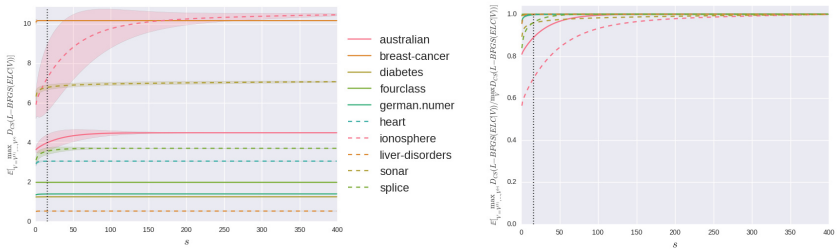

Fig. 4. Expected value of Cauchy-Schwarz Divergence after MELM optimization for s random starts using L-BFGS algorithm (on the left) and its ratio to the maximum obtainable Cauchy-Schwarz Divergence (on the right). Dotted black line shows 16 starts threshold.

shows that for moderate size problems (hundreds to thousands samples with dozens of dimensions) MELM can be relatively easily optimized even though it is a rather complex function with possibly many local maxima.

It is worth noting that truly complex optimization problem is only given by *ionosphere* dataset. One can refer to Table 1 to see that this is a very specific problem where positive class is located in a very low-dimensional linear manifold (approximately 7 dimensional) while the negative class is scattered over nearly 4 times more dimensions.

We check how well MELM behaves when used in a classification pipeline. There are two main reasons for such approach, first if the discriminative manifold is low-dimensional, searching for it may boost the classification accuracy. Second, even if it decreases classification score as compared to non-linear methods applied directly in the input space, the resulting model will be much simpler and more robust. For comparison consider training a RBF SVM in $\mathbb{R}^{60}$ using 1000 data

points. It is a common situation when SVM selects large part of the dataset as the support vectors [16], [18], meaning that the classification of the new point requires roughly $500 \cdot 60 = 30000$ operations. In the same time if we first embed space in a plane and fit RBF SVM there we will build a model with much less support vectors (as the 2D decision boundary generally is not as complex as 60-dimensional one), lets say 100 and consequently we will need $60 \cdot 2 + 2 \cdot 100 = 120 + 200 = 320$ operations, two orders of magnitude faster. Whole pipeline is composed of:

1. Splitting dataset into training $\mathrm{X}_-, \mathrm{X}_+$ and testing $\hat{\mathrm{X}}_-, \hat{\mathrm{X}}_+$.
2. Finding plane embeding matrix $\mathrm{V} \in \mathbb{R}^{d \times 2}$ using tested method.
3. Training a classifier *cl* on $\mathrm{V}^T\mathrm{X}_-, \mathrm{V}^T\mathrm{X}_+$.
4. Testing *cl* on $\mathrm{V}^T\hat{\mathrm{X}}_-, \mathrm{V}^T\hat{\mathrm{X}}_+$.

Table 2 summarizes BAC scores obtained by each method on each of the considered datasets in 5-fold cross validation. For the classifier module we used SVM RBF, KNN and KDE-based density classification. Each of them was fitted using internal cross-validation to find the best parameters. GEM and MELM γ hyperperameters were also fitted. Reported results come from the best classifier.

Table 2. Comparison of 2-dimensional reduction followed by the classifier. I stands for Identity, meaning that we simply trained classifiers directly on the raw data, without any dimensionality reduction. Bold values indicate the best score obtained across all dimensionality reduction pipelines. If the classifier trained on raw data is better than any of the reduced models than its score is also bolded.

	MELM	FA	ICA	GEM	NMF	2ePCA	cPCA	PCA	pPCA	I
australian	**0.866**	0.847	0.829	0.791	0.817	0.764	0.756	0.825	0.769	0.860
breast-cancer	**0.976**	0.973	**0.976**	0.930	**0.976**	0.966	0.967	**0.976**	0.961	0.966
diabetes	**0.744**	0.682	0.705	0.637	0.704	0.689	0.695	0.705	0.646	0.728
fourclass	**1.0**	0.720	**1.0**	**1.0**	0.999	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
german.numer	**0.705**	0.588	0.648	0.653	0.63	0.588	0.602	0.650	0.619	**0.728**
heart	**0.831**	0.792	0.818	0.675	0.811	0.793	0.782	0.817	0.787	**0.837**
ionosphere	**0.892**	0.794	0.757	0.763	0.799	0.783	0.780	0.757	0.826	**0.944**
liver-disorders	**0.710**	0.546	0.545	0.681	0.553	0.531	0.548	0.531	0.557	0.705
sonar	0.766	0.558	0.600	**0.889**	0.657	0.593	0.575	0.600	0.676	0.862
splice	**0.862**	0.718	0.697	0.799	0.691	0.686	0.686	0.697	0.694	**0.887**

In four datasets, MELM based embeding led to the construction of better classifier than both other dimensionality reduction techniques as well as training models on raw data. This suggests that for these datasets the discriminative manifold is truly at most 2-dimensional. At the same time in nearly all (besides *sonar*) datasets the pipeline consisting of MELM yielded significantly better classification results than any other embeding considered.

One of the main applications of MELM is to visualize the dataset through linear projection in such a way that classes do not overlap. One can see comparisons of *heart* dataset projections using all considered approaches in Fig. 5. As one can notice our method finds plane projection where classes are nearly perfectly discriminated. Interestingly, this separation is only obtainable in two dimensions, as neither marginal distributions nor any other one-dimensional projection can construct such separation.

While visual inspection is crucial for such tasks, to truly compare competetive methods we need some metric to measure quality of the visualization. In order to do so, we propose to assign a *visual separability* score as the mean BAC score over three considered classifiers (SVM RBF, KNN, KDE) trained and tested in 5-fold cross validation of the projected data. The only difference between this test and the previous one is that we use whole data to find a projection (so each projection technique uses all datapoints) and only further visualization testing is performed using train-test splits. This way we can capture "how easy to discriminate are points in this projection" rather than "how useful for data discrimination is using the projection". Experiments are repeated using various random subsets of samples and mean results are reported.

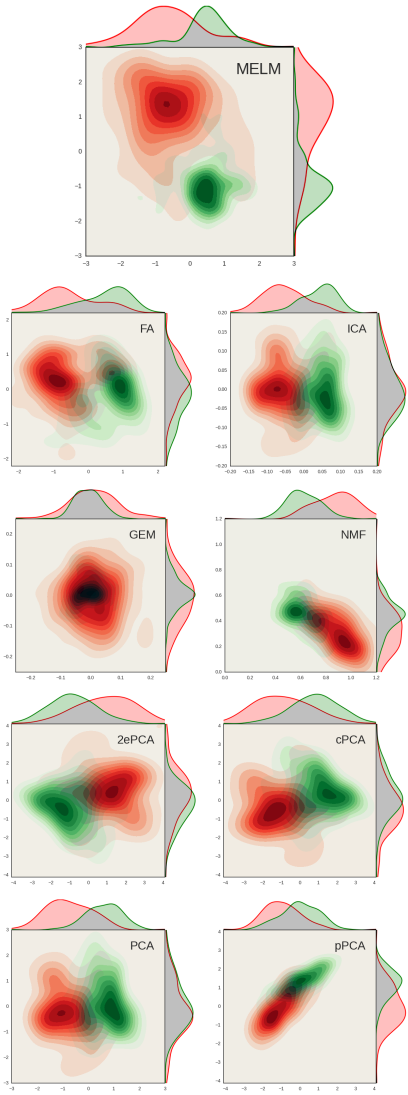

Fig. 5. Comparison of heart dataset 2D projections by analyzed methods. Visualization uses kernel density estimation.

During these experiments MELM achieved essentially better scores than any other tested method (see Table 3). Solutions were about 10% better under our metric and this difference is consistent over all considered datasets. In other words MELM finds two-dimensional representations of our data using just linear projection where classes overlap to a

Table 3. Comparison of 2-dimensional projections discriminativeness.

	MELM	FA	ICA	GEM	NMF	2ePCA	cPCA	PCA	pPCA
australian	**0.888**	0.856	0.845	0.792	0.838	0.782	0.781	0.845	0.792
breast-cancer	**0.985**	0.979	0.979	0.942	0.979	0.967	0.969	0.978	0.965
diabetes	**0.806**	0.732	0.737	0.691	0.737	0.734	0.733	0.734	0.697
fourclass	**0.988**	0.665	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**
german.numer	**0.819**	0.640	0.687	0.686	0.672	0.665	0.657	0.686	0.692
heart	**0.918**	0.822	0.834	0.751	0.839	0.787	0.783	0.833	0.799
ionosphere	**0.990**	0.810	0.798	0.763	0.849	0.804	0.814	0.798	0.863
liver-disorders	**0.763**	0.682	0.659	0.707	0.698	0.691	0.676	0.688	0.715
sonar	**0.996**	0.714	0.717	0.892	0.729	0.702	0.709	0.717	0.735
splice	**0.927**	0.738	0.724	0.829	0.716	0.717	0.718	0.723	0.742

significantly smaller degree than using PCA, cPCA, 2ePCA, pPCA, ICA, NMF, FA or GEM. It is also worth noting that Factor Analysis, as the only method which does not require orthogonality of resulting projection vectors did a really bad job while working with fourclass data even though these samples are just two-dimensional.

As stated before, MELM is best suited for low-dimensional embedings and one of its main applications is supervised data visualization. However in general one can be interested in higher dimensional subspaces. During preliminary studies we tested model behavor up to $k = 5$ and results were similar to the one reported in this paper (when compared to the same methods, with analogous k). It is worth noting that methods like PCA also use a density estimator - one big Gaussian fitted through maximum likelihood estimation. Consequently even though from theoretical point of view MELM should not be used for $k > 5$ (due to the curse of dimensionality [15], it works fine as long as one uses good density estimator (such as a well fitted GMM [17]).

6 Conclusions

In this paper we construct Maximum Entropy Linear Manifold (MELM), a method of learning discriminative low-dimensional representation which can be used for both classification purposes as well as a visualization preserving classes separation. Proposed model has important theoretical properties including affine transformations invariance, connections with PCA as well as bounding the expected balanced misclassification error. During evaluation we show that for moderate size problems MELM can be efficiently optimized using simple first-order optimization techniques. Obtained results confirm that such an approach leads to highly discriminative transformation, better than obtained by 8 compared solutions.

Acknowledgments. The work has been partially financed by National Science Centre Poland grant no. 2014/13/B/ST6/01792.

References

1. Bache, K., Lichman, M.: UCI machine learning repository (2013). http://archive.ics.uci.edu/ml
2. Bhatia, R.: Matrix analysis, vol. 169. Springer Science & Business Media (1997)
3. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) **2**(3), 27 (2011)
4. Cover, T.M., Thomas, J.A.: Elements of information theory, 2nd edn. Willey-Interscience, NJ (2006)
5. Czarnecki, W.M.: On the consistency of multithreshold entropy linear classifier. Schedae Informaticae (2015)
6. Czarnecki, W.M., Tabor, J.: Multithreshold entropy linear classifier: Theory and applications. Expert Systems with Applications (2015)
7. Geng, Q., Wright, J.: On the local correctness of 1-minimization for dictionary learning. In: 2014 IEEE International Symposium on Information Theory (ISIT), pp. 3180–3184. IEEE (2014)
8. Goodfellow, I.J., et al.: Challenges in representation learning: a report on three machine learning contests. In: Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M. (eds.) ICONIP 2013, Part III. LNCS, vol. 8228, pp. 117–124. Springer, Heidelberg (2013)
9. Hinton, G., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7), 1527–1554 (2006)
10. Jozefowicz, R., Czarnecki, W.M.: Fast optimization of multithreshold entropy linear classifier (2015). arXiv preprint arXiv:1504.04739
11. Karampatziakis, N., Mineiro, P.: Discriminative features via generalized eigenvectors. In: Proceedings of the 31st International Conference on Machine Learning (ICML 2014), pp. 494–502 (2014)
12. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems (NIPS 2014), pp. 2177–2185 (2014)
13. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 689–696. ACM (2009)
14. Principe, J.C., Xu, D., Fisher, J.: Information theoretic learning. Unsupervised Adaptive Filtering **1**, 265–319 (2000)
15. Silverman, B.W.: Density estimation for statistics and data analysis, vol. 26. CRC Press (1986)
16. Suykens, J.A., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least squares support vector machines, vol. 4. World Scientific (2002)
17. Tabor, J., Spurek, P.: Cross-entropy clustering. Pattern Recognition **47**(9), 3046–3059 (2014)
18. Wang, L.: Support Vector Machines: theory and applications, vol. 177. Springer Science & Business Media (2005)