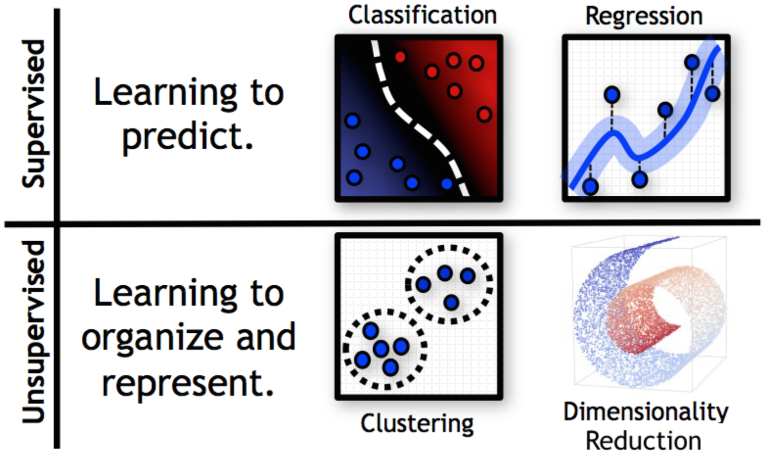
COMPSCI 589 Lecture 18: Linear Dimensionality Reduction and SVD

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).
Created with support from National Science Foundation Award# IIS-1350522.

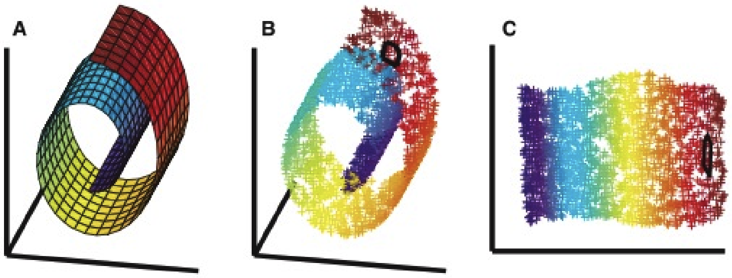
Machine Learning Tasks



The Dimensionality Reduction Task

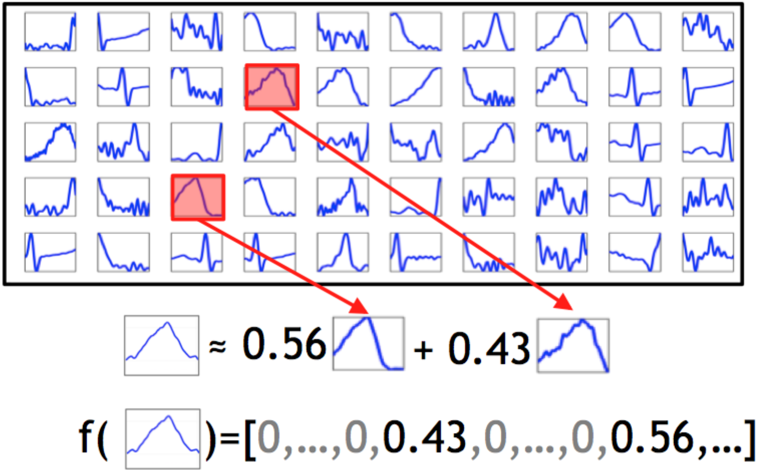
Definition: The Dimensionality Reduction Task

Given a collection of feature vectors $\mathbf{x}_i \in \mathbb{R}^D$, map the feature vectors into a lower dimensional space $\mathbf{z}_i \in \mathbb{R}^K$ where $K < D$ while preserving certain properties of the data.

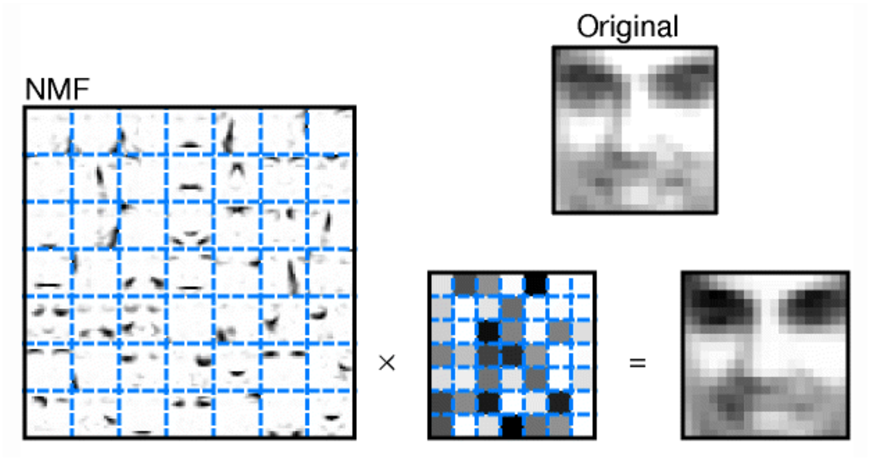


high-dim distribution   high-dim samples   estimated manifold

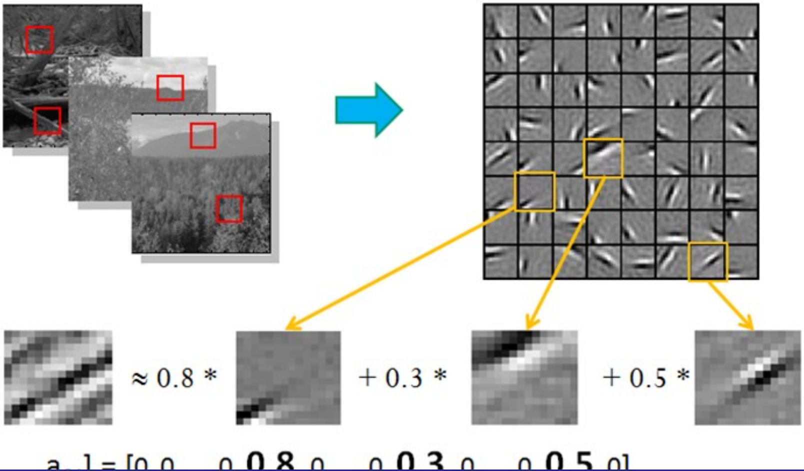
Example: Representing Time Series



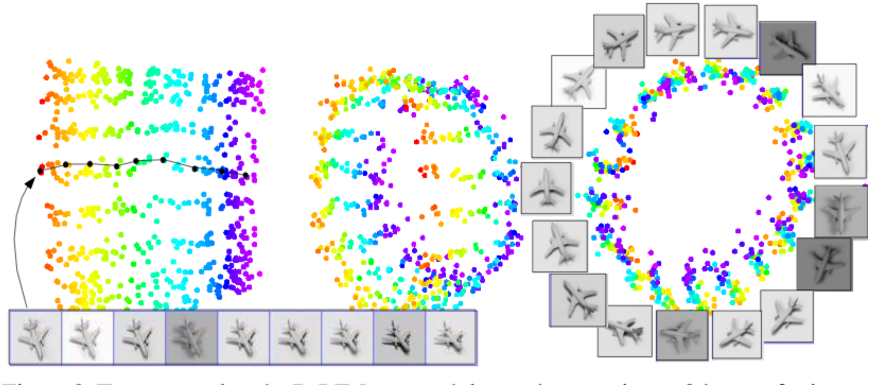
Example: Learning Face Parts



Example: Representing Natural Image Patches



Example: Image Embedding



Applications

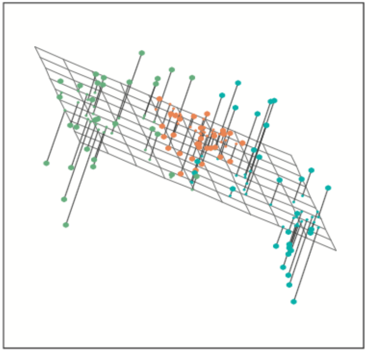
- Can be used as a pre-processing step to enable more accurate classification and regression on manifold data.
- Very low dimensional embeddings (ie: K=2,3) can be used to visualize complex manifold-structured data as in the previous example.
- Can be used to de-noise data by projecting to lower-dimensional space and then projecting back to the feature space.

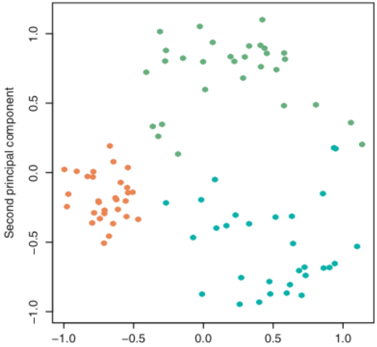
Dimensionality Reduction vs Feature Selection

- The goal of feature selection is to remove features that are not informative with respect to the class label. This obviously reduces the dimensionality of the feature space.
- Dimensionality reduction can be used to compress the feature space for manifold structured data even when there is information in each of the feature dimensions so that none can be discarded.
- Another important property of dimensionality reduction is that it is unsupervised. It will attempt to preserve structure in the data that could be useful for a range of supervised problems, not a specific problem.
- Unlike feature selection, which is a supervised task, dimensionality reduction can sometimes compress out structure useful to a particular supervised task thereby increasing error.

Linear Dimensionality Reduction

- The simplest dimensionality reduction methods assume that the observed high dimensional data vectors $\mathbf{x}_i \in \mathbb{R}^D$ lie on a K-dimensional linear manifold within $\mathbb{R}^D$.





Linear Dimensionality Reduction

- Mathematically, the linear sub-space assumption can be written as follows:

$$\mathbf{x}_{id} = \sum_{k=1}^{K} z_{ik} b_{kd}$$

where $\mathbf{b}_k = [b_{k1}, ..., b_{kD}]$ for $k = 1, ..., K$ are a set of basis vectors describing a K-dimensional linear sub-space of $\mathcal{R}^D$ and z_{ki} are a set of real-valued weights on those basis vectors.

Connection to Linear Regression

- This expression is exactly linear regression where x_{id} is the target, z_{ik} are the weights, and $\mathbf{b}_{kd}$ for each k are the features.

$$\mathbf{x}_{id} = \sum_{k=1}^{K} z_{ik} b_{kd}$$

- This observation is also true if we swap the roles of the weights and the features.
- However, unlike the linear regression case, we only know what corresponds to the targets. We must learn both the features and the weights.

Matrix Form

- If we let $\mathbf{X}$ be the data matrix $\mathbf{X}_{id} = x_{id}$, $\mathbf{Z}$ be a matrix where $\mathbf{Z}_{ik} = z_{ik}$, and $\mathbf{B}$ be a matrix where $\mathbf{B}_{kd} = b_{kd}$, we can express $\mathbf{X}$ as follows:

$$\mathbf{X} = \mathbf{Z} \times \mathbf{B}$$

- $\mathbf{Z} \in \mathbb{R}^{N \times K}$ is often referred to as the factor loading matrix while $\mathbf{B} \in \mathbb{R}^{K \times D}$ are referred to as the latent factors by analogy to regression.

Observation Noise

- Most real world data will be subject to noise. If we assume that $\epsilon \in \mathbb{R}^{N \times D}$ is a matrix of noise values from some probability distribution, we have:

$$\mathbf{X} = \mathbf{Z} \times \mathbf{B} + \epsilon$$

Learning

- The learning problem for linear dimensionality reduction is to estimate values for both **Z** and **B** given only the noisy observations **X**.
- One possible learning criteria is to minimize the sum of squared errors when reconstructing **X** from **Z** and **B**. This leads to:

$$\arg\min_{\mathbf{Z},\mathbf{B}} ||\mathbf{X} - \mathbf{Z}\mathbf{B}||_F$$

where $||\mathbf{A}||_F$ is the Frobenius norm of matrix **A** (the sum of the squares of all matrix entries).

Learning Algorithm

- Not surprisingly, we can obtain a solution to this learning problem by leveraging the OLS solution to linear regression. The algorithm is often referred to as Alternating Least Squares or ALS.
- Starting from a random initialization, ALS iterates between assuming **Z** are known features and optimizing **B** as the unknown weights, and assuming that **B** are the known features and optimizing **Z** as the unknown weights:

$$\mathbf{B} \leftarrow (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T X$$
$$\mathbf{Z}^T \leftarrow (\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}X^T$$

Lack of Uniqueness of Optimal Parameters

- Suppose we run the ALS algorithm to convergence and obtain estimates for $\mathbf{Z}_*$ and $\mathbf{B}_*$ such that:

$$c = ||\mathbf{X} - \mathbf{Z}_*\mathbf{B}_*||_F$$

- Note that if we let $\mathbf{R} \in \mathbb{R}^{K\times K}$ be an arbitrary $K \times K$ invertible matrix, then we obtain exactly the same value c of the objective function for the alternate parameters $\tilde{\mathbf{Z}} = \mathbf{Z}_*\mathbf{R}$ and $\tilde{\mathbf{B}} = \mathbf{R}^{-1}\mathbf{B}_*$:

$$\begin{aligned} c &= ||\mathbf{X} - \mathbf{Z}_*(I)\mathbf{B}_*||_F \\ &= ||\mathbf{X} - \mathbf{Z}(\mathbf{R}\mathbf{R}^{-1})\mathbf{B}||_F \\ &= ||\mathbf{X} - \tilde{\mathbf{Z}}\tilde{\mathbf{B}}||_F \end{aligned}$$

Singular Value Decomposition

- Interestingly, this optimization problem has a continuous subspace of optimal solutions that all obtain the same global minimum value of the objective function.
- Each optimal solution is simply a different representation the same linear subspace.
- We can pick a unique representation for the subspace by specifying additional criteria. Classical Rank-K Singular Value Decomposition (K-SVD) corresponds to the following restriction:

$$\arg\min_{\mathbf{U},\mathbf{S},\mathbf{V}} ||\mathbf{X} - \mathbf{U}\mathbf{S}\mathbf{V}^T||_F$$

where S is a $K \times K$ diagonal matrix with positive elements, $\mathbf{U}$ is an $N \times K$ matrix such that $\mathbf{U}^T\mathbf{U} = I$, and V is a DxK matrix such that $\mathbf{V}^T\mathbf{V} = I$

Trade-Offs

- Minimum Frobenius norm linear dimensionality reduction is nice because it will find the globally optimal linear sub-space, and SVD provides a unique representation for this sub-space.
- The basic ALS algorithm scales as $O(K^3 + D^3)$ per pair of iterations due to the matrix inverses, which is not suitable for high dimensions. The full SVD algorithm scales as $O(min(DN^2, ND^2))$, which also isn't scalable to large data. However, fast approximations exist for both problems.
- A significant limitation of linear dimensionality reduction is that it can fail to achieve a useful compression of the data if the underlying manifold is not actually linear.
- As with clustering, the rank K of the latent sub-space is a free parameter.