CS 189 Introduction to Machine Learning
Summer 2018 DIS5

1 Motivation: Dimensionality reduction

In this problem sheet we explore the motivation for general dimensionality reduction in machine learning and derive from first principles why projection on the first eigenvectors of the covariance matrix of the data has some favorable properties. A deeper understanding on the advantages of PCA and other dimensionality reduction methods is conveyed in the homework.

In general, we assume the following scenario: Suppose we are given n points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathbb{R}^d$ and the dimension of the feature vectors is d (very big, like 10^3). By dimensionality reduction, we refer to a mapping $\psi : \mathbb{R}^d \mapsto \mathbb{R}^k$ that maps vectors from $\mathbb{R}^d$ to $\mathbb{R}^k$ with $k \ll d$.

(a) (Motivation) Given n feature vectors of d dimensions, in which regimes of n, d and why would you want to reduce the dimensionality in practical machine learning applications? Think about the concept of regularization studied extensively in the past few weeks.

Solution:

In general:

There are two big motivations for dimensionality reduction. First, there is the simple one coming from the bias/variance tradeoff. You have seen that every feature essentially brings its own variance to the problem that scales like $\frac{1}{n}$. So, when there are a lot of potential features that we could use, the optimal number of features that we do use might be fewer. Hence, dimensionality reduction. The second is computational. Processing the data is costly. If a system of linear equations has to be solved, the complexity is super-linear in the number of variables. Cutting down the number of features cuts down the number of variables that we need to solve for.

Reducing dimension here seems to be win-win. Though for computing the reduction, we need to be smart, else there are no computational savings.

In slightly more detail: There are three basic regimes $n \ll d$ (high dimensional data regime = underdetermined), $n >\sim d$ and $n \gg d$ (overdetermined):

- When $n \ll d$ (underdetermined system), we use regularization (or model selection) assuming low rank structure to avoid the overfitting that would naturally occur.
- When $n \gg d$ there might appear to be enough data to get good estimates of the variables. However, even here, we can potentially get some advantages from dimensionality reduction (and other forms or regularization) if there is lower-dimensional models have good approximation error. (You saw this with polynomials fitting the exponential function.)

Another idea here (not explored that much in 189) is to do “sketching” to reduce the n samples to something that is fewer since there is plenty of data.

- Dimensionality reduction is in general most helpful in cases when $n \ll d$ or $n >\sim d$ and the problem has lower effective dimension (otherwise $n \ll d$ obviously wouldn't give you a reasonable estimator in the first place).

(b) (Computational aspect) Revisit this in the context of linear regression. What is the computational complexity of performing a linear regression of n data points in d dimensions with $n > d$ (say by solving the normal equations when $\mathbf{X}^\top\mathbf{X}$ is invertible)? If the projection was given to you for free, approximately how many operations would you save if you reduced the dimension from $d = 10^3$ to $d = 10$?

Solution:

Solving linear regression requires $O(nd^2)$ for $n > d$ (and $O(n^2d)$ for $n < d$). We discuss the case $n > d$ for which the computational complexity can be seen by considering the solution via normal equations, which is

$$\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}. \tag{1}$$

Forming the matrix $\mathbf{X}^\top\mathbf{X}$ costs $O(nd^2)$ and inverting it costs $O(d^3)$. Forming $\mathbf{X}^\top\mathbf{y}$ costs $O(nd)$ and the final matrix multiplication of the two $d \times d$ matrices costs $O(d^3)$. The total cost is therefore $O(d^2(n+d))$ which in the case $d < n$ is equal to $O(nd^2)$. An alternative approach to compute the computational complexity of L.R. is via SVD computation.

Reducing the dimension by a factor of 100 therefore gives a 10,000 fold speedup for solving the linear regression, which is quite substantial.

(c) (Brainstorming possible projections) What are some naive and less naive dimensionality reduction methods you could think of and what would their computational costs be (approximately)? Which methods require either previous data of the same distribution or the data itself, which are projection methods are independent of the data?

Solution:

Brainstorm methods: Throwing elements away, Random matrix (independent Gaussian or Bernoulli entries), Randomly subsampled Fourier or Hadamard matrices, PCA. If curious, more about these topics can be learned from `https://arxiv.org/pdf/1104.5557.pdf`.

Complexities:

- Randomized matrix: $O(dkn)$, data independent
- Randomized FT, Hadamard would take $O(nd \log k)$, data independent
- PCA (SVD + projection step) $O(d^2n)$ (for full) vs. $O(dkn)$ (approximate) + projection step $O(dkn)$, needs raw data to even compute the dimensions.

Exact SVD does not give an overall computational gain, since the computation for the projection matrix itself already takes $O(nd^2)$ computations. For randomized methods, $k > O(\log n)$ are needed (see homework), random projections itself is already helpful ending up with $O(dn \log n)$. Approximate SVD methods get us improvements on the same scale. Using randomized FT or Hadamard instead gets us even smaller complexity $O(dn \log \log n)$.

(d) (Desiderata for projection) With the above goals in mind, let's think about a concrete scenario where we want to do binary classification. What are intuitively good properties of the data which makes it easy or possible for a classification algorithm to work well?

In the homework you will prove that PCA and random projections are guaranteed to preserve some of these properties.

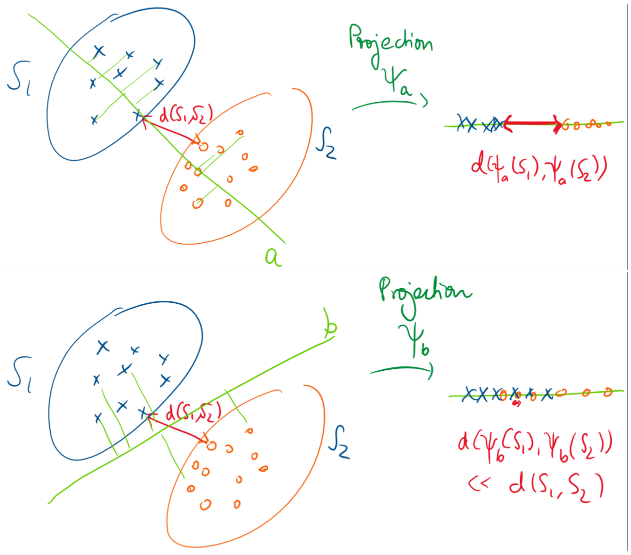
Solution: Note that many notions vaguely mentioned here will be rigorously discussed in much more detail in later lectures, discussions, and homeworks. This is brought up here to motivate good properties of dimensionality reduction techniques.

Intuitively speaking, if the minimum separation between samples belonging to two different classes is large, the classification task is "easy" in the sense of robust to noise. Mathematically, we may define the distance between sets as the minimum pairwise distance

$$d^2(S_1, S_2) := \min_{i \in S_1, j \in S_2} \|x_i - x_j\|_2^2.$$

When the two sets are linearly separable, this notion is similar to the notion of gap and margin which we will cover later and is most commonly introduced in the context of support vector machines.

Projections which preserve Euclidean properties (like angles and norms) of the space in the reduced dimensional space are thus helpful to keep an originally "easy" classification problem, still "easy". Consider the following two dimensional sets for the two classes (crosses vs. circles) and two candidate projection directions.



Projection ψ_a is "better" as it keeps both sets separated and the minimum distance also does not shrink very much in value.

In order to preserve minimum pairwise distance, it suffices if we preserve all pairwise distances (which you prove in homework). Preserving pairwise distances formally means, i.e. that

$$(1-\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\|^2 \leq (1+\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad \text{for all } i, j \in [N],$$

where we use $[N]$ as shorthand for all the indices of the data points.

2 Derivation of PCA

PCA is often used as a tool in data visualization and reduction of computation load and noise. PCA can be done by eigenvalue decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after removing the mean from the data matrix for each feature/column. In this question we will derive PCA. There are two equivalent perspectives to understand PCA. PCA aims to either find

- the directions of maximum variance, or
- the projections of minimum reconstruction error

given a dataset.

(a) In the first part, we will derive PCA from the perspective of *maximum variance*. You want the line such that projecting your data onto this line will retain the maximum amount of information, i.e., variance. Assuming that the feature matrix $\mathbf{X}$ has zero mean across each of its column, we can formulate the optimization problem as

$$\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \sum_{i=1}^{n} \left(\mathbf{x}_i{}^\top \mathbf{w}\right)^2 = \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} \tag{2}$$

where $\mathbf{x}_i$ is the feature of ith sample, i.e., the ith row of the matrix X.

Show that the maximizer for this problem is equal to the eigenvector $\mathbf{v}_1$ that corresponds to the largest eigenvalue λ_1 of matrix $\mathbf{X}^\top\mathbf{X}$. Also show that optimal value of this problem is equal to λ_1.

Solution:

Background: First let us make clear which quantity we are maximizing and what its interpretation is. When we have a set of points $S = \{\mathbf{x}_i\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^d$, what does the term variance even mean? Recall that for random vectors we have the covariance matrix $\mathbf{\Sigma} = \mathbb{E}(\mathbf{x} - \mathbb{E}(\mathbf{x}))(\mathbf{x} - \mathbb{E}(\mathbf{x}))^\top$. The expectation is taken over the distribution of $\mathbf{x}$. Now there are two questions that arise

- What is the distribution in the case when we have a set of observed samples?
- Given the covariance matrix, what is a scalar variance quantity as a function of that matrix?

With respect to the distribution: On the set of points S we can always define the uniform distribution with $P(\mathbf{x}) = \frac{1}{n}$ if $\mathbf{x} = \mathbf{x}_i$ for some i and zero elsewhere. This is equivalent to the probability of observing x when we draw a random vector from the set S uniformly. This probability mass function corresponds to what we call *the empirical distribution*. This term is especially meaningful when the covariate vectors $\mathbf{x}$ are drawn from a true underlying distribution - in which case this empirical distribution is "close" to the underlying one. The covariance matrix of a set of points is taken over this distribution which is thus defined as

$$\mathbf{\Sigma} = \mathbb{E}(\mathbf{x} - \mathbb{E}(\mathbf{x}))(\mathbf{x} - \mathbb{E}(\mathbf{x}))^\top = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$$

When $\bar{\mathbf{x}} = 0$ (i.e. we have subtracted the mean in our samples), we obtain $\mathbf{\Sigma} = \mathbf{X}^\top\mathbf{X}$ (revisit sum of outer product representation of matrix-matrix multiplication if the last step is not clear).

With respect to variance measuring: Given a Gaussian-style probability distribution over $\mathbf{x}$ we want to capture the total "amount of randomness" that exists in the system. The quantity $\operatorname{tr}(\mathbf{\Sigma})$ turns out to be a reasonable choice, because of the following fact which has been covered in lecture: If a random vector $\mathbf{x}$ has covariance $\mathbf{\Sigma} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, then $\mathbf{z} := \mathbf{V}^\top\mathbf{x}$ has covariance $\mathbf{\Lambda}$ and all entries of $\mathbf{z}$ are i.i.d. scalar random variables with $\mathbf{z}_i$ having variance λ_i. Since each element of $\mathbf{z}$ therefore contributes λ_i noise to the model independently from each other, $\operatorname{tr}\mathbf{\Sigma} = \sum_{i=1}^{n}\lambda_i$ represents the total noise introduced. This is the *variance* that we refer to when dealing with sets of points in $d > 1$ dimensions.

Solution to the problem: We start by invoking the spectral decomposition of $\mathbf{X}^\top\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, which is a symmetric positive semi-definite matrix.

$$\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} = \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top\mathbf{w} = \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} (\mathbf{V}^\top\mathbf{w})^\top\mathbf{\Lambda}\mathbf{V}^\top\mathbf{w} \tag{3}$$

Here is an aside: note through this one line proof that left-multiplying a vector by an orthogonal (or rotation) matrix preserves the length of the vector:

$$\|\mathbf{V}^\top\mathbf{w}\|_2 = \sqrt{(\mathbf{V}^\top\mathbf{w})^\top(\mathbf{V}^\top\mathbf{w})} = \sqrt{\mathbf{w}^\top\mathbf{V}\mathbf{V}^\top\mathbf{w}} = \sqrt{\mathbf{w}^\top\mathbf{w}} = \|\mathbf{w}\|_2$$

Define a new variable $\mathbf{z} = \mathbf{V}^\top\mathbf{w}$, and maximize over this variable. Note that because $\mathbf{V}$ is invertible, there is a one to one mapping between $\mathbf{w}$ and $\mathbf{z}$. Also note that the constraint is the same because the length of the vector $\mathbf{w}$ does not change when multiplied by an orthogonal matrix.

$$\max_{\mathbf{z}:\|\mathbf{z}\|_2=1} \mathbf{z}^\top\mathbf{\Lambda}\mathbf{z} = \max_{\mathbf{z}:\|\mathbf{z}\|_2=1} \sum_{i=1}^{d}\lambda_i z_i^2$$

From this new formulation, it is obvious to see that we can maximize this by throwing all of our eggs into one basket and setting $z_i^* = 1$ if i is the index of the largest eigenvalue, and $z_i^* = 0$ otherwise. Thus,

$$\mathbf{z}^* = \mathbf{V}^\top\mathbf{w}^* \implies \mathbf{w}^* = \mathbf{V}\mathbf{z}^* = \mathbf{v}_1$$

where $\mathbf{v}_1$ is the ”principle” eigenvector, and corresponds to λ_1. Plugging this into the objective function, we see that the optimal value is λ_1.

(b) Let us call the solution of the first part $\mathbf{w}_1$. Next, we will use a *greedy procedure* to find the ith component of PCA by doing the following optimization

$$\begin{array}{ll} \text{maximize} & {\mathbf{w}_i}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}_i \\ \text{subject to} & {\mathbf{w}_i}^\top \mathbf{w}_i = 1 \\ & {\mathbf{w}_i}^\top \mathbf{w}_j = 0 \quad \forall j < i. \end{array} \tag{4}$$

Show that the maximizer for this problem is equal to the eigenvector $\mathbf{v}_i$ that corresponds to the ith eigenvalue λ_i of matrix $\mathbf{X}^\top \mathbf{X}$. Also show that optimal value of this problem is equal to λ_i.

Solution: From the previous part, it is obvious to see that we can maximize this by throwing all of our eggs into one basket and setting $z_i^* = 1$ if i is the index of ith largest eigenvalue and others to 0. Plugging this into the objective function, we see that the optimal value is λ_i.

(c) Show that the previous *greedy procedure* finds the global maximum, namely for any $k < d$, $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k$ is the solution of the following maximization problem

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{k} {\mathbf{w}_i}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}_i \\ \text{subject to} & {\mathbf{w}_i}^\top \mathbf{w}_i = 1 \\ & {\mathbf{w}_i}^\top \mathbf{w}_j = 0 \quad \forall i \neq j. \end{array} \tag{5}$$

Solution: It is sufficient to prove that the maximum variance has upper bound $\sum_{i=1}^{k} \lambda_i$, since this was achieve by the greedy algorithm. For any k orthonormal vectors $\mathbf{w}_i$, variance in this plane is

$$\sum_{i=1}^{k} {\mathbf{w}_i}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}_i = \sum_{i=1}^{k} {\mathbf{w}_i}^\top \mathbf{V}^\top \mathbf{\Lambda} \mathbf{V} \mathbf{w}_i = \sum_{j=1}^{d} \lambda_j \left(\sum_{i=1}^{k} [\mathbf{V}\mathbf{w}_i]_j^2 \right).$$

It is good to notice that the $\mathbf{V}\mathbf{w}_i$ are themselves orthonormal by the properties of $\mathbf{V}$ from the spectral theorem for symmetric matrices. Consequently, it is natural to define

$$c_j = \sum_{i=1}^{k} [\mathbf{V}\mathbf{w}_i]_j^2 = \sum_{i=1}^{k} \langle \mathbf{V}\mathbf{w}_i, \mathbf{e}_j \rangle^2,$$

where $\mathbf{e}_j$ is the jth standard basis vector corresponding to the jth coordinate. It is sufficient to prove that $c_j \leq 1$ and $\sum_{j=1}^{d} c_j = k$.

To prove $c_j \leq 1$, we can think that c_j represents the length of $\mathbf{e}_j$ projected to the orthogonal space spanned by $\mathbf{V}\mathbf{w}_i$, which is always smaller or equal to $\|\mathbf{e}_j\| = 1$.

To prove $\sum_{j=1}^{d} c_j = k$, we have

$$\sum_{j=1}^{d} c_j = \sum_{j=1}^{d} \sum_{i=1}^{k} \langle \mathbf{V}\mathbf{w}_i, \mathbf{e}_j \rangle^2 = \sum_{i=1}^{k} \sum_{j=1}^{d} \langle \mathbf{V}\mathbf{w}_i, \mathbf{e}_j \rangle^2 = \sum_{i=1}^{k} \|\mathbf{V}\mathbf{w}_i\|_2^2 = k.$$

Notice that the second to last equality holds by the definition of the squared Euclidean norm — summing up the squares of the coordinates.

(d) Finally, we will show that PCA from the perspective of *minimizing the reconstruction error* from projection, i.e., minimizing the perpendicular distance between the principle component subspace and the data points. Let's say we want to find the best 1D space that minimizes the reconstruction error. The projection of the feature vector $\mathbf{x}$ onto the subspace spanned by a unit vector $\mathbf{w}$ is

$$P_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}\left(\mathbf{x}^\top \mathbf{w}\right). \tag{6}$$

Show that the minimizer $\mathbf{w}$ for the reconstruction error

$$\min_{\mathbf{w}:|\mathbf{w}|=1} \sum_{i=1}^{n} \|\mathbf{x}_i - P_{\mathbf{w}}(\mathbf{x}_i)\|_2^2 \tag{7}$$

is as same as the $\mathbf{w}$ in Equation (2).

Solution: We have

$$\begin{aligned}
&\min_{\mathbf{w}:|\mathbf{w}|=1} \sum_{i=1}^{n} \|\mathbf{x}_i - P_{\mathbf{w}}(\mathbf{x}_i)\|_2^2 && (8)\\
=\ &\min_{\mathbf{w}:|\mathbf{w}|=1} \sum_{i=1}^{n} \left(\|\mathbf{x}_i\|^2 - 2(\mathbf{x}_i - P_{\mathbf{w}}(\mathbf{x}_i))^\top P_{\mathbf{w}}(\mathbf{x}_i) - \|P_{\mathbf{w}}(\mathbf{x}_i)\|^2\right) && (9)\\
=\ &\min_{\mathbf{w}:|\mathbf{w}|=1} \sum_{i=1}^{n} \left(\|\mathbf{x}_i\|^2 - \|P_{\mathbf{w}}(\mathbf{x}_i)\|^2\right) && (10)\\
=\ &\min_{\mathbf{w}:|\mathbf{w}|=1} \sum_{i=1}^{n} \|\mathbf{x}_i\|^2 - \sum_{i=1}^{n} \|(\mathbf{x}_i{}^\top \mathbf{w})\mathbf{w}\|^2 && (11)\\
=\ &\min_{\mathbf{w}:|\mathbf{w}|=1} \sum_{i=1}^{n} \|\mathbf{x}_i\|^2 - \underbrace{\sum_{i=1}^{n} (\mathbf{x}_i{}^\top \mathbf{w})^2}_{\text{Variance Term}}. && (12)
\end{aligned}$$

where the second equality follows from the fact that $P_{\mathbf{w}}(\mathbf{x}_i)$ is an orthogonal projection onto the subspace spanned by $\mathbf{w}$ (and thus the error is orthogonal to any vector in the subspace including $P_{\mathbf{w}}(\mathbf{x}_i)$. Thus we see that minimizing reconstruction error is as same as maximizing variance as what we do in Equation (2). Note that this problem can also be shown in alternative ways which you can find in the notes.