CS 189 Introduction to Machine Learning
Spring 2022 Marvin Zhang

DIS9

1 Derivation of PCA

Assume we are given n training data points $(\mathbf{x}_i, y_i)$. We collect the target values into $\mathbf{y} \in \mathbb{R}^n$, and the inputs into the matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ where the rows are the $d-$dimensional feature vectors $\mathbf{x}_i^\top$ corresponding to each training point. Furthermore, assume that the data has been centered such that $\frac{1}{n}\sum_{i=1}^n \mathbf{x_i} = \mathbf{0}$, $n > d$ and $\mathbf{X}$ has rank d. The covariance matrix is given by

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

When $\bar{\mathbf{x}} = 0$ (i.e., we have subtracted the mean in our samples), we obtain $\Sigma = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$.

(a) Maximum Projected Variance: We would like the vector $\mathbf{w}$ such that projecting your data onto $\mathbf{w}$ will retain the maximum amount of information, i.e., variance. We can formulate the optimization problem as

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

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

Hint: Use the spectral decomposition of Σ and consider reformulating the optimization problem using a new variable.

Solution:

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

$$\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} = \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top\mathbf{V}\Lambda\mathbf{V}^\top\mathbf{w} = \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} (\mathbf{V}^\top\mathbf{w})^\top\Lambda\mathbf{V}^\top\mathbf{w} \tag{2}$$

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.

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

From this new formulation, we can 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. In other words, $\mathbf{z}$ is a one hot vector. Thus,

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

where $\mathbf{v}_1$ is the principal 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 above part $\mathbf{w}_1$. Next, we will use a *greedy procedure* to find the ith component of PCA by doing the following optimization

$$\begin{aligned} &\text{maximize} && \mathbf{w}_i^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}_i \\ &\text{subject to} && \mathbf{w}_i^\top \mathbf{w}_i = 1 \\ & && \mathbf{w}_i^\top \mathbf{w}_j = 0 \quad \forall j < i, \end{aligned} \tag{3}$$

where $\mathbf{w}_j$, $j < i$ are defined recursively using the same maximization procedure above. Show that the maximizer for this problem is equal to the eigenvector $\mathbf{v}_i$ that corresponds to the ith eigenvalue λ_i of Σ. Also show that optimal value of this problem is equal to λ_i.

Solution: We can use the same strategy as from the previous part to write the optimization problem as

$$\begin{aligned} &\text{maximize} && \textstyle\sum_{i=1}^d \lambda_i z_i^2 \\ &\text{subject to} && \|\mathbf{z}\|_2 = 1 \\ & && \mathbf{z}_i^\top \mathbf{z}_j = 0 \quad \forall j < i, \end{aligned} \tag{4}$$

We see that we can maximize this by throwing all of our eggs into one basket and setting $z_k^* = 1$ if k is the index of the ith largest eigenvalue and others to 0. Plugging this into the objective function, we see that the optimal value is λ_i.

2 Kernel PCA

You have seen how to use PCA to do dimensionality reduction by projecting the data to a subspace that captures most of the variability visible in the observed features. The underlying hope is that these directions of variation are also relevant for prediction the quantities of interest.

Standard PCA works well for data that is roughly Gaussian shaped, but many real-world high dimensional datasets have underlying low-dimensional structure that is not well captured by linear subspaces. However, when we lift the raw data into a higher-dimensional feature space by means of a nonlinear transformation, the underlying low-dimensional structure once again can manifest as an approximate subspace. Linear dimensionality reduction can then proceed. As we have seen in class so far, kernels are an alternate way to deal with these kinds of nonlinear patterns without having to explicitly deal with the augmented feature space. This problem asks you to discover how to apply the "kernel trick" to PCA.

Let $\mathbf{X} \in \mathbb{R}^{n \times \ell}$ be the data matrix, where n is the number of samples and ℓ is the dimension of the raw data. Namely, the data matrix contains the data points $\mathbf{x}_j \in \mathbb{R}^{\ell}$ as rows

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} \in \mathbb{R}^{n \times \ell}. \tag{5}$$

(a) **Compute $\mathbf{X}\mathbf{X}^\top$ in terms of the singular value decomposition $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n \times \ell}$ and $\mathbf{V} \in \mathbb{R}^{\ell \times \ell}$.** Notice that $\mathbf{X}\mathbf{X}^\top$ is the matrix of pairwise Euclidean inner products for the data points. **How would you get $\mathbf{U}$ if you only had access to $\mathbf{X}\mathbf{X}^\top$? Solution:** We have $\mathbf{X}^\top\mathbf{X} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ and $\mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^\top$. Notice that the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{X}\mathbf{X}^\top$.

(b) Given a new test point $\mathbf{x}_{test} \in \mathbb{R}^{\ell}$, one central use of PCA is to compute the projection of $\mathbf{x}_{test}$ onto the subspace spanned by the k top singular vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$.

Express the scalar projection $z_j = \mathbf{v}_j^\top \mathbf{x}_{test}$ onto the j-th principal component as a function of the inner products

$$\mathbf{X}\mathbf{x}_{test} = \begin{pmatrix} \langle \mathbf{x}_1, \mathbf{x}_{test} \rangle \\ \vdots \\ \langle \mathbf{x}_n, \mathbf{x}_{test} \rangle \end{pmatrix}. \tag{6}$$

Assume that all diagonal entries of $\mathbf{\Sigma}$ are nonzero and non-increasing, that is $\sigma_1 \geq \sigma_2 \geq \cdots > 0$. Assume that $\ell > n$ and that $rank(\mathbf{X}) = n$.

Hint: Express $\mathbf{V}^\top$ in terms of the singular values $\mathbf{\Sigma}$, the left singular vectors $\mathbf{U}$ and the data matrix $\mathbf{X}$. If you want to use the compact form of the SVD, feel free to do so.

Solution: We begin with the compact form of the SVD, which is: $\mathbf{X} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^\top$, where $r = rank(\mathbf{X})$. If we featurize each datapoint, then we can assume that $\ell >> n$ and $r = rank(\mathbf{X}) = n$ (the design matrix is full row rank).

Thus, $\mathbf{U}_n$ is the full $n \times n$ orthogonal matrix, $\mathbf{\Sigma}_n$ is a $n \times n$ diagonal matrix of non-zero singular values, and $\mathbf{V}_n^\top$ is a $n \times \ell$ matrix the first n eigenvectors (rows) of $\mathbf{V}^\top$.

Multiplying both sides by inverses to isolate $\mathbf{V}_n^\top$, we get $\mathbf{V}_n^\top = \mathbf{\Sigma}_n^{-1}\mathbf{U}_n^\top\mathbf{X}$. Selecting out the j-th row, we have:

$$z_j = \mathbf{v}_j^\top \mathbf{x}_{test} = \sigma_j^{-1}\mathbf{u}_j^\top \mathbf{X}\mathbf{x}_{test}$$

for $j = 1, \ldots, n$.

(c) How would you define kernelized PCA for a general kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ (to replace the Euclidean inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$)? For example, the RBF kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\delta^2}\right)$.

Describe this in terms of a procedure which takes as inputs the training data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^\ell$ and the new test point $\mathbf{x}_{test} \in \mathbb{R}^\ell$, and outputs the analog of the previous part's z_j coordinate in the kernelized PCA setting. You should include how to compute U from the data, as well as how to compute the analog of $\mathbf{X}\mathbf{x}_{test}$ from the previous part.

Invoking the SVD or computing eigenvalues/eigenvectors is fine in your procedure, as long as it is clear what matrix is having its SVD or eigenvalues/eigenvectors computed. The kernel $k(\cdot, \cdot)$ can be used as a black-box function in your procedure as long as it is clear what arguments it is being given.

Solution:

(a) Obtain the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ as eigenvectors from the eigendecomposition of $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top \in \mathbb{R}^{n \times n}$ with $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. This is much faster (and only way for the RBF kernel) than computing the eigenvectors of $\mathbf{\Phi}^\top\mathbf{\Phi} \in \mathbb{R}^{d \times d}$, where d is the lifted feature dimension.

(b) Kernelize the inner products $z_j = \frac{1}{\sigma_j}\mathbf{u}_j^\top \mathbf{X}\mathbf{x}_{test}$ via:

$$z_j = \frac{1}{\sigma_j}\mathbf{u}_j^\top \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_{test}) \\ k(\mathbf{x}_2, \mathbf{x}_{test}) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_{test}) \end{pmatrix} \tag{7}$$

The projected point (with reduced dimension k) is therefore:

$$\mathbf{z}_{test} = \begin{bmatrix} z_1 \\ \vdots \\ z_k \end{bmatrix}$$