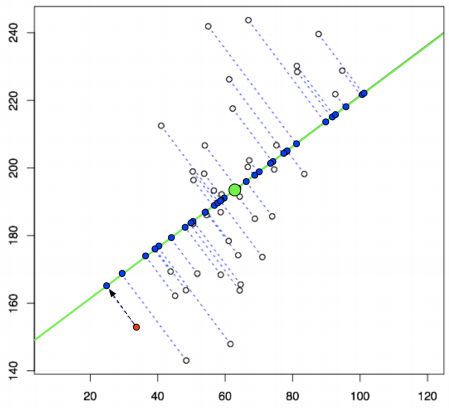
CS 189 Introduction to Machine Learning
Fall 2019 Jennifer Listgarten & Stella Yu

Gorilla 2

1 PCA and Kernel PCA

Let's say that you are given n samples of d-dimensional real world data. That is you have n vectors x_i and we can represent them in an n x d design matrix X. For this problem we will assume that the data are centered: $\frac{1}{n}\sum x_i = 0$.

(a) We learned in class that the objective of PCA is to find some direction(s), w that explain the most data. Another way to view this is by maximizing the variance between projected data points onto this direction w.



In this image, for example, we wish to find a direction w, the linear line, that spreads the projected data points (points on the line) as much as possible.

(a) Consider a point x_i what is its scalar projection onto w?

Solution:

$$x_i^\top \frac{w}{||w||_2}$$

(b) How can we express the entire data set's X scalar project onto w?

Solution:

$$\frac{Xw}{||w||_2}$$

(c) Now we would like to spread these projected points as far as possible. What is the objective function we would like to maximize? Express your answer in the form $\frac{1}{n}v^\top Sv$, where S is a symmetric matrix and v is a unit vector. *Hint: Start with* $Var = E[X - E[X]^2]$

Solution:

$$\begin{aligned}
Var(\tilde{x_1}, \tilde{x_2}, ...\tilde{x_n}) &= \frac{1}{n}\sum_{i=1}^{n}\left(x_i^\top \frac{w}{||w||_2} - \frac{1}{n}\sum_{j}^{n} x_j^\top \frac{w}{||w||_2}\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(x_i^\top \frac{w}{||w||_2} - \left(\frac{w^\top}{||w||_2}\right)\frac{1}{n}\sum_{j=1}^{n} x_j\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(x_i^\top \frac{w}{||w||_2}\right)^2 \\
&= \frac{1}{n}\frac{||Xw||_2^2}{||w||_2^2} \\
&= \frac{1}{n}\frac{w^\top X^\top X w}{||w||_2^2}
\end{aligned}$$

Letting $S = X^\top X$, and $v = \frac{w}{||w||_2}$, we obtain the desired form of the solution.

(d) The resulting expression from the previous part, $v^\top Sv$, is called the *Rayleigh quotient*. Consider having eigenvectors $v_1, \ldots, v_d$ of S with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_d$. Which eigenvector maximizes the *Rayleigh quotient*?

Solution: We choose the eigenvector with the largest eigenvalue, v_d.

(b) Now let's say you wish to augment this data with a feature mapping $\phi : \mathbb{R}^d \to \mathbb{R}^D$. A data point x_i will be mapped to $\phi(x_i)$ and we will call our new n x D design matrix, Φ. We wish to use PCA over this new augmented data. For this question assume that the data is also centered in this feature space.

(a) What is the covariance matrix of our new data? What are its dimensions? Why might this be a difficult for us, or even possibly infeasible? Consider the case when $D >> d$.

Solution: Cov = $\frac{1}{n}\Phi^\top\Phi$ with dimension D x D. Since D can be very large the covariance matrix may be difficult too large to calculate or even impossible if we map to an infinite feature space.

(c) To reduce the computational complexity of this PCA, we choose to instead to kernelize PCA. That is, rather than solving for w as before, we will try to solve for the dual weights α such that $w = \sum_{i=1}^{n}\phi(x_i)\alpha_i = \Phi^\top\alpha$.

(a) Suppose we are given a kernel function $\kappa(x_i, x_j) = \phi(x_i)\cdot\phi(x_j)$, which we can compute quickly. What is our kernel matrix K in terms of Φ?

Solution:

$$K = \Phi\Phi^\top$$

(b) Let us call $S = \sum_i \phi(x_i)\phi(x_i)^\top = \Phi^\top\Phi$, or the scaled by n covariance matrix. Let w be an eigenvector of S with nonzero eigenvalue λ, then in the original PCA our goal was to find the eigenvector w of S corresponding to the largest eigenvalue. Let a corresponding eigenvector of K be α_w. Prove that $w = \Phi^\top\alpha_w$.

Solution:

$$\begin{aligned} Sw &= \lambda w \\ \Phi^\top\Phi w &= \lambda w \\ \Phi\Phi^\top\Phi w &= \Phi\lambda w \\ K\Phi w &= \lambda\Phi w \end{aligned}$$

Let $b = \Phi w$, then we have $Kb = \lambda b$, so that b is an eigenvector of K with eigenvalue λ, and:

$$\begin{aligned} Sw = \Phi^\top\Phi w = \Phi^\top b = \lambda w \\ \longrightarrow w = \frac{\Phi^\top b}{\lambda} \end{aligned}$$

Setting $\alpha_w = b/\lambda$ gives us the claim. Note that if b is an eigenvector of K with eigenvalue λ, then $cb : c \in \mathbb{R}$ is also an eigenvector of K with eigenvalue λ.

(d) Finally, now let's see how we can use the kernel

(a) We encounter a data a new data point x and wish to project its feature representation $\phi(x)$ onto $\frac{w}{||w||}$ where $w = \Phi^\top\alpha_w$ as above. Express the projection in terms of α_w, λ, and $\kappa_x = [\kappa(x_1,x), \kappa(x_2,x), \ldots \kappa(x_n,x)]^\top$.

Solution:

$$\frac{w^\top\phi(x)}{||w||^2} = \frac{\alpha_w^\top\Phi\phi(x)}{\alpha_w^\top\Phi\Phi^\top\alpha_w} = \frac{\alpha_w^\top\kappa_x}{\alpha_w^\top K\alpha_w} = \frac{\alpha_w^\top\kappa_x}{\alpha_w^\top(\lambda\alpha_w)} = \frac{\alpha_w^\top\kappa_x}{\lambda||\alpha_w||^2}$$

(b) Show how the expression from part (a)(c) can be written in terms of α_w and K.

Solution:

$$\frac{1}{n}\frac{w^\top\Phi^\top\Phi w}{||w||^2} = \frac{1}{n}\frac{\alpha_w^\top\Phi\Phi^\top\Phi\Phi^\top\alpha_w}{\alpha_w^\top\Phi\Phi^\top\alpha_w} = \frac{1}{n}\frac{\alpha_w^\top KK\alpha_w}{\alpha_w^\top K\alpha_w} = \frac{1}{n}\frac{\alpha_w^\top K^2\alpha_w}{\alpha_w^\top K\alpha_w}$$

2 Logistical Descent

We can generally break down an optimization problem into 3 parts, a hypothesis function $h(x) = z$, a loss function $L(z,x)$, and an objective function $J(h,L)$. Our goal is to minimize the objective

function and often times we cannot directly find the parameters or weights w. We can however iteratively find them through gradient descent, and in convex cases we can find the global optimum.

Consider the following:

- $s(\gamma) = \frac{1}{1+e^{-\gamma}}$ the logistic function.
- $h(x_i) = s(w^T x_i)$ our hypothesis using logistic regression. Let $s_i = s(w^T x_i)$.
- $L(y_i, x_i) = -y_i \ln z - (1-y_i)\ln(1-z)$ the cross entropy loss function (z is our hypothesis output)
- $\frac{1}{2}\lambda||w||^2 + \frac{1}{n}\sum_n L$ the L_2 regularized objective function

(a) What is the complete objective function that we are trying to minimize?

Solution:

$$J = \frac{1}{2}\lambda||w||^2 - \sum_i^n (y_i \ln s(w^T x_i) + (1-y_i)\ln(1-s(w^T x_i)))$$

$$J = \frac{1}{2}\lambda||w||^2 - \sum_i^n (y_i \ln s_i + (1-y_i)\ln(1-s_i))$$

(b) Now derive the gradient $\nabla_w J$:

Solution: First note:

$$s'(\gamma) = \frac{e^{-\gamma}}{(1+e^{-\gamma})^2} = s(\gamma)(1-s(\gamma))$$

$$\nabla_w s(w^T x_i) = s(w^T x_i)(1-s(w^T x_i))x_i$$

$$\nabla_w s_i = s_i(1-s_i)x_i$$

Let $s_i = s(w^T x_i)$ for simpler notation:

$$\nabla_w J = \nabla_w (\frac{1}{2}\lambda ||w||^2 - \sum_i^n (y_i \ln s(w^T x_i) + (1-y_i)\ln(1-s(w^T x_i))))$$

$$\nabla_w J = \nabla_w (\frac{1}{2}\lambda ||w||^2 - \sum_i^n (y_i \ln s_i + (1-y_i)\ln(1-s_i)))$$

$$\nabla_w J = 2\frac{1}{2}\lambda w - \nabla_w \sum_i^n (y_i \ln s_i + (1-y_i)\ln(1-s_i))$$

$$\nabla_w J = \lambda w - \sum_i^n (\frac{y_i}{s_i}\nabla_w s_i - \frac{1-y_i}{1-s_i}\nabla_w s_i))$$

$$\nabla_w J = \lambda w - \sum_i^n (\frac{y_i}{s_i} - \frac{1-y_i}{1-s_i})\nabla_w s_i$$

$$\nabla_w J = \lambda w - \sum_i^n (\frac{y_i}{s_i} - \frac{1-y_i}{1-s_i}) s_i(1-s_i)x_i$$

$$\nabla_w J = \lambda w - \sum_i^n (y_i - s_i)x_i$$

(c) What are the batch and stochastic weight updates:

- Batch:

  Solution:

  $$w \longleftarrow w + \varepsilon(\lambda w - \sum_i^n (y_i - s_i)x_i)$$

- Stochastic:

  Solution:

  $$w \longleftarrow w + \varepsilon(\lambda w - (y_i - s_i)x_i)$$