CS 189/289A Introduction to Machine Learning
Fall 2024 Jennifer Listgarten, Saeed Saremi

DIS6

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.

(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 $d = 10$?

2 The Minimizing Reconstruction Error Perspective

One perspective on PCA is 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. This is very closely linked to the interpretation of maximizing variance along a vector, which is covered in your homework 4.

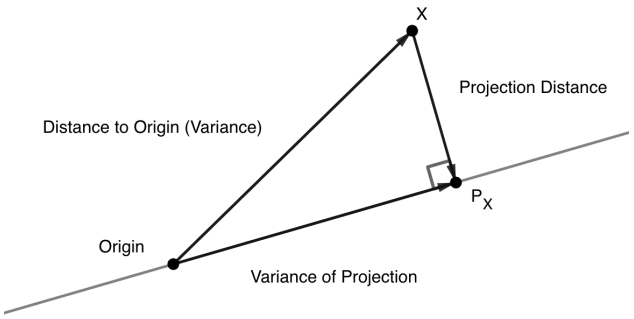
(a) Show the (vector) projection of the feature vector x onto the subspace spanned by a unit vector w is

$$P_w(x) = w\left(x^\top w\right). \tag{1}$$

(b) Now, we want to choose w to minimize the reconstruction error. Show that taking w as the minimizer for the corresponding problem below gives us the same result as before.

$$\min_{w:|w|=1} \sum_{i=1}^{n} \|x_i - P_w(x_i)\|_2^2 \tag{2}$$

150



The above image serves as a useful visualization. Consider mean centered data. A data point has some fixed distance from the origin. We may consider finding a lower dimensional representation as either maximizing the variance of the projecting or minimizing the projection distance. The squared quantities must sum to a constant (the distance to the origin or original variance) thus minimizing one is equivalent to maximizing the other.

3 t-sne? Never heard of her

In this question we'll explore t-sne, which stands for t-distributed stochastic neighborhood embeddings. This is a **nonlinear** dimensionality reduction technique (as opposed to PCA), and is great when dealing with high dimensional data that isn't linear in fashion.

For the purposes of this problem, assume that we have a dataset $X = \{x_1, x_2, \ldots, x_n\}$ where each x_i is d-dimensional. We'll project this down into $Y = \{y_1, y_2, \ldots, y_n\}$, where each y is 2 or 3 dimensions. Y is initially generated randomly (either through a gaussian, or another process, and then iteratively modified through gradient descent).

We'll walk through the process outlined in the original t-sne paper, linked here.

(a) The classical stochastic neighborhood embedding algorithm generates probabilities

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|_2^2/(2\sigma_i^2)\right)}{\sum_{k\neq i}\exp\left(-\|x_i - x_k\|_2^2/(2\sigma_i^2)\right)}.$$

Why did we choose to model probabilities this way? What does the σ_i term represent?

(b) p is not symmetric i.e. $p_{i|j} \neq p_{j|i}$ generally. When does this occur, and what is one way to fix the p matrix so that it is symmetric (assuming we don't change σ_i, σ_j?)

(c) Given lower dimensional projections y_i, t-SNE defines the following:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|_2^2)^{-1}}{\sum_{k\neq i}(1 + \|y_i - y_k\|_2^2)^{-1}}.$$

Contrast this to above where we modeled similarities using gaussians. Why might using a t-distribution be better and what problems can it potentially solve?

(d) y_i is then optimized through gradient descent. In order to do this, we need to define a cost function to optimize. t-SNE uses the cost function

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right).$$

Why is the KL-divergence used here? And what is $\frac{\partial C}{\partial y_i}$?