Video 1: Dimensionality Reduction via Principal Component Analysis

wbg231

December 2022

introduction

- video link
- data with a large number of features is difficult to work with
- we want to reduce the Dimensionality of our data while maintaining the maximal amount of information

linear Dimensionality Reduction

- we mosel data as samples form a d- dimension random vector $\tilde{x}$
- assume we have centered the data (ie it has mean zero)
- for any orthonormal basis $\{b_1...b_d\}$ we can see that

$$\tilde{x} = \Sigma_{i=1}^{n} \tilde{a}[i] b_i : \quad \tilde{a}_i = b_i^T \tilde{x}$$

- so thus we have $\{\tilde{a}_1...\tilde{a}_d\}$ is an equivalent representation of $\tilde{x}$
- can we use some subset of those coefficient to reprint our vector in lower dimension
- so for any subset $\{\tilde{a}_1..\tilde{a}_k\}$ of our coefficients $\{\tilde{a}_1..\tilde{a}_d\}$ we aproximate $\tilde{x}$ as

$$approx_{b_1...b_k}(\tilde{x}) := \Sigma_{i=1}^{k} \tilde{a}[i] b_i : \quad \tilde{a}[i] = b_i^t \tilde{x}$$

how to chose the best subset of orthonormal basis vectors

- we can write $\tilde{x}$ = aproximate + erorr = $\Sigma_{i=1}^{d} \tilde{a}[i] b_i = \Sigma_{i=1}^{k} \tilde{a}[i] b_i + \Sigma_{i=k+1}^{d} \tilde{a}[i] b_i$
- we can express the magnitude of these quantities in terms of there l2 norm

- as the basis vectors are orthonormal we can write $||\tilde{x}||_2^2 = ||\Sigma_{i=1}^k \tilde{a}[i] b_i||_2^2 + ||\text{error}||_2^2 = \Sigma_{i=1}^k \tilde{a}[i]^2 + ||\text{error}||_2^2$
- meaning that $||\text{error}||_2^2| = ||\tilde{x}||_2^2 - \Sigma_{i=1}^k \tilde{a}[i]^2 = ||\tilde{x}||_2^2 - \Sigma_{i=1}^k (b_i^t \tilde{x})^2$
- so we have an expression for the norm of the error and we want to to minimize this naturally
- this is a random quantity so we want to minimize it's expected value

minimize expected value of norm of square loss

- $E[||\text{error}||_2^2|] = E[||\tilde{x}||_2^2 - \Sigma_{i=1}^k (b_i^t \tilde{x})^2] = E[||\tilde{x}||_2^2] - \Sigma_{i=1}^k E[(b_i^t \tilde{x})^2] = E[||\tilde{x}||_2^2] - \Sigma_{i=1}^k E[(b_i \tilde{x} - 0)^2] = E[||\tilde{x}||_2^2] - \Sigma_{i=1}^k E[(b_i \tilde{x} - E[\tilde{x}])^2] = E[||\tilde{x}||_2^2] - \Sigma_{i=1}^k E[(b_i \tilde{x} - E[\tilde{x}])^2] = E[||\tilde{x}||_2^2] - \Sigma_{i=1}^k var[b_i^t \tilde{x}] = E[||\tilde{x}||_2^2] - \Sigma_{i=1}^k b_i^t \Sigma_{\tilde{x}} b_i$
- what choice of $b_1..b_k$ minimize the error?

Principal directions

- we talked about this last time
- let $u_1...u_d$ be the eigenvectors of $\Sigma_{\tilde{x}}$
- $u_1 = argmax_{||a||_2=1} var(a^t \tilde{x})$ is the first Principal direction
- and all other Principal directions are given by $u_k = argmax_{||a||=1, a \perp u_1..a \perp u_{k-1}} var(a^t \tilde{k})$

Principal Component Analysis

- so we can pick our basis vectors are the k first Principal Components and this will minimize the mean l2 norm error
- let $u1_{\cdot}.u_d$ be the eigenvectors of covariance matrix $\Sigma_{\tilde{x}}$
- $\{u_1...u_k\} = argmin_{\{b_1..b_k\} ||b_i||_2=1 \forall i \in [1,k] b_i \perp b_j \forall i \neq j} E[||\tilde{x} - approx_{b_1..b_k}(\tilde{x})||_2^2]$
- thus we can write the optimal linear k dimensional approximation as

$$approx_{u_1...u_k}(\tilde{x}) = \Sigma_{i=1}^k \tilde{w}_i u_i : \quad \tilde{w}_i := u_i^t \tilde{x}$$

- which are the first k Principal Component of the covariance matrix of $\tilde{x}$

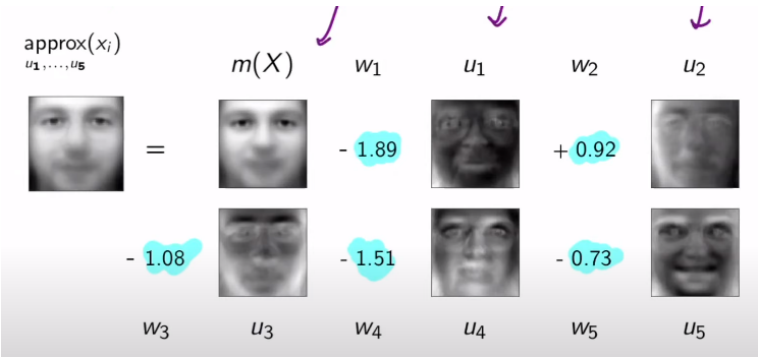
example

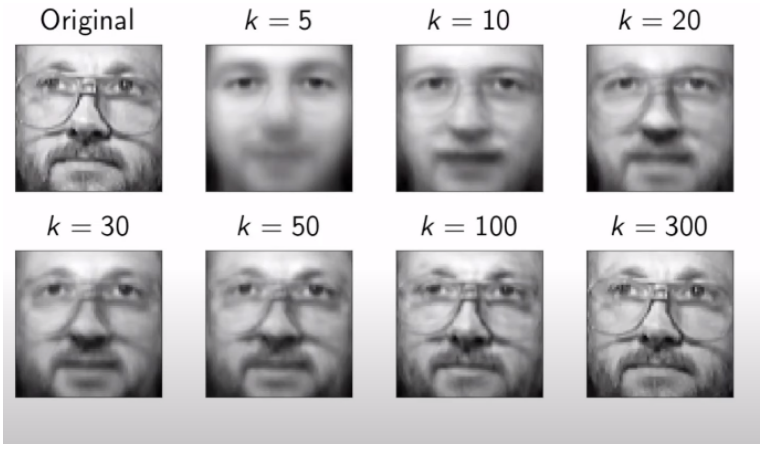
- example we have a dataset with 3 varieties of wheat
- each data point corresponding to a wheat seed has 7 features what is the best 2d representation of this data set (ie what captures the most l2 norm of our data, this is the same as capturing as much variance of our data as possible)
- the steps to do this as
    1. compute the covariance matrix of our dataset
    2. find the eigenvectors of that covariance
    3. take the top k
    4. get the the k Principal Component by taking the inner product of those Principal directions and our data
- this does well because it preserves that the varieties of wheat are for the most part separate
- this preserves important structures in our data
- if we used the last two Principal Components then we would get basically no variance and it would be very uninformative

faces example

- we are again using a dataset of faces
- 

- here suppose that each of our input vectors $X_i \in \mathbb{R}^{4096}$ and is basically a flattened out version of our 64 by 64 pixel value matrix
- we are going to center the data by subtracting the center mean
- so we can compute the covariance matrix for our dataset and then the first k Principal directions

- we can plot the Principal directions and see that they generally capture very coarse features of our dataset and get more fine as we get higher Principal directions (this makes sense as there is less variance in these finer scale features)
- so keep in mind that as we use these Principal directions we are just taking the directions that capture the most variance, that does mean we will likely capture some of the overarching structure of the dataset, but it does not mean that we will capture all the variation relevant to every down stream task.
- so suppose we take the 5 first Principal directions of our faces we can deffine our approximation in 5 dimensions as $approx_{u_1..u_5} := m(X) + \Sigma_{i=1}^{5} w_j[i]u_j : \quad w_j[i] := u_j^t ct(x_i)$
- 

- graphically this is what that looks like , ie the face of a single example in the 5 Principal Components
- with low Dimensionality we lose a lot of the information but we gain much more information as we keep adding Components
- 

- each of these representations are the Principal Components projected back onto the original space

face recognition example

- suppose given images of the same type as the last example
- we want to given a new picture identify who the person is in our training set
- item here we are going to use nearest neighbors which is not good in practice we would use a cnn

nearest neighbor classification

- a nearest neighbor classifier predicts as

$$i^* := argmin_{i \in [1,n]} ||x_{test} - x_i||_2$$

  that is just predicts the closest neighbor over our whole training set
- since we are doing an argmax over our whole training set we need to look at n vectors in $\mathbb{R}^d$ so the total time to classify a new point is $O(nd)$
- we can speed this up with practice
- we often use PCA as a pre processing steps

classification in a reduced space

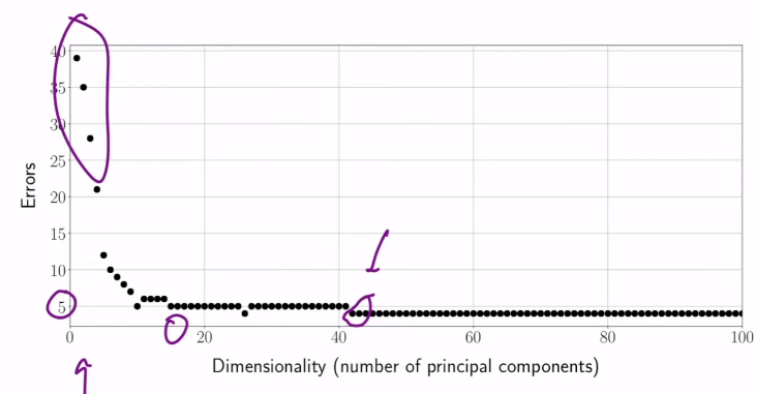
- compute sample mean and k first Principal directions $u_1..u_k$ from training data
- for each new data point x_test
  1. center the new point using the training sample mean to get $ct(x_{test}) = x_{test} - M(X)$
  2. compute the first k Principal Components

$$w_{test}[i] := u_i^t ct(x_test)$$

  3. apply our neighbors neighbors model to make a prediction

$$i_k^* := argmin_{i \in [1,n]} ||w_{test} - x_{1:k}[i]||_2$$

- so now the cost of predicting a new point is only $O(nk)$
- of course we are ignoring that we also need to do the pca before hand

- the good news is once you have done that, the marginal cost for each test point is lower
- 

- we can see the the error exponentially drops off in the number of Principal Components (at least generally)
- the info we are losing in Dimensionality reduction can either screen out noise (which is good) or hurt us if we are loosing info that is important for the classification

optimality

- we have argued that the using the first k Principal Components is optimal but we have not yet proved it
- want to show that

$$\{u_1...u_k\} = argmin_{\{b_1..b_k\}||b_i||_2=1 \forall i \in [1,k] b_i \perp b_j \forall i \neq j} E[||\tilde{x} - approx_{b_1..b_k}(\tilde{x})||_2^2] = E[||x||_2^2] - \Sigma var(b_i^t \tilde{x})$$

- we are going to show this with induction

k=1

- here is our basis step this holds directly by the spectral theorem $u_1 = argmax_{||b||=1} var(b^t \tilde{x})$

induction step

- suppose this holds for all $i \leq k$
- so lets fix an arbitrary set of orthonormal vectors $b_1..b_k$

- lets write $\Sigma_{i=1}^{k} var[b_i^t \tilde{x}] = \Sigma_{i=1}^{k} E[b_i^t \tilde{x}^2] = E[\Sigma_{i=1}^{k} b_i^t \tilde{x}^2] = E[||\Sigma_{i=1}^{k} b_i^t \tilde{x} b_i||_2^2] = E[||p_s(\tilde{x})||_2^2]$
- where $s = span(b_1..b_k)$ so that is the projection of $\tilde{x}$ on to the span of the first k Principal directions
- note that we can use any orthonormal basis of our subspace S $a_1..a_k$
- that is $P_S \tilde{x} = \Sigma_{i=1}^{k} b_i^t \tilde{x} b_i = \Sigma_{i=1}^{k} a_i^t \tilde{x} a_i$
- so we need to consider how to chose the right $a_1..a_k$
- S has dimension k, thus there is at least one vector $a_\perp \in S$ such that $a_\perp \perp \{u_1...u_{k-1}\}$ that is is orthonormal to our first k-1 Principal directions this is true because K has Dimensionality k
- this vector will have $var(u_k^t \tilde{x}) \geq var(a_\perp^t \tilde{x})$ by the pca as $u_k = argmax_{||a||_2=1, a \perp u_1 ... a \perp u_{k-1}} var(a^t \tilde{x})$
- so if we set $a_k = a_\perp$ and chose $a_1...a_{k-1}$ such that $span(a_1..a_k) = S$
- we have by the spectral theorem $var(u_k^T \tilde{x}) \geq var(a_{perp}^T \tilde{x}) = var(a_k^T \tilde{x})$
- and by our induction hypothesis we have $\Sigma_{i=1}^{k-1} var(u_i^t \tilde{x}) \geq \Sigma_{i=1}^{k-1} var(a_i^t \tilde{x})$
- meaning that $\Sigma_{i=1}^{k} var(u_i^t \tilde{x}) \geq \Sigma_{i=1}^{k} var(a_i^t \tilde{x}) = E[||P_S \tilde{x}||_2^2] = \Sigma_{i=1}^{k} var(b_i^t \tilde{x})$ which was our arbitrary set of k vectors fixed at the start
- and thus our induction step holds and the proof is completed the Principal Components minimize the mean of our squared approximation error