Machine Learning

Prof. Adil Khan

Objectives

1. A quick recap of last week
2. Feedback on the last topic from last week's recommended research
   - Constrained Optimization
   - L1 vs L2 as constrained optimization problems
3. High Dimensional Data
4. What is Principal Component Analysis? How does it help with high dimensional data? What is its objective function? How is it motivated?

Recap (1)

Naïve Bayes Classifier

$$p(y_{new} = c|\boldsymbol{x}_{new}, \boldsymbol{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{x}_{new}|c)p(c)}{\sum_{c=1}^{C} p(\boldsymbol{x}_{new}|\boldsymbol{c})\ p(c)}$$

$$p\left((x_1, \cdots, x_p)_{new}|c\right) = \prod_{i=1}^{p} p(x_i|c)$$

$$c_{NB} = \underset{c_j \in C}{\arg\max}\, P(c_j) \prod_i P(x_i \mid c_j)$$

Recap (2)

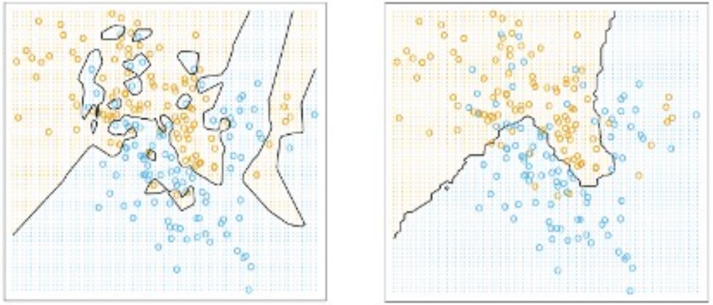
k-nearest Neighbor (KNN)

- Step 1: choose a value for k
- Step 2: Take the k neighbors of the new data point according to Euclidean distance
- Step 3: Among these k neighbor data points, count the number of points in each category
- Step 4: Assign the new data points to the category where you counted the most neighbors

Things to Remember about KNN

- Non parametric model
- Data is the model
- Curse of dimensionality
- Computational cost
- Feature scaling
- Handling missing data

$k = 1$ $k = 15$



Data

Training Data | Val Data | Test Data

And we train model as follows

Training with $k = 1$ →	Err = 5.2	
Training with $k = 5$ →	Err = 2.1 →	Err = 2.4
Training with $k = 15$ →	Err = 3.7	

Recap (3)

Regularization

- Helps to reduce *overfitting*

$$Success = Goodness\ of\ the\ Fit + Simplicity\ of\ the\ model$$

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T.\boldsymbol{x}_i)^2$$

$$\sum_{j=1}^{p} w_j^2$$

L_2 Regularization

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T.\boldsymbol{x}_i)^2 + \lambda\sum_{j=1}^{p} w_j^2$$

$$\lambda \geq 0\ is\ a\ tuning\ parameter$$

Ridge Regression

L_1 Regularization

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T.\boldsymbol{x}_i)^2 + \lambda\sum_{j=1}^{p} |w_j|$$

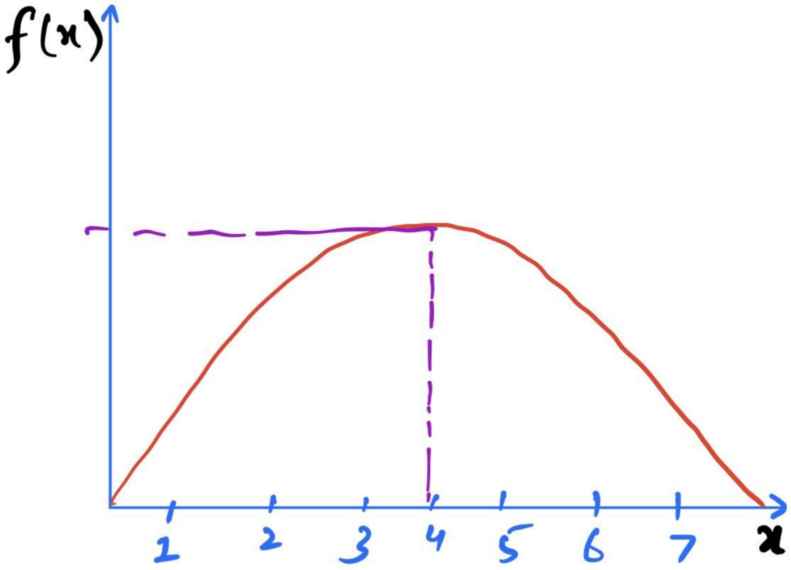
$$\lambda \geq 0\ is\ a\ tuning\ parameter$$

Lasso Regression

Why does $\boldsymbol{L_1}$ regularization give sparse models but $\boldsymbol{L_2}$ does not?

Optimization Problem

- Mathematical problem in which we want to MAXIMIZE or MINIMIZE a given function

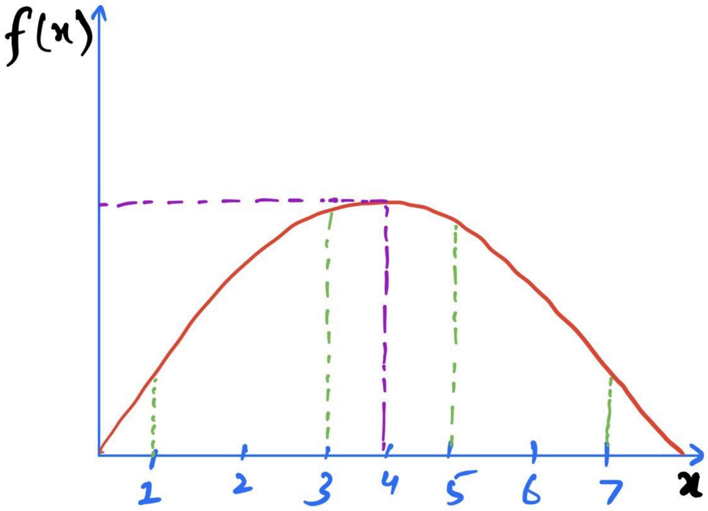


Solution: $x = 4$

But what if we are told that x must be odd?

Optimization Problem (2)

- Mathematical problem in which we want to MAXIMIZE or MINIMIZE a given function



But what if we are told that x must be odd?

Solution: x = 3

Constrained Opimization Problems

- Optimization problems where a function $f(x)$ has to be maximized or minimized subject to (s.t.) some constraint(s) $\phi(x)$

$$min\ f(x) \qquad\qquad max\ f(x)$$

$$s.t.\ \phi(x) \qquad\qquad s.t.\ \phi(x)$$

Constrained Opimization Problems

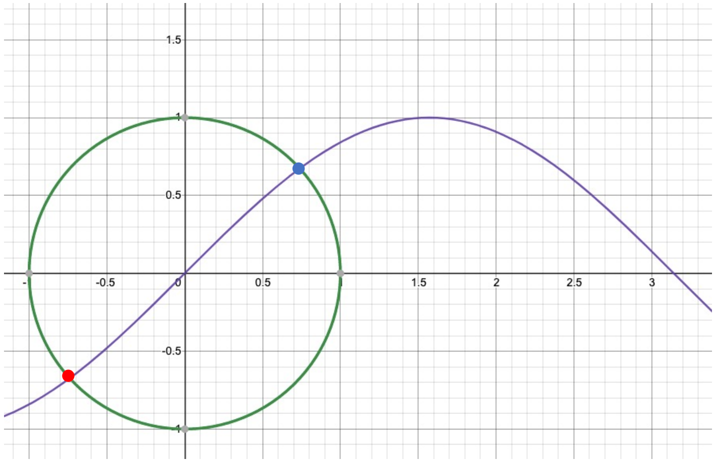
- One more example

$$f(x) = \sin(x)$$

$$\phi(x):\ x^2 + y^2 = 1$$

$$max\ f(x) \qquad min\ f(x)$$

$$s.t.\ \phi(x) \qquad s.t.\ \phi(x)$$



Solving Constrained Optimization Problems

- Such optimization problems are solved using **Lagrange Multipliers**

- In particular, we take our objective function and the constraint(s) and do the following

    - We make a new objective function
    - This new objective function contains both the original objective and additional term(s)
    - The additional term(s) represent(s) our constraint(s)

Solving Constrained Optimization Problems (2)

In particular, we take our objective function and the constraint(s) and do the following

- We make a new objective function
- This new objective function contains both the original objective and additional term(s)
- The additional term(s) represent(s) our constraint(s)

$$\underset{w}{\operatorname{argmin}} \; f(w) \quad \longrightarrow \quad \underset{w}{\operatorname{argmin}} \; f(w) - \alpha(g(w) - a)$$

subject to $g(w) < a$ $\quad\quad$ subject to $\alpha > 0$

α are called Lagrange Multipliers

This is all the detail that you need to know about them for this course

Alternate Forms of Regularization Objectives

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T.\mathbf{x}_i)^2 + \lambda\sum_{j=1}^{p} w_j^2$$

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T.\mathbf{x}_i)^2 + \lambda\sum_{j=1}^{p} |w_j|$$

$$\min_{w}\left\{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T.\mathbf{x}_i)^2\right\} \ \mathit{subject\ to} \sum_{j=1}^{p} w_j^2 \le s$$

$$\min_{w}\left\{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T.\mathbf{x}_i)^2\right\} \ \mathit{subject\ to} \sum_{j=1}^{p} |w_j| \le s$$

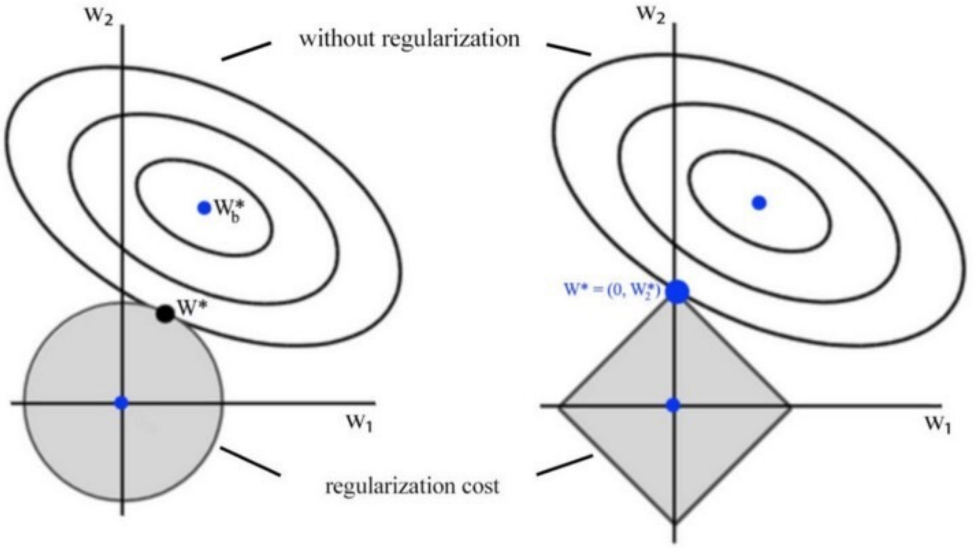
Thus, these are constrained optimization problems

Constraints in a Two Dimensional Space

$$w_1^2 + w_2^2 \leq s$$

$$|w_1| + |w_2| \leq s$$

Optimization Problem in a Two-Dimensional Space



L2 regularization promotes small parameters

L1 regularization promotes sparse parameters

High Dimensional Data

Curse of Dimensionality

1. Everyone is crazy about big data these days
2. Big data is our friend
3. We can put it to use for many creative applications
4. But let’s start with asking ourselves, how can a data become BIG?

Curse of Dimensionality (2)

- Making data big

1. Take huge number of samples
2. Measure huge number of **dimensions** for each sample

Curse of Dimensionality (3)

- Example of high dimensional data
  - Subjects: 3192
  - Measurements: 500,000

nature.com > nature > letters > article

MENU

nature
International journal of science

Altmetric: 247 Citations: 607 More detail »

Letter

Genes mirror geography within Europe

John Novembre, Toby Johnson, Katarzyna Bryc, Zoltán Kutalik, Adam R. Boyko, Adam Auton, Amit Indap, Karen S. King, Sven Bergmann, Matthew R. Nelson, Matthew Stephens & Carlos D. Bustamante

Nature **456**, 98–101 (06 November 2008)
doi:10.1038/nature07331
Download Citation

Received: 30 May 2008
Accepted: 12 August 2008
Published online: 31 August 2008

Addendum: 13 November 2008

Curse of Dimensionality (4)

- High Dimensional Data is
  - Difficult to visualize
  - Difficult to analyze
  - Difficult to understand – to get insight from the data (correlation and predictions)

The General Problem

- We have a dataset of n data points, where each point is p-dimensional

$$X = \{(\boldsymbol{x}_i) | \boldsymbol{x}_i \in \mathbb{R}^p \}_{i=1}^{n}$$

- The number of parameters in a machine learning model usually depends on the parameter p

- Thus if p is very large, it can make parameter estimation challenging

- It can also make visualization of the data very hard

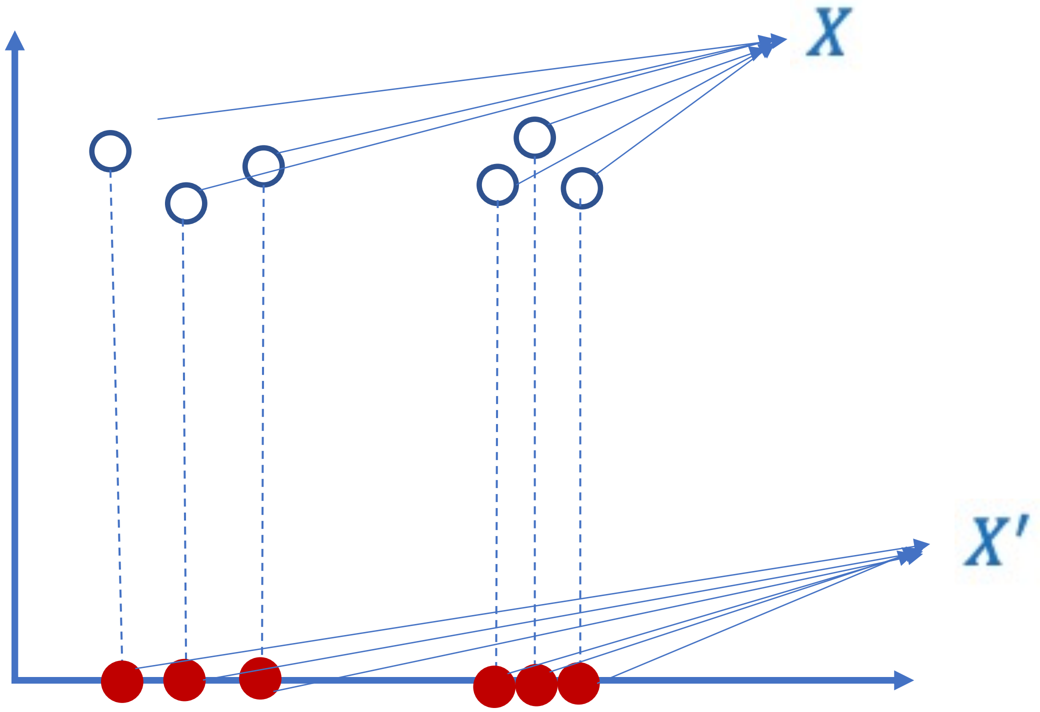
Solution

- To solve the problem, we usually transform every p-dimensional point $\boldsymbol{x}_i$ to a new d-dimensional point $\boldsymbol{x}_i'$
- Such that $d < p$

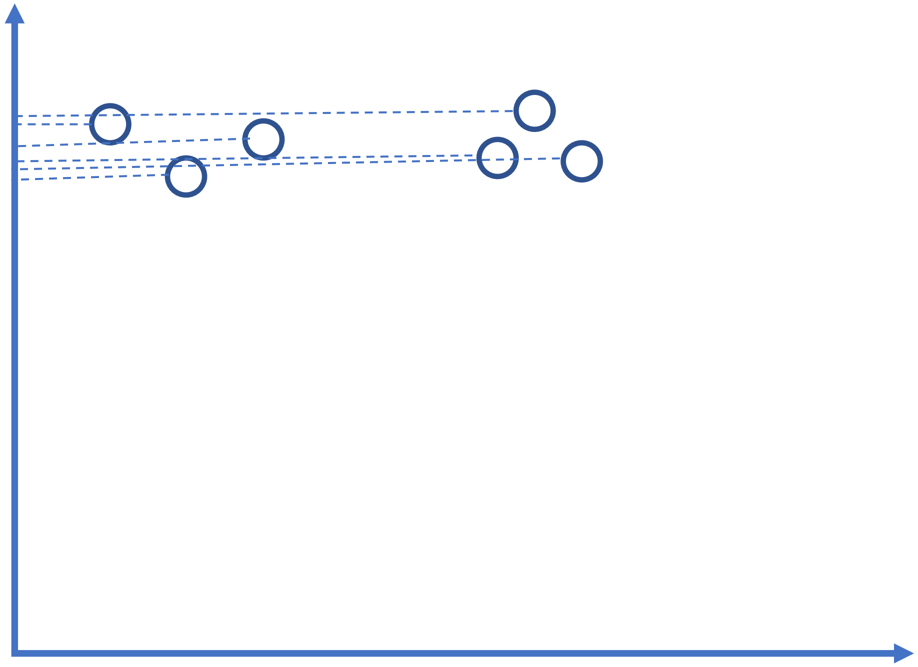
$$X' = \{(\boldsymbol{x}_i') \,|\, \boldsymbol{x}_i' \in \mathbb{R}^d\}_{i=1}^{n}$$

- This process is called ***projection***

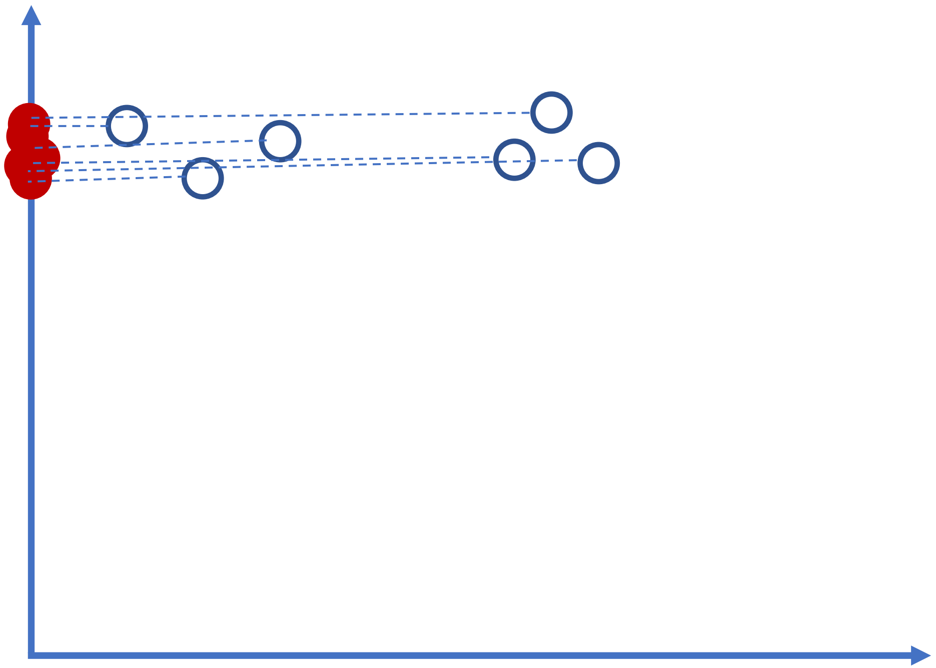
Data Projection



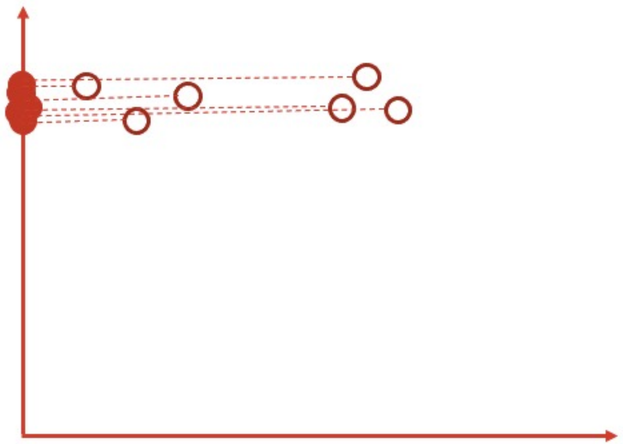
Data Projection (2)

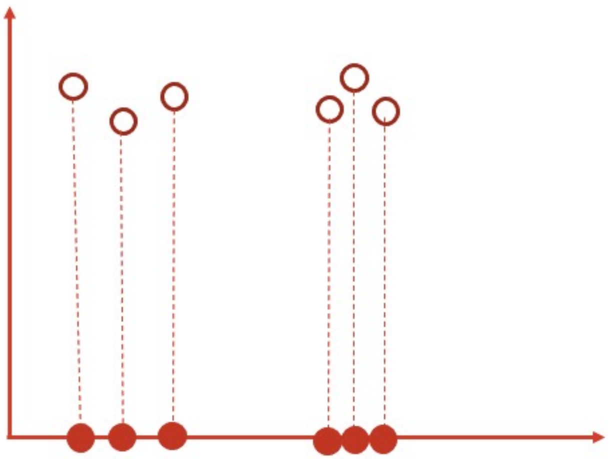


Data Projection (3)



Data Projection (4)





Which one should we choose between these two?

Thus

- When projecting data to a lower dimensional space, we would like to retain as much of the structural information about our data as possible

- And this is where variance can help us

- We can compute variance in each one-dimensional space as

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i' - \mu_{x'})^2 \qquad \mu_{x'} = \frac{1}{n}\sum_{i=1}^{n} x_i'$$

- Which we will try maximize when deciding on our projecting directions.

Principal Component Analysis (PCA)

PCA

- One of the most widely used technique for projecting data into lower dimensions

PCA (2)

- When projecting data from p-dimensional to a d-dimensional space, PCA defines d vectors, each represented as $\boldsymbol{w}_j$ where $j = 1, \cdots, d$

- Each vector is p-dimensional – that is, $\boldsymbol{w} \in \mathbb{R}^p$

- The $i - th$ projected point is represented as $\boldsymbol{x}'_i = [x'_{i1}, x'_{i2}, \cdots, x'_{id}]^T$

$$x'_{id} = \boldsymbol{w}_d^T \boldsymbol{x}_i$$

PCA (3)

- PCA uses variance in the projected space as its criteria to choose $\boldsymbol{w}_d$
- In particular, $\boldsymbol{w}_1$ will be the vector that will keep the variance in x'_{i1} as high as possible
- $\boldsymbol{w}_2$ Will be choses to maximize the variance, too, but with an additional constraint
- $\boldsymbol{w}_2$ must be orthogonal to $\boldsymbol{w}_1$

In general,

$$\boldsymbol{w}_i^T \boldsymbol{w}_j = 0, \qquad \forall i \neq j$$

PCA (4)

- In addition the previous constraint, the PCA also demands that

$$\boldsymbol{w}_d^T \boldsymbol{w}_d = 1$$

- Which means that each vector should have a length of 1

PCA (5)

- Finally, PCA requires that each original dimension has zero mean

$$\mu_{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i = 0$$

What do we know so far?

1. PCA reduces dimensionality by projecting data from a high dimensional space to a lower dimensional space
2. It does so by using a set of vectors
3. Vectors will be chosen such that they maximize the variance in the project space
4. Furthermore, the vectors
   - Should be orthogonal
   - Have unit length
5. Finally, data should have zero mean

How Does PCA Work?

How does PCA work?

- In order to understand how PCA works, let's start with projection into a 1-dimensional space, that is, $d = 1$

- In this case, for each $\boldsymbol{x}_i$ the result of projection of will be a scalar value

$$x_i' = \boldsymbol{w}^T \boldsymbol{x}_i$$

How does PCA work? (2)

- The variance in the projected space is given by

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i' - \mu_{x'})^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}(x_i' - 0)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}(x_i')^2$$

$$\mu_{x'} = \frac{1}{n}\sum_{i=1}^{n}x_i'$$

$$= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{w}^T\boldsymbol{x}_i$$

$$= \boldsymbol{w}^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\right) = \boldsymbol{w}^T\boldsymbol{\mu}_x$$

$$\mu_{x'} = 0$$

How does PCA work? (3)

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i')^2$$

- We also know that

$$x_i' = \boldsymbol{w}^T \boldsymbol{x}_i$$

- Substituting its value in the above equations gives us

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{w}^T \boldsymbol{x}_i)^2$$

How does PCA work? (4)

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_i)^2 \qquad = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{w}^T\boldsymbol{x}_i\,\boldsymbol{x}_i^T\boldsymbol{w}$$

$$= \boldsymbol{w}^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\,\boldsymbol{x}_i^T\right)\boldsymbol{w}$$

- Where C is the sample covariance matrix

$$\boxed{\sigma^2 = \boldsymbol{w}^T C\,\boldsymbol{w}}$$

Recall that the PCA Wants to

- Maximize the variance σ^2
- And we just derived that

$$\boxed{\sigma^2 = \boldsymbol{w}^T C\, \boldsymbol{w}}$$

- Therefore, the projection that maximized σ^2 would also maximize C

Maximizing σ^2: Trivial Solution

- Increase the value of the elements in $\boldsymbol{w}$
- And that is why, we already set the constraint

$$\boxed{\boldsymbol{w}^T\boldsymbol{w} = 1}$$

- Thus we are dealing with a constraint optimization problem

PCA Objective

- Find $\boldsymbol{w}$ that maximizes the following

$$\boxed{\mathcal{L} = \boldsymbol{w}^T C\, \boldsymbol{w} - \lambda(\boldsymbol{w}^T\boldsymbol{w} - 1)}$$

- Where λ is the Lagrange multiplier

Finding the Optimum $\boldsymbol{w}$

$$\boxed{\mathcal{L} = \boldsymbol{w}^T C\, \boldsymbol{w} - \lambda(\boldsymbol{w}^T \boldsymbol{w} - 1)}$$

- Take partial derivative with respect to $\boldsymbol{w}$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 2C\, \boldsymbol{w} - \lambda(2\boldsymbol{w} - \mathbf{0})$$

$$= 2C\, \boldsymbol{w} - \lambda(\boldsymbol{w})$$

- Setting it to 0, we get a very important result

$$\boxed{\lambda \boldsymbol{w} = C\, \boldsymbol{w}}$$

Let's Analyze What We Got

$$\boxed{\lambda \boldsymbol{w} = C\,\boldsymbol{w}}$$

- λ is a scalar
- $\boldsymbol{w}$ is a vector
- C is a matrix

1. Thus, multiplying $\boldsymbol{w}$ with C only scales it (only changes its length)
2. Thus, $\boldsymbol{w}$ that maximizes the variance is one of the **eigenvectors** of C and λ is the eigen value of $\boldsymbol{w}$

But $\boldsymbol{w}$ is Which EigenVector of $\boldsymbol{C}$?

- C is an $p \times p$ matrix
- Thus, it has p eigenvectors
- How do we know which one corresponds to highest variance in the projected space?

But $\boldsymbol{w}$ is Which EigenVector of $\boldsymbol{C}$? (2)

- Our expression for variance σ^2 is

$$\boxed{\sigma^2 = \boldsymbol{w}^T \boldsymbol{C}\, \boldsymbol{w}}$$

- We also know that $\boldsymbol{w}^T \boldsymbol{w} = 1$

- Thus we can write the equation of σ^2 as

$$\boxed{\sigma^2 \boldsymbol{w}^T \boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{C}\, \boldsymbol{w}}$$

But $\boldsymbol{w}$ is Which EigenVector of C? (3)

$$\boxed{\sigma^2 \boldsymbol{w}^T \boldsymbol{w} = \boldsymbol{w}^T C\, \boldsymbol{w}}$$

- Removing $\boldsymbol{w}^T$ from both sides, we get

$$\boxed{\sigma^2 \boldsymbol{w} = C\, \boldsymbol{w}}$$

- Note that we just showed that $\lambda \boldsymbol{w} = C\, \boldsymbol{w}$

- Thus

$$\boxed{\sigma^2 \boldsymbol{w} = \lambda \boldsymbol{w} = C\, \boldsymbol{w}}$$

Takeaway

$$\boxed{\sigma^2 \boldsymbol{w} = \lambda \boldsymbol{w} = C\, \boldsymbol{w}}$$

- Given a **(eigenvector $\boldsymbol{w}$, eigenvalue $\boldsymbol{\lambda}$)** pair of C, λ corresponds to the amount of variance in the projected space defined by $\boldsymbol{w}$

- Thus if we found p (eigenvector, eigenvalue) pairs of C, the pair with the highest eigenvalue corresponds to the vector that would maximize the variance in the projected space the most

Thus, Given $X = \{(\boldsymbol{x}_i) | \boldsymbol{x}_i \in \mathbb{R}^p\}_{i=1}^{n}$, PCA Works as Follows

1. Transform the data to have zero mean by subtracting μ_x from each point
2. Compute the sample covariance matrix C
3. Find p (eigenvector, eigenvalue) pairs of C
4. Find the eigenvectors corresponding to d highest eigenvalues $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_d$
5. Compute X' as $X' = XW$, where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_d]$

Recommended Reading

1. Section 7.2, *A First Course in Machine Learning*, by Simon Rogers and Mark Girolami

2. Section 6.2 from *Introduction to Statistical Learning* by Gareth James, Daniela Witten, Trevor Hastie, amd Robert Tibshriani

Summary

- Constrained Optimization Problems
- L1 vs L2 as constrained optimization problems
- High Dimensional Data and Their Issues
- Principal Component Analysis