‘Mathematic machine learning’ book for math

e.g.

Derivation not always works

# Gradient descent

s a small positive number (learning factor)

Multiple steps to the min x+ goes always larger than x and converge to min\_x

# Newton method

One step for 2-order

Symmetric matric

Eigenvalues are non-negative --- SPD

Use newton method

But for higher order eq, Q-1 would be enormous

Inner product and norm

Text

Description automatically generated

For matrix, extend inner product definition,

Multiplication of matrices

Generalized norm-p,

Eigenvalues and eigenvectors for non-zero matrix **u**.

Matrix A is square. Otherwise, there is only SVD.

Iden-potent matrix

The eigenvalue is

For arbitrary matrix Q,

For any square matrix Q, A is an iden-poten generated by Q, A is identity matrix

SPD, semi-positive definite and square,

A is a SPD

Eig(A) are non-negative

There exists a matrix B,

For every vertor u, <u,Au> >=0

Therefor, the result is non-negative

U and V are orthonormal vectors, D is a diagonal matrix, the elements are eigenvalue of A

Convex functions

Convex functions can always be solved for local minima

Convex domain, high dimensional cases,

The test segments should be also inside the domain

If the function is continuous and differentiable,

The convex function can be determined by Hessian matrix,

If is a SPD

# Supervised or unsupervised

Difference, prediction and inference, basic steps, fundamental trade-off, bias-variance trade-off

# Examples for supervise learning

Product sales for 200 markets

Advertising budget for TV, radio and newspaper

Connection function between the product sale and advertising budget

Notation

Sales is a response or target that we wish to predict, Y

TV, radio, newspaper are features or inputs or predictors, as X

The quantitative function can be written as

The noise is irreducible

# Unsupervised learning

Observations are given, (X1,Y1), (X2,Y2),….

K-means is an example of unsupervised learning

A method to determine cluster in terms of merely coordinates

# How do we estimate f?

Prediction   is a black box

Inference How Y changes as a function (Y) of X

# prediction

supervised learning have two steps for prediction

use (X1,Y1), (X2,Y2),…. To predict

use   to estimate   and a new

bias:

Note that

# Inference

How:

Parametric and non-parametric

Parametric Assume the data-generating process fowllow a probability distribution

Select a statistical model

Estimate the parameters of the model from the training data

# Parametric

Assume a functional form of *f*, e.g., and select a method to fit the model

N observations , ith observation, jth dependent variable

For a linear model,

Take derivative of the minimization function with respect to *,*

Alternatively,

And then take the derivative

# Non-parametric method

There is no assumption function

It could lead to an overfitting because no errors from the fitting

Trade-offs

Lose some prediction accuracy vs interpretability

Good fit vs over-fit or under-fit

Parsimony vs black box

Graphical user interface, text, application

Description automatically generated

Top left are on linear prediction with high interpretability; bottom right are more flexible but less details

# Model accuracy

Text, letter

Description automatically generated

Both overfitting and underfitting will lead to high error

A picture containing text

Description automatically generated

EPE ~ estimating prediction error

# Overcome overfitting

Choose right level of complexity

Data set sufficiency

Add some regularization term

e.g.,

controls the estimation >0, <1

# Lanczos algorithm

Linear regression

Python package: Sklearn.linear\_model.LinearRegression

R^2 value can be negative

When R^2 is 1, means a perfect fitting or prediction

When the prediction is mean value,  , means a trivial fitting only for mean values

# Projection matrix

For A matrix, the first column usually comes with (1,1,1,1,…)

The regression is to solve

We can take derivative for

Derivative for a column function should take transpose to the column first

Let

For weighted case

Where W,

# Exploring Overfitting

If the complexity of a fitting is 0,

The fitting result is a constant, indicating mean value of the sampling

Parameters need to be optimized is M+1

Bias is low if M+1 >= sampling, but the variance becomes larger

Table

Description automatically generated

w\* are coefficients of the fitting

# SVD

Reduced SVD

Eigenspace

If A is (1,1,1,,…)

If A is square,

If A is R^nxp(p>n),

rank(ATA) <= rank(A) <=n

In this case (ATA)-1 is not defined because p > n, the ATA is not full rank.

Therefore, modified lanczoc algorithm

Take the derivative of the function to X

Let

I identity here is pxp, the always has inverse because:

For a matrix A, can be expressed use the form, (UTU=I, VTV=I), is diagonal

The complexity is p^3

If we apply , the complexity reduced to n^3

If p>>n, the computational complexity can be reduced

If UTU=I and VTV=I,

The results are equivalent but the complexity is reduced

# SVD

In SVD, , if U or V is not square, is not necessarily identity. But always

In the case of square matrix

For the minimization

Therefore, the minimization is always 0 when A is a square matrix. It leads to square U, and .

If A is not square matrix,,n>p (dimension), U~nxp, ~pxp, V~pxp

But here is not identity

The fore the minimization is not 0 in this case

The expression above is a projection matrix

Analogous:

Find the orthogonal matrix to the plane we want to minimize

Meanwhile,

Is a eden-poten matrix

# Practice

506 data, first 100 data for training

13 features, includes intercept, 14 variables

Extract training data, selecting different amounts of training data

Some output to correct training

Standardize data points, z score in matlab, can achieve standardization

D degree expand, feature can be very high

**Ridge regression** to regulate features.

**Cross validation, google it**.

Make use of

To reduce complexity

# Subset selecting

## Best subset selection

denote null model, no predictors

For k=1,2,3,…p

For all models that contain exactly k predictors

Pick the best among these , call it

Best is defined as having the smallest RSS, or equivalent largest

If we have more freedom, we probable overfit the model

If we only care about , is best. **But** when p is large, is not easy to find it.

Therefore, using cross-validated prediction error, , or BIC, or adjusted

adjusted is , so when k increase, we will have penalty on R2 value

if go through all models, the complexity would be

## Forward stepwise selection

For k=0,1,2,3..p-1

choose the best around p -k models, and call it . The best is defined as having smallest RSS or highest R2.

The complexity would be

# Hw preparation

1.

Ridge regression

Prove and compare time consuption

2.

bias-variance trade-off

k-NN search

derive k-NN Error expression in TA’s notes

3.

Shrinkage Method

Note property for A, find the solution for three different regressions (vanilla, lasso and subset)

4

Linear regression and extension

In 3rd problem, expend to 2nd order. The intercept will not be expend, don’t forget to standardize data

And check overfitting with more and more features (more degree)

In 4th problem,Geometry mean, in log space for lambda of ridege regression

In 5th, also logspace for lambda. When lambda is extremely high, the beta will go to zero to minimize the expression of coefficients.

# Shrinkage method

In subset selection, we choose some coefficients in beta as zeros to reduce the complexity.

In the shrinkage method,

## The lasso regression

L1 norm is not derivativable

Alternative minimization for convex functions

Then, next iteration

Next,…, tth iteration ….

Algorithm for lasso regression:

For conversity, use 10-3 as for the program stop

Loss(i)-loss(i-1) < , the loop will stop

Lasso regression can automatically choose subset for coefficients, some components may go to zeros as the lambda increases

# Unsupervised learning

Reasonable guess from limited information

# PCA

Principal component analysis

It can reduce the dimension, or complexity of least square solution without losing important information

Variance calculation

For any vector p, the var(p)=||p||2. (1/n is ignored because it’s constant)

, when

X~Rnxp, project X along a certain object w, and X is already centered or under zero score

If the data is not centralized, we should move the axis leading to mean value for the data is zero.

It is equivalent to , because

Assume , we obtain . Therefore,

If we want to maximize the , it is to maximize the eigenvalue of .

By using SVD, [U,S,V]=svd(XTX), we will get the solution in decreasing order. While, in the case of [V,D]=eig(XTX), the order is randomized.

Assume

In the case of 2d,

If ,

If , the output will be 0, because of orthogal properties of SVD

Therefore,

Therefore, whatever , it will output corresponding eigenvalue

If , we will get the eigenvalue orthogonal to each other

If ,

Therefore,

And

We will have best solutions (not convex problem)

# Another perspective to PCA

Observations, , , ,…, consider the rank-q linear model for representing

Where is vector, is features, q denotes the number of features, is the basis

Function is the projections on the hyper plane (or the hyper plane itself), and xi are data points.

The best solution for is  ,  , because and

Therefore,

So, the minimization reach

Subscribe F is similar to L2 norm

# Group order:

10

Presentation order: 8

# PCA finalization and Examples

X is centered and W is orthonormal, , meaning each feature is independent

The solution for a least square problem will be

Leading to

The best solution for W will be

Because,

Therefore,

If we consider each column of W is an important feature, eigen feature.

The SVD on can lead to eigen features, has k feature,

If more features are included, like top 20 for top 100, the prediction will be more accurate.

It means, the results of SVD on will have feature space U, the selections of U will lead to different predictions. In order to reduce some less important features and decrease computational price, we can only include to k features from eigen space U.

# Kmeans

Break ties in the definition of smallest index of C*i*. it can lead to different convergent results.

Loss function is same as least square to determine which Kmean is better.

# Linear Discriminate Analysis

Project and reduce dimension to distinguish data sets

Biplot, plot data and features with different scales labeled with different colors

PCA: component maximize the variance

LDA: projection to lower dimension and classify at same time if the data is labelled

Criteria: In same class, variance should be small; while different classes should have large variance

If first class, x1, x2, x3, …, xn; 2nd class, y1, y2, y3, …, ym;

Projection w, the projection data would be , the centroid of X will be , similar for Y,

It will be the distance between two centroids, need to maximize

It will be the distances between data in the same cluster, need to minimize

Therefore, to put two parts on the fractions to do maximization

Dual variables introduction, Lagrange multiplier , to add constrain into objective function

e.g.

Let is the objective function, perform derivatives for all variables, I = x, y, lambda

It leads to , and

To get the best solution

Therefore,

Alternatively,

Take derivative for both variables,

It leads to an eigenvalue problem, and makes the problem solve for max of

The magnitude of w doesn’t matter.

We can think about , a scaler. Because leads to a scaler as well

Only one none zero eigenvalue for w, the rest are all zeros.

If we have the case of multiple classes

The objective function needs to be defined, total variance,

The goal is

Given a new datapoint x, the way to determine it belongs to X0 or X1 is to compare,

And

Entropy of informatics:

For x1,x2,x3,…xn

Maximize entropy

Equivalent to

Let J is the objective function, we take the derivative to the function J for every variable,

The result leads to an even value for all x,

# Gradient Descent

Where *f(x)* is differentiable

In each update, the objective function will be smaller.

In the end, we have to check

Gradient descent,

According to the definition, each step goes to a direction perpendicular to the tangent line of the curve, then next step will be finding the next perpendicular direction (<x,y> is the inner product)

If we plug in the gradient descent into the Taylor expansion above

# Convergence of Gradient Descent

Strongly convex strongly smooth function (SC, SS)

SC means the curvature is larger than second order polynomial function at a point

SS means the sharpness is not that high

Means a straight line, or tangent line at the point of the curve

Means a second polynomial function at the point

denote the , if we consider , and plug it in to right hand side of

It leads to

Therefore, in each iteration, the objective function will drop at least , it will stop if and only if .

If we change the , we will have

So, can be selected at most 1/alpha

Which is the lower boundary of the function, taken together

The rate of the convergence

Which is the definition of linear convergence, and

In the end, we need to limit ( is the optimal result, plus denote the each approaching step), in this case,

We can define to be the condition number, so we get

In this way, we can evaluate the iteration step by considering the precision .

In the case of vanilla least square

Is a convex problem with respect to , we can know that , thus

In the case of ridge regression,

Is also a convex function with respect to , we have , thus

If the largest SVD result for is nothing but zero, the function is just a convex one but not a strongly convex one

Therefore,

Consequently,

Then

Because

We need , sublinear convergence rate, which is slower than strongly convergence rate,

Kappa can be calculated in matlab use ‘cond(array)’

## Heavy ball method

Where, theta term is the momentum

The iteration complexity:

Nesterov Accelerate Gradient, best and fastest method

Step-size or

## Back tracking line search for step size

Try step size, change it according to next objective function value, usually shrinkage step size to its half

while converge:

End

# Hw2

Outliers for 2nd problem, can affect centroid selection. Sometimes median of the data may be a good centroid

3rd problem, use gradient descent way to solve, figure out learning rates, then in ridge regression case, figure out which one is converging faster

4th problem, use link to get standard image, 512x512 pixels, how to use less memory to approximate the original image. It is similar to PCA, but here, it will be treated with SVD, using significant features to achieve it.

# Spectral Clustering

Topics before are all linear. In some case distance-based method is not the optimal one.

## Graph theory

Undirected or directed graph

Partition of G

## Edge cut

Define

Where   means complementary, a subset of the set that excludes , rest set

:= the number of vertices in A

Degree (vol) is how many edges for a vertex

## Adjacency matrix, A

Shows the connections between any two vertices

If directed graph, the adjacency matrix will be symmetric

-neighborhood graph:

k-nearest neighbor graph:

## Degree matrix, D

Will be diagonal matrix for degrees of each vertex

Laplacian matrix

L=D-A, all rows or columns are linear dependent

It is SPD, the eigen value are non-negative

The diagonal element >= to sum of other elements from the row or the column, dominant theorem

Properties of L

For an arbitrary vector f,

Each adding term will be small, because if fi and fj are far away, wij will be small, vice versa

## RatioCut objective function

Where |Ai| is the number of nodes, or its volume

To minimize the objective function, it is equivalent to minimize , where

K means how many groups, i means how many data points, a points can only belong to one group

Therefore, the matrix (H) composed of hij is orthonormal

Not easy to deal with discontinuous constrain that

Therefore, the constrain is conserved, the objective function becomes:

Recall PCA problem, , the solution will be max SVD value of X^TX

Similarly,

Will result in the minimum SVD value of L

Once we get SVD(L), we can do kmean on the eigenspace of SVD(L)

# Data completion

SVD for approximation

SVD(Z),

For missing data

Rank function is not convex

## Relaxation

To relax a non-convex function into a convex one

Rank(M) ----- to ||M||\*, which is nuclear norm, sum of all singular values and it is convex

We can relax it to be convex:

Where is nuclear norm, assume [U, S, V]=SVD(M)

# Singular Value Thresholding (SVT)

For a X in nxm, have SVD [U,S,V’]

Consider the convex optimization problem

The solution is soft-threshold SVD

+ sign after each () indicate the results must be non-negative

# Soft-impute for matrix completion

Where D =

For the element not observed, just set them to be 0

In the case of ,

First step,

Assume first SVT(Z) is

Update

Second step

Keep iterating

# Hard-impute

Make sure the rank is exact r

Take SVD on each step, and take top r singular values

# Tensor data Recovery

For 3D or higher

Sensing operator A, if sensing iteration is high, more confidence for the recovery

More objects to optimize, use a AltMin for each objective until the entire objective function is converged

# Why nuclear norm instead of rank regularization

Rank is not convex, but nuclear norm is convex, except norm L0 (Li>1), other norms (Li>=1) are convex

# Comparison for Hard and Soft Thresholding

For a given vector z

e.g. ,

For a given matrix Z

is the singular matrix of Z

For a given vector z how to optimize

Lasso problem

For a given matrix Z

If singular value is larger than lambda, minus lambda, otherwise, 0

# Naïve Bayes

¼\*0.6+3/4\*0.4=0.55

P(red|o)=P(o|red)P(red)/P(o)=3/4\*0.4/0.45=2/3

P(fe|nflu)=0.15,P(fe|flu)=0.2

P(flu|fe)=P(fe|flu)P(flu)/P(fe)=0.2\*0.9/(0.8\*0.15+0.9\*0.2)=0.18/(0.12+0.18)=0.6

Example

P(N)=2/5;P(H)=3/5;

P(H|w,h,f)=P(w,h,f)P(H)/const

=P(w|H)P(h|H)P(f|H)P(H)

Because P(w|H)=0,the result will be 0, let’s treat it as a tiny number

## Laplacian Smoothing

e.g. P(w|H)=(0+1)/(3+3)=1/6

avoid 0 possibility in testing data

P(h|H)=

P(f|H)=

# Bayes Decision Boundary

for x coordinates

Compare for a larger probability above to decide which cluster a test data point will belong to

# Support Vector Machines (SVM)

SVM is intended for binary classification

## Hyperplane

Alternatively,

W here is the vector that perpendicular to the hyperplane

In general,

The distance between

And

Is

The objective is to

For the data points (positive labelled) above the separating hyperplane

It is equivalent to

It is equivalent to

Larger than

But it is not easy to max first for equation former. But if we first min the L, it will be doable by taking derivative. And the problem here is convex, the two equations above are equivalent.

Then plug it into original equation, and get

Equivalent to

Make use of sequential minimal optimization (SMO), optimize alpha pairwise

Once alpha is determined, w can be solved using w definition

When we determine b, if , , vice versa.

For those points that have non-zero , the data points will determine (support) the hyperplane

In other words,

For the points that support the hyperplane

For test data points, just plug into and check its sign

# Flexible cases In SVM

The key is to formulate the objective function, is to minimize *w*.

As for constrain, we can modify it by introducing a variable for each point i.

For the points inside two hyperplanes, their should be

Figure out objective function and do minimization of the maximization of the objective function

In the convex function case, we can flippe min and max, and do derivative to get the optimization

For each point I, we will have

In the end, it ends up with the same minimization

Complementary condition is

# HW3

X,Y,Z

From the constrain,

Let

u,v here can be from , first column

# Karush-Kuhn-Tucker (KKT) Practice

Assume , hyperplane: ,

Margin is

We need to maximum the margin

Is equivalent to

KKT condition:

is a dual variable, >=0

We know that L is smaller than J, so that

The objective is to

When the problem is convex one, we can flip max and min to get a easier computation

There is no constrain on w and b, so we can take the derivative with respect to w, b, leading to

In the case of

The objective is to ()

Because it is a convex function, we can change the order of min and max, leading to

We can take the derivative for min part

The objective function becomes

We can get to maximize the formular above

In other case

Therefore, we can do

And change the order of max and min

Derivative

Therefore

Because , therefore, , leading to objective is 0

# KKT conditions

After SMO, we may have

If we get , it is possible.

The number of nonzero should be more than 2, because we need 2 more points to support the planes

It could be possible is all zero, because the probability of the point on the unique line is zero compared to the probability on the entire plane

# Kernel

Extra feature can be introduced to do clustering like x (can’t be separated) to x^2 (separable)

It is called kernel

If sufficient dimensions is introduced, everything is separable

We can find a projection , , and d<=p

In the soft margin SVM problem

We can define in KKT conditions

A good projection or Kernel Method is to compute on the original space

We can verify that

Polynomial Kernel

Gaussian Kernel

It can go to infinite dimension

SVM Gaussian kernel, polynomial kernel, and then other clustering method

# Multi-classes classification

We can consider one cluster vs others as a binary case for each cluster (1 vs all)

We can get a table to check classifications

e.g. for a points in 3 clusters, after SVM we can get its classification in the form of a table

We can say the points belong to cluster 1, choose majority column

Another way is to do 1vs 1 classification, in k groups, choose 2 from k , and to check the classification table in . Choose the majority column

1 vs all is more popular due to less training compared to 1 vs 1.

# Logistic Regression

Surrogate relaxation

or

To generate a curve fit the discontinuous situation

The surrogate function should be continuous and differentiable

We can have

For a new data point, we can use formula above to tell it is negative or positive

If we denote , then the sigmoid function is S shape, and z is a scalar

# Properties of sigmoid function

Bounded in (0,1)

Monotonically increasing with respect to z

Logistic regression is not a regression model but a classification one. First fit , then determine the probability to each class

# Likelihood function

Maximize the likelihood function, take the log of it

If we denote ,  , we can formulate the objective

The objective function is convex, we can use gradient descent-based method to optimize it

# Stochastic gradient descent (SGD)

Choose a certain part of the data to represent the entire data set