Assignment 3

Machine Learning, Summer term 2014, Ulrike von Luxburg

To be discussed in exercise groups on May 5-7

Exercise 1 (Linear mapping, 1+1+1+1 points) Load the data from `Adot.mat`. Each column of matrix X represents one data point.

(a) Use the following code to calculate a linear mapping V. Apply the linear mapping on X to get $Y = VX$. Plot both X and Y in the same figure. What does the linear mapping V do?

```
theta = pi/3;
V = [cos(theta) -sin(theta);sin(theta) cos(theta)];
```

(b) Now apply the transpose of the linear mapping on Y to get $Z = V^tY$. Plot Z and describe what does the linear mapping V^tV do.

(c) What does the linear mappings `D1=[2 0;0 2]` and `D2=[2 0;0 1]` do? Apply them on X and plot the results.

(d) What does the linear mapping $A = V^t * D2 * V$ do? Apply it on X and plot the result.

Exercise 2 (Inverse of a matrix, 1 point) Assume that V is a $n \times n$ matrix such that $VV^t = V^tV = I$, where I is the identity matrix. Moreover, D is a diagonal matrix

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix} \tag{1}$$

where $d_i > 0$. Prove that the inverse of the matrix $A = VDV^t$ is $A^{-1} = VD^{-1}V^t$. Here, D^{-1} is a diagonal matrix with diagonal entries $1/d_i$. To prove that $B = A^{-1}$, it is enough to show that $AB = BA = I$.

Exercise 3 (Convexity, 2 points) Prove that the least squares loss function $\|Y - Xw\|^2$ is a convex function of w.

Exercise 4 (Linear regression with weights, 4 point) Consider a data set in which each data point X_i is associated with a weighting factor $r_i > 0$, so that the empirical least squares error becomes

$$E = \frac{1}{n}\sum_{i=1}^{n} r_i(Y_i - \langle w, X_i\rangle)^2.$$

Find an expression for the solution w that minimizes this error function.

Exercise 5 (Ridge regression, 2+2+4 points) In this exercise, you will implement the ridge regression algorithm. Load the synthetic train and test data from `dataRidge.mat`.

(a) Run the linear least square and plot the training points, the predicted line and the predicted values for the test data.

(b) Linear least square with polynomial basis functions

Web page: `http://www.informatik.uni-hamburg.de/ML/contents/people/luxburg/teaching/2014-ss-vorlesung-ml/`
Login: “machine”, Password: “learning”

- Run the linear least square with polynomial basis functions

$$\Phi_i(x) = x^i; i = 1, ..., 15. \tag{2}$$

  Illustrate the learned regression function by applying it on `xx=-1.5:0.01:2.5`.
- Describe the class of functions that you can learn with these basis functions.

(c) Ridge regression

- Write a function `RidgeLLS(X,Y,lambda)` which implements the ridge regression. Here, X is the design matrix.
- Apply the ridge regression on the test data using the set of basis functions in Equation 2. Plot the prediction function (on the previous figure) for regularization constant $\lambda \in \{0.0001, 0.1, 10\}$.
- Report the prediction error with respect to λ for $\lambda \in \{2^i; i = -15, -14, ..., 1\}$.

Exercise 6 (Prediction complexity of linear least square and kNN, 1 point) Compare the prediction running time for linear least square method and kNN regression (computational complexity). How much information do you need to keep for predicting with each method (space complexity)?

MACHINE LEARNING- ASSIGNMENT 3

VICTOR BERNAL ARZOLA

Exercise 2

Given that $VV^T = V^TV = 1$ and $D = diag\left(d_i\right)$ an invertible matrix.

Proposition. *Show that for $A = VDV^T$ then $A^{-1} = VD^{-1}V^T$*

Proof. it is straighforward to check that

$$AB = VDV^TVD^{-1}V^T = VD\left(V^TV\right)D^{-1}V^T = V\left(DD^{-1}\right)V^T = VV^T = 1$$

and

$$BA = VD^{-1}V^TVDV^T = VD^{-1}\left(V^TV\right)DV^T = V\left(D^{-1}D\right)V^T = VV^T = 1$$

then □

$$\Rightarrow A^{-1} = B$$

Exercise 3

Proposition. *Prove that the least squares loss function is a convex function of w.*

A way to prove that a function is convex is by showing that the second order derivative (if it exists) is positive semi-definite.

Proof. Recall that if the matrix $z^TX \neq 0\ \forall z \neq \mathbf{0}$ then the matrix $M = XX^T$ satisfy that □

$$z^TMz = z^TXX^Tz = \left\|z^TX\right\|_2^2 > 0$$

then XX^T is a positive definite matrix.

Exercise 4

Consider a data set in which each data pointX_i is associated with a weighting factor r_i

Proposition. *Prove that the weighted least squares loss function is a convex function of w.*

Proof. Let $W = diag(r_i)$ where $r_i > 0$ Therefore, $W^T = W$
optimize the weighted loss function □

$$\frac{1}{n}\sum_{i=1}^{n} r_i\left(y_i - \langle X_i, w\rangle\right)^2$$

is equivalent to optimize (see[3])

$$= \left(y - Xw\right)^T W\left(y - Xw\right)$$

$$= \left(y^TW - w^TX^TW\right)\left(y - Xw\right)$$

$$= y^TWy - w^TX^TWy - y^TWXw + w^TX^TWXw \tag{0.1}$$

using differentiation we have

$$\frac{\partial}{\partial w}\left(y^TWy - w^TX^TWy - y^TWXw + w^TX^TWXw\right)$$

Output	Complexity	Operation
X^TX	$O(md^2)$	Matrix Multiplication
X^Ty	$O(md)$	Matrix Multiplication
$\left(X^TX\right)^{-1}$	$O(d^3)$	Inverse (Gauss Jordan)
$w = \left(X^TX\right)^{-1}X^Ty$	$O(d^2)$	Matrix Multiplication

TABLE 1. Time Complexity Least Squares Operations

Here X is a mxd matrix, y a mx1 matrix.

$$= -X^TWy - y^TWX + 2w^TX^TWX$$

$$= -2X^TWy + 2X^TWXw$$

is equal to zero, leading to

$$w = \left(X^TWX\right)^{-1}X^TWy \tag{0.2}$$

The Hessian given by differentiating again 0.1

$$\frac{\partial}{\partial w^T}\left(-X^TWy - y^TWX + 2w^TX^TWX\right) = 2X^TWX \tag{0.3}$$

is positive semi-definite matrix.

EXERCISE 5

Complexity of a algorithm. How long does this sorting program run? The number of (machine) instructions which a program executes during its running time is called its time complexity in computer science. The better the time complexity of an algorithm is, the faster the algorithm will carry out his work in practice. For the time computational complexity we must add up how many machine instructions it will execute (as a function of the size of its input) and then simplify the expression to the largest term and can include any simplifying constant factor.
Apart from time complexity, its space complexity is also important: This is essentially the number of memory cells which an algorithm needs. A good algorithm keeps this number as small as possible, too.Time and Space complexity are different aspects of calculating the efficiency of an algorithm. Time complexity deals with finding out how the computational time of an algorithm changes with the change in size of the input. On the other hand space complexity deals with finding out how much (extra)space would be required by the algorithm with change in the input size. To calculate time complexity of the algorithm the best way is to check if we increase in the size of the input, will the number of comparison(or computational steps) also increase and to calculate space complexity the best bet is to see additional memory requirement of the algorithm also changes with the change in the size of the input.[2]
There is often a time-space-trade off involved in a problem, that is, it cannot be solved with few computing time and low memory consumption. One then has to make a compromise and to exchange computing time for memory consumption or vice versa, depending on which algorithm one chooses and how one parameterizes it. Generally, the notation $f(n) = O(g(n))$ says that the function f is asymptotically bounded from above by the function g. Lets suppose X has m data points in d dimensions.
So we have $O(md^2 + d^3 + md + d^2)$. Now, we shall assume that $m \gg d$ (more data points than dimensions). In this case, $O(md^2)$ dominates over $O(d^3)$, $O(md)$ and $O(d^2)$. Thus the asymptotically time computational complexity of LLS is $O(md^2)$.
Thus the asymptotically space computational complexity of LLS is $O(md)$
Now, we shall assume that $k \ll m$ and $m \gg d$ (more data points than dimensions). In this case, $O(md)$ dominates the rest. Thus the asymptotically time computational complexity of KNN is $O(md)$.
Thus the asymptotically space computational complexity of KNN is $O(md)$

Object	Space
X	md
X^T	dm
y	m
$X^T X$	d^2
$X^T y$	d
$\left(X^T X\right)^{-1}$	d^2
w	d

TABLE 2. Space Complexity Least Squares Operations

Here X is a mxd matrix, y a $mx1$ matrix.

Output	Complexity	Operation
$X_{test} - X_{train}, i$	$O(md)$	m subtraction for d dimensions
$\left(X_{test} - X_{train}, i\right)^2$	$O(m)$	m squares
$\sum_{i=1}^{n} \left(X_{test} - X_{train}, i\right)^2$	$O(m)$	m sums
$\sqrt{\sum_{i=1}^{n} \left(X_{test} - X_{train}, i\right)^2}$	$O(m)$	square roots for m points
sort	$O(m \ln m)$[1]	sort m points
majority of k nearest	$O(k)$	Mode

TABLE 3. Time Complexity KNN

Here X is a mxd matrix, y a $mx1$ matrix.

Object	Space
X	md
y	m
$distance$	m
$sort$	m
k	k

TABLE 4. Space Complexity Least Squares Operations

Here X is a mxd matrix, y a $mx1$ matrix.

REMARKS

- The Hessian matrix $\mathbb{H}_{ij} = \frac{\partial f}{\partial x_i \partial x_j}$ of a convex function $f\left(\mathbf{x}\right)$ is positive semi-definite(i.e $z^T \mathbb{H} z \geq 0$ $\forall z \neq \mathbf{0}$). Refining this property allows us to test if a critical point x is a local maximum or a local minimum, as follows.
- If the Hessian $\mathbb{H}$ is positive definite at x, then $f\left(\mathbf{x}\right)$ attains a local minimum at $\mathbf{x}$. If the Hessian is negative definite at $\mathbf{x}$, then $f\left(\mathbf{x}\right)$ attains a local maximum at $\mathbf{x}$. If the Hessian has both positive and negative eigenvalues then $\mathbf{x}$ is a saddle point for $f\left(\mathbf{x}\right)$. Otherwise the test is inconclusive. This implies that, at a local minimum (resp. a local maximum), the Hessian is positive-semi-definite (resp. negative semi-definite).
- In one variable, the Hessian contains just one second derivative; if it is positive then x is a local minimum, and if it is negative then x is a local maximum; if it is zero then the test is inconclusive. In two variables, the determinant can be used, because the determinant is the product of the eigenvalues. If it is positive then the eigenvalues are both positive, or both negative. If it is negative then the two eigenvalues have different signs. If it is zero, then the second derivative test is inconclusive

REFERENCES

[1] https://en.wikipedia.org/wiki/Sorting_algorithm#Comparison_of_algorithms
[2] https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations
[3] https://en.wikipedia.org/wiki/Matrix_calculus