IFT-6390 Fundamentals of Machine Learning
Professor: Ioannis Mitliagkas

Homework 2 - Theoretical part

- This homework must be done and submitted to Gradescope and can be done in groups of up to 2 students. You are welcome to discuss with students outside of your group but the solution submitted by a group must be its own. Note that we will use Gradescope's plagiarism detection feature. All suspected cases of plagiarism will be recorded and shared with university officials for further handling.

- Only one student should submit the homework and add you should add your group member on the submission page on gradescope

- You need to submit your solution as a pdf file on Gradescope using the homework titled `(6390: GRAD) Homework 2 Theory`.

1. **Bias-Variance decomposition** [2 points]

   Consider the following data generation process: an input point x is drawn from an unknown distribution and the output y is generated using the formula

   $$y = f(x) + \epsilon,$$

   where f is an unknown deterministic function and $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$. This process implicitly defines a distribution over inputs and outputs; we denote this distribution by p.

   Given an i.i.d. training dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn from p, we can fit the hypothesis h_D that minimizes the empirical risk with the squared error loss function. More formally,

   $$h_D = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} (y_i - h(x_i))^2$$

https://web.engr.oregonstate.edu/~tgd/classes/534/slides/part9.pdf

Slide8

where $\mathcal{H}$ is the set of hypotheses (or function class) in which we look for the best hypothesis/function.

The expected error[1] of h_D on a fixed data point (x', y') is given by $\mathbb{E}[(h_D(x') - y')^2]$. Two meaningful terms that can be defined are:

- The bias, which is the difference between the expected value of hypotheses at x' and the true value $f(x')$. Formally,

$$\mathit{bias} = \mathbb{E}[h_D(x')] - f(x')$$

- The variance, which is how far hypotheses learned on different datasets are spread out from their mean $\mathbb{E}[h_D(x')]$. Formally,

$$\mathit{variance} = \mathbb{E}[(h_D(x') - \mathbb{E}[h_D(x')])^2]$$

Show that the expected prediction error on (x', y') can be decomposed into a sum of 3 terms: $(\mathit{bias})^2$, *variance*, and a *noise* term involving ϵ. You need to justify all the steps in your derivation.

[1] Here the expectation is over random draws of the training set D of n points from the unknown distribution p. For example (and more formally): $\mathbb{E}[(h_D(x')] = \mathbb{E}_{(x_1,y_1)\sim p} \cdots \mathbb{E}_{(x_n,y_n)\sim p} \mathbb{E}[(h_{\{(x_1,y_1),\ldots,(x_n,y_n)\}}(x')]$.

2. **Optimization** [10 points] Assume a 1D logistic function:

$$\sigma(wx) = \frac{1}{1 + e^{-wx}}$$

where $x, w \in \mathbb{R}$, and the associated cost function:

$$L(w) = -y \log \sigma(wx) - (1 - y) \log (1 - \sigma(wx))$$

(a) Show that the cost function associated with logistic regression is convex. You can use one of the following two definitions of convexity:

https://www.cs.mcgill.ca/~dprecup/courses/ML/Lectures/ml-lecture05.pdf

- f is convex if and only if

$$\forall x_1, x_2, t \in [0, 1] : f(tx_1 + (1 - t)x_2) < tf(x_1) + (1 - t)f(x_2)$$

- f is convex if and only if $\frac{d^2 f}{dx^2}(x) > 0$ for all x

You can also use another definition of convexity but you have to explicitly state it.

(b) Find the gradient of $\sigma(wx)$ at some point w. What are the dimensions of the gradient?

Closed form -Analytically(Matrix)
Stationary point--> it is 0

(c) Find all of the stationary points of $L(w)$ analytically, i.e. through a closed-form expression (Justify).

(d) Show how the gradient descent update rule looks like in this case by substituting $\sigma(wx)$ with its form above. Use the following notation: w_0 represents our point at initialization, w_1 represents our point after one step, etc.

3. **Least Squares Estimator and Ridge Regression** [10 points]

We consider the problem of learning a vector-valued function $f : \mathbb{R}^d \to \mathbb{R}^p$ from input-output training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ where each $\mathbf{x}_i$ is a d-dimensional vector and each $\mathbf{y}_i$ is a p-dimensional vector. We choose our hypothesis class to be the set of linear functions from $\mathbb{R}^d$ to $\mathbb{R}^p$, that is function satisfying $f(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ for some $d \times p$ regression matrix $\mathbf{W}$, and we want to minimize the squared error loss function

$$J(\mathbf{W}) = \sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{W}^\top \mathbf{x}_i\|_2^2 \qquad (1)$$

over the training data.

Let $\mathbf{W}^*$ be the minimizer of the empirical risk:

$$\mathbf{W}^* = \arg\min_{\mathbf{W} \in \mathbb{R}^{d \times p}} J(\mathbf{W}).$$

https://see.stanford.edu/materials/aimlcs229/
ps2_solution.pdf

Page 12: part 2-CS229 notes

(a) Derive a closed-form solution for $\mathbf{W}^*$ as a function of the data matrices $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathbb{R}^{n \times p}$.
(hint: once you have expressed $J(\mathbf{W})$ as a function of $\mathbf{X}$ and $\mathbf{Y}$, you may find the matrix cookbook useful to compute gradients w.r.t. to the matrix $\mathbf{W}$)

ttps://statisticaloddsanc
ends.wordpress.com/201
3/02/09/xtx-and-ridge-re
gression/

Rigde regression

A variation of the least squares estimation problem known as ridge regression considers the following optimization problem:

$$\arg\min_{\mathbf{W}} J(\mathbf{W}) + \lambda \|\mathbf{W}\|_{\mathbf{F}}^{\mathbf{2}} \tag{2}$$

where $\lambda > 0$ is a regularization parameter. The regularizing term penalizes large components in $\mathbf{W}$ which causes the optimal $\mathbf{W}$ to have a smaller norm.

(e) Derive the solution of the ridge regression problem. Do we still have to worry about the invertibility of $\mathbf{X}^\top \mathbf{X}$?

Penalty,as the
complexity increases

(f) Explain why the ridge regression estimator is likely to be more robust to issues of high variance compared with the least squares estimator.

if lambda is high ,then it causes
bias
if lambda is low, then variance issue

(g) How does the value of λ affect the bias and the variance of the estimator?

4. **k-fold cross-validation** [10 points]

Let $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a training sample set drawn i.i.d. from an unknown distribution p. To estimate the risk (a.k.a. the test error) of a learning algorithm using D, k-fold cross validation (CV) involves using the ith fold of the data $D_i = \{(x_j, y_j) \mid j \in \text{ind}[i]\}$ (where ind$[i]$ are the indices of the data points in the ith fold) to evaluate the risk of the hypothesis returned by a learning algorithm trained on all the data except those in the ith fold, $D_{\setminus i} = \{(x_j, y_j) \mid j \notin \text{ind}[i]\}$.

Formally, if we denote the hypothesis returned by the learning algorithm trained on $D_{\setminus i}$ as $h_{D \setminus i}$, the k-fold CV error is given by

$$\text{error}_{k-fold} = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{n/k} \sum_{j \in \text{ind}[i]} l(h_{D \setminus i}(x_j), y_j)$$

where l is the loss function.

In this exercise, we will investigate some interesting properties of this estimator.

k-fold is unbiased

(a) State the definition of the risk of a hypothesis h for a regression problem with the mean squared error loss function.

(b) Let D' denote a dataset of size $n - \frac{n}{k}$. Show that

$$\mathop{\mathbb{E}}_{D\sim p}[\text{error}_{k-fold}] = \mathop{\mathbb{E}}_{\substack{D'\sim p,\\ (x,y)\sim p}}[(y - h_{D'}(x))^2]$$

where the notation $D \sim p$ means that D is drawn i.i.d. from the distribution p, h_D denotes the hypothesis returned by the learning algorithm trained on D. Explain how this shows that error_{k-fold} is an (almost) unbiased estimator of the risk of h_D.

Complexity of k-fold We will now consider k-fold in the context of linear regression where inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are d-dimensional vectors. Similarly to exercise 3, we use $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{y} \in \mathbb{R}^n$ to denote the input matrix and the vector of outputs.

(c) Assuming that the time complexity of inverting a matrix of size $m \times m$ is in $\mathcal{O}(m^3)$, what is the complexity of computing the solution of linear regression on the dataset D? (i.e. similar to the solution of 3 (a))

(d) Let $\mathbf{X}_{-i} \in \mathbb{R}^{(n-\frac{n}{k})\times d}$ and $\mathbf{y}_{-i} \in \mathbb{R}^{(n-\frac{n}{k})}$ be the data matrix and output vector obtained by removing the rows corresponding to the ith fold of the data. Using the formula for $error_{k-fold}$ mentioned at the start of this question, write down a formula of the k-fold CV error for linear regression. Specifically, substitute the loss expression with the actual loss obtained by using the analytical solution for linear regression. What is the complexity of evaluating this formula?

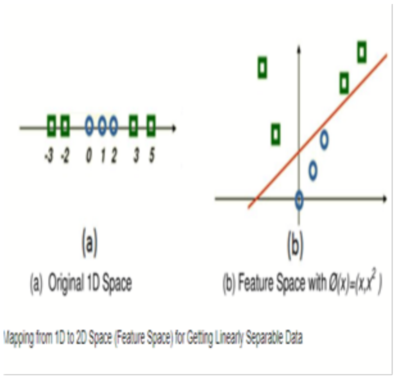
(e) It turns out that for the special case of linear regression, the k-fold validation error can be computed more efficiently. Show that in the case of linear regression we have

$$\text{error}_{k-fold} = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{\mathbf{y}_i - \mathbf{X}_i\mathbf{w}^*}{1 - \mathbf{X}_i(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}_i^\top}\right)^2$$

where $\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ is the solution of linear regression computed on the whole dataset D. What is the complexity of evaluating this formula?

5. **Feature Maps** [8 points]

In this exercise, you will design feature maps to transform an original dataset into a linearly separable set of points. For the following questions, if your answer is '*yes*', write the expression for the proposed transformation; and if your answer is '*no*', write a brief explanation. You are expected to provide explicit formulas for the feature maps, and these formulas should only use common mathematical operations.



https://towardsdatascience.com/the
-kernel-trick-c98cdbcaeb3f

(a) [2 points] Consider the following 1-D dataset (Figure 1). Can you propose a 1-D transformation that will make the points linearly separable?

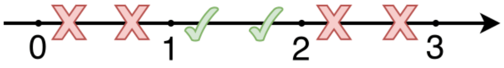


Figure 1: 1D dataset

(b) [2 points] Consider the following 2-D dataset (Figure 2). Can you propose a transformation into 1D that will make the data linearly separable?

Similar,
https://www.csd.uwo.ca/~oveksler/
Courses/CS434a_541a/Lecture11.pc
f

http://www.biocomp.unibo.it/gigi/2
015-2016/SSB/Kernels.pdf

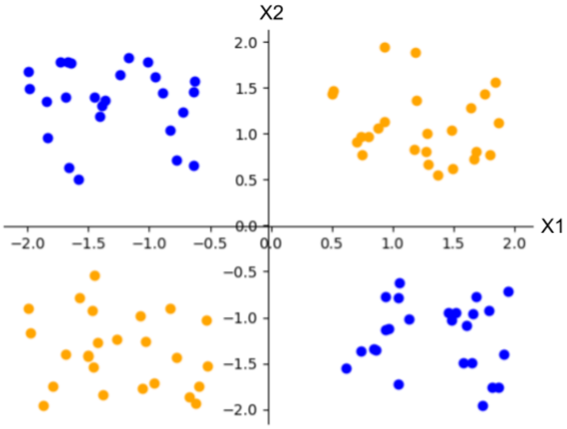
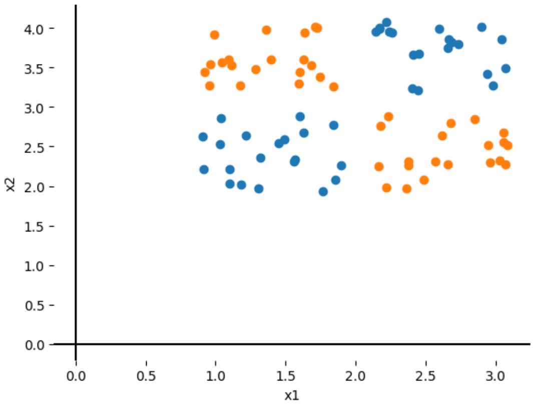


Figure 2: 2D dataset

(c) [4 points] Using ideas from the above two datasets, can you suggest a transformation of the following dataset (as shown in

Figure 3) that makes it linearly separable? If '*yes*', also provide the kernel corresponding to the feature map you proposed.



https://www.cse.iitk.ac.in/users/purushot/slides/svm-iiita09.pdf

Figure 3: Another 2D dataset