

Homework 2 (Classification, Overfitting)

For questions, please refer to Moodle.
Released on **22 March, 2022**

GENERAL INSTRUCTIONS

- Submission of solutions is not mandatory but solving the exercises are highly recommended. The master solution will be released next week.
- Part of the exercises are available on Moodle as a quiz. These problems are marked with [📝].

Exercise 1: True-False Classification with Asymmetric Losses

For a company's new spam email filter, we have designed two different classifiers to determine if an email is spam ($y = 1$) or non-spam ($y = -1$). Assessing the quality of a spam classifier is difficult. A standard 0 – 1 classification loss does not account for the asymmetric business costs of incorrectly classifying a non-spam email as spam vs incorrectly classifying a spam email as non-spam. To address this, we set our null hypothesis to be that an email is non-spam and evaluate the two classifiers. Each classifier uses two features. Figure 1 illustrates the dataset with the two classifiers (A and B) each represented by a hard decision boundary that classifies the datapoints into two groups. The two classifiers classify the groups identically but using different decision boundaries. So first we will just study the performance of A.

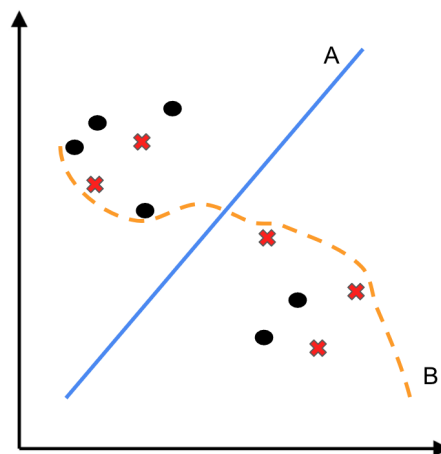


Figure 1: The '•' labels indicate that an email is non-spam ($y = -1$), '×' indicates spam ($y = 1$). Everything to the right of line labelled A is classified by A as spam. Everything below the curve of B is classified as spam by B.

- [📝] Compute the False Positive Rate (FPR) of classifier A.
- [📝] Compute the False Discovery Rate (FDR) of classifier A.
- [📝] Compute the precision of classifier A.
- [📝] Compute the recall of classifier A.

(e) ☒ Compute the F1 Score of classifier A.

(f) ☒ Decide whether the following statements are True or False.

When comparing two classifiers on validation data, the best is always the classifier with highest F1 score.

☐ True ☐ False

Classifier A could have been computed by hard-margin linear Support Vector Machine (SVM) without modifying the data, but not a soft margin SVM.

☐ True ☐ False

Classifier A is more robust than B to adversarial perturbations of the training datapoints.

☐ True ☐ False

The company decides that using a hard decision boundary is not the best way to trade-off costs in the spam filter problem. Instead, they design 3 classifiers C, D, E that each use a threshold τ which controls the trade-off between false-positive rate (FPR) and true-discovery rate (TPR). For example, classifier C learns a prediction function \hat{f}_C that maps from the input features x to $[0, 1]$. An email is then classified as spam if $\hat{f}_C(x) < \tau$. Figure 2 depicts the trade-off between FPR and True Positive Rate (TPR) for each classifier as a result of varying τ .

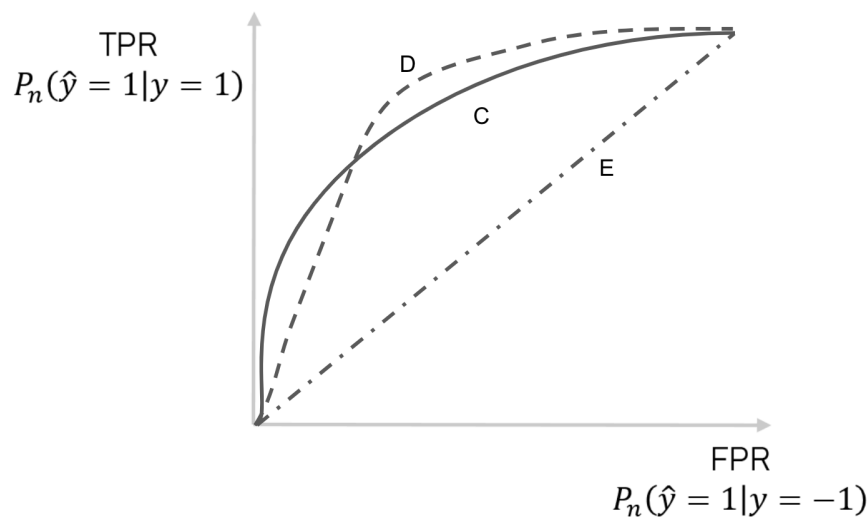


Figure 2: ROC curve for 3 different classifiers.

(g) ☒ For any of the classifiers, could \hat{f} be independent of y ?

1. No. 2. Yes: C. 3. Yes: D. 4. Yes: C and D. 5. Yes: E.

(h) ☒ Out of the 3 classifiers, given that the desired trade-off between FPR and TPR is unknown, what set contains all classifiers that *could* be the optimal classifier (only considering TPR and FPR as metrics)?

1. {C} 2. {D}. 3. {E} 4. {C, D} 5. {C, D, E}

Exercise 2: Ridge Regression

In the first homework, you have extensively worked with linear regression, i.e., the hypothesis space is the set of all affine functions. The goal of this exercise is to study the regularized version of this problem. Let $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ are the training data given to you. The ridge regression optimization problem with parameter $\lambda > 0$ is given by

$$\arg \min_w L_{\text{ridge}}(w) = \arg \min_w \left[\sum_{i=1}^n (y_i - w^\top x_i)^2 + \lambda \|w\|^2 \right]. \quad (1)$$

In this exercise, the $n \times d$ matrix $X \in \mathbb{R}^{n \times d}$ denotes a matrix with the x_i as its rows and the vector $y \in \mathbb{R}^n$ consisting of the scalars y_i . Moreover, here, $\|\cdot\|$ is always the Euclidean norm. We refer to any w_{ridge}^* that attains the above minimum as a solution to the problem.

- (a) Show that L_{ridge} has a positive semi-definite Hessian.
- (b) A continuously differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is called α -strongly convex for some $\alpha > 0$, if for any points $x, y \in \mathbb{R}^d$ one has

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\alpha}{2} \|y - x\|^2.$$

If f is twice differentiable, an equivalent condition is that for any point $x \in \mathbb{R}^d$, one has

$$D^2 f(x) \succeq \alpha I,$$

which means $D^2 f(x) - \alpha I$ is positive semi-definite for all $x \in \mathbb{R}^d$. Prove that a strongly convex function admits a unique minimizer in \mathbb{R}^d .

Hint: This is not an easy exercise. First prove that $f(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$ to show that there is some minimizer.

- (c) Use (b) to show that (1) admits the unique solution w_{ridge}^* for any matrix X .
- (d) What is the role of the term $\lambda \|w\|^2$ in L_{ridge} ? What happens to w_{ridge}^* as $\lambda \rightarrow 0$ and $\lambda \rightarrow \infty$? You do not need to give a complete proof, only an intuitive answer suffice.

Exercise 3: Subgradients and the Lasso

Recall the linear regression problem, where $X \in \mathbb{R}^{n \times d}$ and $\mathbf{y} \in \mathbb{R}^n$. The lasso problem is a penalized linear regression formulated as

$$\arg \min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_1. \quad (2)$$

Here, $\|\mathbf{w}\|_1$ is the ℓ_1 norm, defined as $\|\mathbf{w}\|_1 = |w_1| + \dots + |w_d|$. The goal of this exercise is to characterize the optimal solution of (2).

It is evident that objective (2) is convex in \mathbf{w} (if you do not see this, try to convince yourself). Hence, all local minimizers are going to be global minimizers. A very easy way to find the local minima is to set the gradient of the objective to zero. But wait... what is the gradient of $\|\mathbf{w}\|_1$? You guessed correct: at the point $\mathbf{0}$, $\|\cdot\|_1$ is *not* differentiable. However, one can construct a vector that *works* like a gradient, called a *subgradient*. Formally, a subgradient of a convex function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ at the point \mathbf{x} is a vector \mathbf{p} such that for any point $\mathbf{z} \in \mathbb{R}^d$,

$$f(\mathbf{z}) \geq f(\mathbf{x}) + \langle \mathbf{p}, \mathbf{z} - \mathbf{x} \rangle. \quad (3)$$

The set of all subgradients at the point \mathbf{x} is denoted by $\partial f(\mathbf{x})$. It can be shown that if f is differentiable at the point \mathbf{x} , there is only one subgradient, that is, $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$.

- (a) For $f(x) = |x|$ find the subgradients at the point $x = 0$.
- (b) Use the previous part to find the subgradients at the point $\mathbf{x} = \mathbf{0}$ for the function $f(\mathbf{x}) = \|\mathbf{x}\|_1$.

A nice result in convex optimization tells us that \mathbf{x} is a local minimum of the convex function f if and only if $\mathbf{0} \in \partial f(\mathbf{x})$. Observe that this is a generalization of the differentiable case.

- (c) Find all the subgradients of the objective (2). Check that the optimum value \mathbf{w}^* of the optimization problem satisfies

$$\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w}^*) = \lambda \mathbf{p},$$

where \mathbf{p} is a subgradient of $\|\mathbf{x}\|_1$ at the point \mathbf{w}^* (see (3)).

For the rest of this exercise, we treat the one-dimensional case ($d = 1$). Similar arguments can be made for the general case, but we do not cover those for simplicity. Hence, our optimization problem becomes

$$\arg \min_{w \in \mathbb{R}} \frac{1}{2} \sum_{i=1}^n (x_i w - y_i)^2 + \lambda |w|.$$

- (d) With the method of subgradients and using the results you have for steps 1 and 3, find the optimal w^* . Inspect your answer. Do you see a thresholding effect? Try to explain why this phenomenon happens.
- (e) **(Optional)** Try to solve the problem for general $d > 1$.

Exercise 4: Model selection and regularization.

4.1 Validation Sets

Assume that you have access to a dataset $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ of $n = 10\,000$ data samples (x_i, y_i) that are drawn i.i.d. (independently and identically distributed) from some (unknown) distribution $p(\mathbf{x}, y)$.

You now need to decide how to split this dataset into a training set $\mathcal{D}_{\text{train}}$ and a validation set \mathcal{D}_{val} so that you can run the following standard procedure to learn and evaluate a regression model:

Step 1: Training the regression model on $\mathcal{D}_{\text{train}}$ by minimizing the training loss


$$\hat{f}_{\mathcal{D}_{\text{train}}} = \arg \min_f \left(\frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{train}}} (y_i - f(x_i))^2 \right). \quad (4)$$

Step 2: Estimating the generalization error of the learned model $R(\hat{f}_{\mathcal{D}_{\text{train}}})$ by computing the empirical error on \mathcal{D}_{val} defined as

$$\hat{R}_{\mathcal{D}_{\text{val}}}(\hat{f}_{\mathcal{D}_{\text{train}}}) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{val}}} (y_i - \hat{f}_{\mathcal{D}_{\text{train}}}(x_i))^2. \quad (5)$$

Remember that for a fixed estimator $f(\mathbf{x})$, the generalization error is defined as:

$$R(f) = \mathbb{E}_{(x, y) \sim p} [(y - f(\mathbf{x}))^2].$$

 Decide whether the following statements are True or False.


- $\hat{R}_{\mathcal{D}_{\text{val}}}(\hat{f}_{\mathcal{D}_{\text{train}}})$ is more likely to provide a better estimate of
- (a) the true generalization error $R(\hat{f}_{\mathcal{D}_{\text{train}}})$, when using a validation set of size 500 as opposed to a validation set of size 1000. ☐ True ☐ False

- Choosing a training set of size 1000 is more likely to provide
- (b) a model $\hat{f}_{\mathcal{D}_{\text{train}}}$ that has a lower true generalization error $R(\hat{f}_{\mathcal{D}_{\text{train}}})$ compared to training set of size 2000. ☐ True ☐ False

- The prediction error on the training set is always less than or equal to the prediction error on the validation set, i.e.,
- (c) $\hat{R}_{\mathcal{D}_{\text{train}}}(\hat{f}_{\mathcal{D}_{\text{train}}}) \leq \hat{R}_{\mathcal{D}_{\text{val}}}(\hat{f}_{\mathcal{D}_{\text{train}}})$. ☐ True ☐ False

4.2 Cross-Validation

Now suppose that we used cross-validation to determine what algorithm M (e.g., SGD with which parameters) to use in order to fit f to dataset D .

 Decide whether the following statements are True or False.

- (a) Leave One Out Cross-validation (LOOCV) is used to estimate generalization performance of a prediction function. ☐ True ☐ False
- (b) Leave One Out Cross-validation (LOOCV) is used to compute training error. ☐ True ☐ False
- (c) Performing Leave One Out Cross-validation (LOOCV) on the same dataset multiple times will always yield the same result if M deterministically produces f given D . ☐ True ☐ False

4.3 Akaike Information Criterion

Define the **Akaike Information Criteria** as $AIC = 2k - 2\ln(\hat{L})$ where \hat{L} is the likelihood of f on D_{train} . k is some measure of the number of parameters used to fit f . AIC can be used for model selection by selecting models with low AIC score.

[📌] Decide whether the following statements are True or False.

- (a) AIC is an estimator used to compute generalization error. ☐ True ☐ False
- (b) AIC encourages overfitting. ☐ True ☐ False

4.4 Regularization

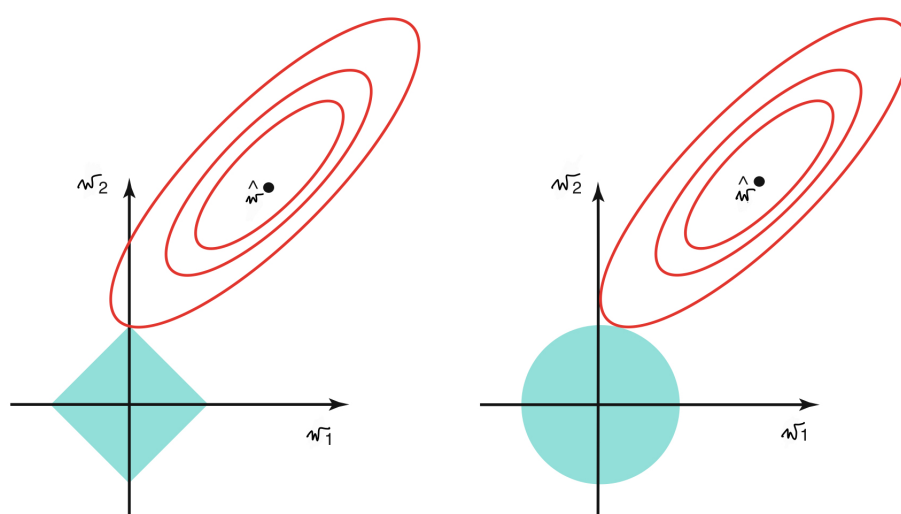


Figure 3: Blue diamond (left) and circle (right) represent constraints for the weights of f . Red curves represent contours for the loss function. That is, points with the same loss value lie on the same red curve. \hat{w} achieves the minimum training loss for the unconstrained problem. The solution to the constrained regression problem is pictured as the intersection of the contour with the constraint set, since this is the lowest loss achievable within the constraint set.

Here instead of computing a prediction function based upon $\sum_{x,y \in D_{\text{train}}} l(f(x), y) + \lambda \text{Reg}(f)$ where Reg is a regularizer, we do regression by optimizing $\sum_{x,y \in D_{\text{train}}} l(f(x), y)$ subject to the constraint $\text{Reg}(f) = \eta$.

[📌] Now answer the following questions.

- (a) Choose which plot in Figure 3 corresponds to the ridge regression
1. left 2. right
- (b) Choose which plot in Figure 3 corresponds to the lasso regression
1. left 2. right

Exercise 5: Support Vector Machines

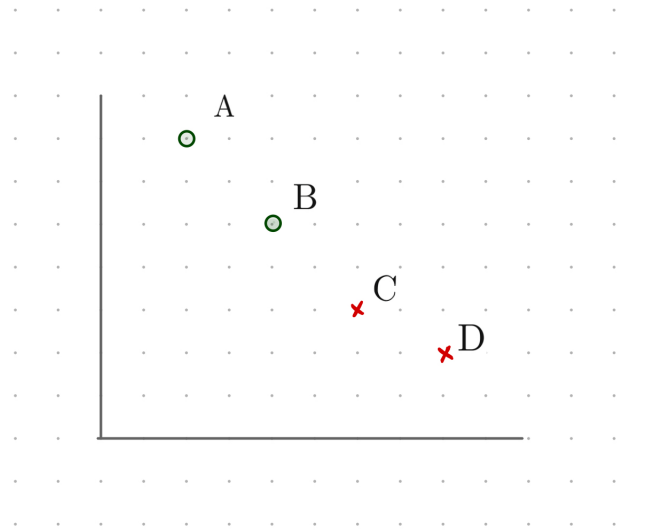


Figure 4: The four points $A = (1, 3.5)$, $B = (2, 2.5)$, $C = (3, 1.5)$, $D = (4, 1)$ are from two different classes.

Figure 4 illustrates a training dataset with 4 points A, B, C, D where 1 or -1 indicates the class to which the point belongs. The data is linearly separable and we want to find maximum margin hyperplane that would divide the points.

- Derive the equation $w_1x_1 + w_2x_2 + b = 0$ for the hyperplane corresponding to the maximum margin classifier.
- Compute the margin of the maximum margin classifier to two decimal places.
- Graphically illustrate a binary classification dataset which, with hard SVM, would have the maximum possible number of support vectors in each class.
- Consider a general setup with input $\{(x_i, y_i)\}_{i=1}^N$, where $y_i = 1$ or $y_i = -1$, and the data is linearly separable. As you have computed above for 2D case, maximum margin classifier aims at a hyperplane that (a) separates the two classes of data and (b) the distance between them is as large as possible. Now, write down the objective of maximum margin classifier in terms of $\|w\|$. What is the relation between the value of the margin and $\|w\|$?
- In the tutorial, we will discuss soft SVM and how to compute subgradients with respect to the objective used. Write a pseudo-code implementing SGD for soft SVM. The inputs are the data points $\{(x_i, y_i)\}_{i=1}^N$, regularization constant λ , batch size b , learning rates $\{\eta_t\}_{t \in \mathbb{N}}$, and number of iterations T . Explicitly write the loss you are minimizing and the closed form of the gradients you are computing at each descent step.