WRITE FIRST NAME, LAST NAME, AND ID NUMBER ("MATRICOLA") BELOW AND READ ALL INSTRUCTIONS BEFORE STARTING WITH THE EXAM! TIME: 2.5 hours.

FIRST NAME: .....

LAST NAME: .....

ID NUMBER: .....

INSTRUCTIONS

- solutions to exercises must be in the appropriate spaces, that is:
  - Exercise 1: pag. 1, 2, 3
  - Exercise 2: pag. 4, 5, 6
  - Exercise 3: pag. 7, 8, 9
  - Exercise 4: pag. 10, 11, 12

  Solutions written outside the appropriate spaces (including other papersheets) will not be considered.
- the use of notes, books, or any other material is forbidden and will make your exam invalid;
- electronic devices (smartphones, calculators, etc.) must be turned off; their use will make your exam invalid;
- this booklet must be returned in its entirety.

Exercise 1 [8 points]

Consider the regression problem when the training data is $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_m, y_m)\}$ with $\mathbf{x}_i = [x_{i,1}, x_{i,2}] \in \mathbb{R}^2$ and $y_i \in \mathbb{R}$ for $i = 1, \dots, m$.

1. Formally define the problem when the hypothesis class is $\mathcal{H} = \{h(\mathbf{x}) \text{ s.t. } h(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2; w_0, w_1, w_2 \in \mathbb{R}\}$ and the squared loss is used. In particular, describe what the goal is.

2. Describe ℓ_2 regularization within the context described above. Given an hypothesis $h \in \mathcal{H}$, let $L_S(h)$ be its training error (i.e., the average loss on the training set), $J(h) = L_S(h) + \lambda R(h)$ be the regularized training error, where $R(h)$ is the ℓ_2 regularization function. Formally define the regularization function for ℓ_2 regularization and *derive* the hypothesis that minimizes the ℓ_2 regularized training error.

3. Given an hypothesis h, let $\mathcal{L}(h)$ be the true (or generalization) error of h. Let h_S be the hypothesis that, given data S, minimizes the regularized training error $J(h)$. Plot the typical behavior of $L_S(h_S)$ and $\mathcal{L}(h_S)$ as a function of $\lambda \geq 0$, and describe how this is linked to overfitting.

[Solution: Exercise 1]

1) Given the training data $S = \{(\vec{x}_1, y_1), \dots (\vec{x}_m, y_m)\}$, with $\vec{x}_i \in \mathbb{R}^2$, $y_i \in \mathbb{R}$ $\forall\, i = 1, \dots, m$, our goal is to find an hypothesis $\hat{h} \in \mathcal{H}$ of low generalization error, that is, we want $\hat{h}$ such that

$$\mathbb{E}_{(\vec{x},y)\sim D}\left[\ell(\hat{h}, (\vec{x}, y))\right] \text{ is low, where}$$

$\ell(\hat{h}, (\vec{x}, y)) = (\hat{h}(\vec{x}) - y)^2$ and D is the (unknown) probability distribution from which $(\vec{x}_i, y_i) \in S$, $i = 1, \dots, m$ have been drawn (independently).

[Solution: Exercise 1]

2) l_2 regularization in the context above corresponds to ridge regression. In particular, the design matrix is

$$X = \begin{bmatrix} - & \vec{x}_1^{\,1} & - \\ - & \vec{x}_2^{\,1} & - \\ & \vdots & \\ - & \vec{x}_m^{\,1} & - \end{bmatrix}, \text{ where } \vec{x}_i^{\,1} = [1, x_{i,1}, x_{i,2}] \text{ and } \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

Let $\vec{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$. Since an hypothesis $h \in \mathcal{H}$ is defined by the corresponding vector $\vec{w}$, we have that the training error is

$$L_S(h) = L_S(\vec{w}) = \frac{1}{m} (\vec{y} - X\vec{w})^T (\vec{y} - X\vec{w}), \text{ and}$$

minimizing $L_S(h)$ (or $L_S(\vec{w})$) corresponds to minimize $(\vec{y} - X\vec{w})^T (\vec{y} - X\vec{w})$. In ridge regression, the regularization function $R(h) = \|\vec{w}\|^2$, therefore the hypothesis that minimizes the regularized training error $L_S(h) + \lambda R(h)$ is

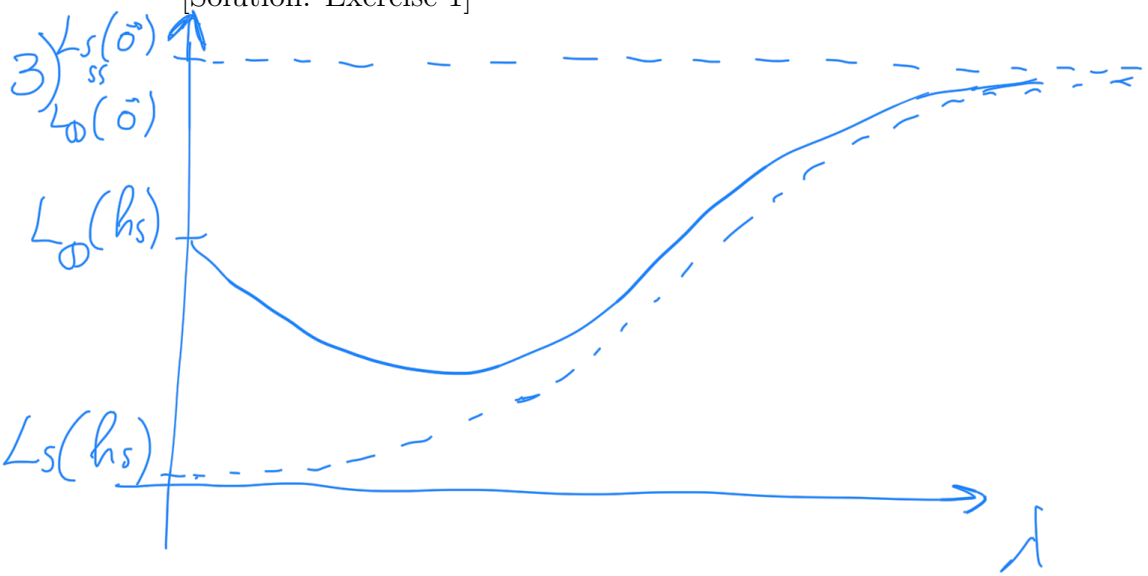
$$\arg\min_{\vec{w}} (\vec{y} - X\vec{w})^T (\vec{y} - X\vec{w}) + \lambda \vec{w}^T \vec{w}.$$ To find it, we compute the gradient and find $\vec{w}$ such that it is zero.

$$\frac{\partial (L_S(\vec{w}) + \lambda R(\vec{w}))}{\partial \vec{w}} = 2\lambda \vec{w} - 2X^T(\vec{y} - X\vec{w}) = 0$$

$$\iff (\lambda I + X^T X)\vec{w} = X^T \vec{y} \iff \vec{w} = (\lambda I + X^T X)^{-1} X^T \vec{y}$$

Note that $\lambda I + X^T X$ is positive definite, thus invertible.

[Solution: Exercise 1]



For low values of λ, we are ignoring the complexity of h_S and only care about minimizing the training error, thus we may find h_S with low training error $L_S(h_S)$ but high generalization error $L_{\mathcal{D}}(h_S)$, which corresponds to overfitting. Increasing λ we take the complexity of h_S more and more into account, and for some value of λ we may have a good balance between training error and complexity, thus providing an hypothesis h_S with lower generalization error.

Exercise 2 [8 points]

Consider the classification problem in machine learning.

1. Provide a formal definition, describing data, loss functions, classification rules etc.

2. Assuming inputs (or features) $x \in \mathbb{R}$, and consider the model class, which is a modified version of logistic regression, defined as the set of models obtained composing the sigmoid function

$$\frac{1}{1+e^{-z}}$$

with the function

$$z = h_{\mathbf{w}}(x) = w_1 + w_2 x^2$$

where the parameters are $\mathbf{w} = [w_1,\ w_2]^\top \in \mathbb{R}^2$. Assume the loss is $\ell(h,(x_i,y_i)) = (y_i - h(x_i))$ if $y_i = 1$, and $\ell(h,(x_i,y_i)) = h(x_i)$ if $y_i = 0$. Write the stoochastic gradient descent update for learning this model from data (x_i,y_i), $i = 1,..,m$.

[Solution: Exercise 2]

1) The classification problem is a supervised learning problem, with
- domain set X, that is the set of all possible objects to make predictions about, where a domain point $\vec{x} \in X$ is called instance, and is usually represented by a vector of features
- label set Y, that defines the set of possible labels. Y is a discrete set in classification. For example, in binary classification $Y = \{-1, 1\}$ or similar.

(, and often finite,)

Given training data $S = ((\vec{x}_1, y_1), \dots, (\vec{x}_m, y_m))$, with $\vec{x}_i \in X$, $y_i \in Y$ $\forall\ i = 1,\dots,m$, we need to choose an hypothesis class H, which defines the possible models or classification rules

[Solution: Exercise 2]

we can pick to make prediction, and a loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^+$, with $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, that given an hypothesis $h \in \mathcal{H}$ provides a measure of how much we lose by predicting the label $h(\vec{x})$ for $\vec{x}$ instead of the (correct) label y. The goal is then to find an hypothesis $h \in \mathcal{H}$ with low generalization error, defined as:

$$L_{\mathcal{D}}(\hat{h}) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(\hat{h}, z)]$$

where $\mathcal{D}$ is the unknown probability distribution over $\mathcal{Z}$ from which $(\vec{x}_i, y_i) \in S$, $i = 1, \dots, m$, have been drawn (as independent samples).

2) In general, for stochastic gradient descent (SGD), let $\vec{w}^{(t)}$ be the weights defining the model in iteration t. Then the update rule is given by:

- pick $(\vec{x}_i, y_i) \in S$ uniformly at random
- $\vec{w}^{(t+1)} \leftarrow \vec{w}^{(t)} - \eta \nabla \ell(\vec{w}^{(t)}, (\vec{x}_i, y_i))$

We need to compute $\nabla \ell(\vec{w}^{(t)}, (\vec{x}_i, y_i))$ for specific model class and loss we are considering. Each model h in our model class is a function of the form: $h(x) = \frac{1}{1 + e^{-(w_1 + w_2 x^2)}}$. Since the loss depends on the value of y_i, the gradient will depend on y_i as well.

[Solution: Exercise 2]

Let's consider the two cases:

i) $y_i = 1 \Rightarrow \nabla \ell(h(x), (x,y)) = \left[\frac{\partial \ell}{\partial w_1}, \frac{\partial \ell}{\partial w_2}\right]^T$, and:

$$\frac{\partial \ell}{\partial w_1} = \frac{\partial z}{\partial w_1} \cdot \frac{\partial \ell}{\partial z} = 1 \cdot \frac{\partial}{\partial z}\left(1 - \frac{1}{1+e^{-z}}\right) = -\frac{e^z}{(1+e^z)^2}, \text{ with } z = w_1 + w_2 x^2$$

$$\frac{\partial \ell}{\partial w_2} = \frac{\partial z}{\partial w_2} \cdot \frac{\partial \ell}{\partial z} = x^2 \cdot \left(-\frac{e^z}{(1+e^z)^2}\right) = -x^2 \frac{e^z}{(1+e^z)^2}$$

ii) $y_i = 0 \Rightarrow \nabla \ell(h(x), (x,y)) = \left[\frac{\partial \ell}{\partial w_1}, \frac{\partial \ell}{\partial w_2}\right]^T$, and:

$$\frac{\partial \ell}{\partial w_1} = \frac{\partial z}{\partial w_1} \cdot \frac{\partial \ell}{\partial z} = 1 \cdot \frac{\partial}{\partial z}\left(\frac{1}{1+e^{-z}}\right) = \frac{e^z}{(1+e^z)^2}$$

$$\frac{\partial \ell}{\partial w_2} = \frac{\partial z}{\partial w_2} \cdot \frac{\partial \ell}{\partial z} = x^2 \frac{e^z}{(1+e^z)^2}.$$

Therefore the SGD update rule is:

- pick $(x_i, y_i) \in S$ uniformly at random;
- $z \leftarrow w_1^{(t)} + w_2^{(t)} x_i^2$
- if $y_i = 1$ then $\vec{w}^{(t+1)} \leftarrow \vec{w}^{(t)} + \eta \begin{bmatrix} e^z/(1+e^z)^2 \\ x^2 e^z/(1+e^z)^2 \end{bmatrix}$;

else $\vec{w}^{(t+1)} \leftarrow \vec{w}^{(t)} - \eta \begin{bmatrix} e^z/(1+e^z)^2 \\ x^2 e^z/(1+e^z)^2 \end{bmatrix}$; // $y_i = 0$

$$\frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z} \Rightarrow \frac{d}{dz}\left(\frac{e^z}{1+e^z}\right) = \frac{e^z(1+e^z) - e^z e^z}{(1+e^z)^2}$$

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}; \quad \frac{d}{dz} e^z = e^z$$

$$= \frac{e^z}{(1+e^z)^2}$$

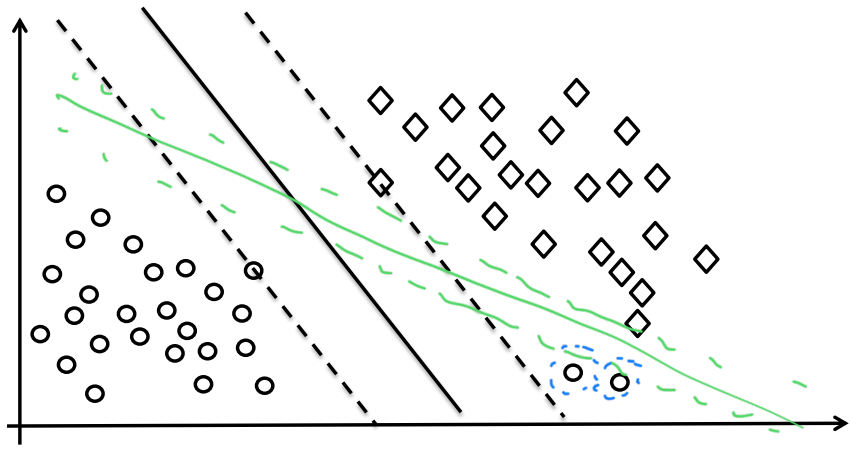
Exercise 3 [8 points]

The Soft-SVM classifier aims at minimizing the following function: $\lambda||\mathbf{w}||^2 + \frac{1}{m}\sum_i \xi_i$.

1. Briefly explain how the Soft-SVM classification method works and which are the constraints under which the function has to be minimized.

2. The figure shows the results of a binary classification performed using a Soft-SVM model with parameter $\lambda = 1$. The training samples are the circles and diamonds and the two shapes correspond to the two classes to which the samples belong. The solid line is the computed separating hyperplane, while the dotted lines represent the margins. For which points ξ_i is different from 0?

3. Does the margin increase or decrease when λ decreases? Guess how the solution changes when a very small value for the λ parameter (i.e., $\lambda \approx 0$) is used, and draw an estimate of the separating hyperplane that could be obtained in this case.



[Solution: Exercise 3]

1) Soft-SVM is similar to hard-SVM, but can be used also when the training data is not linearly separable (i.e., the training data cannot be perfectly classified with a linear model). Soft-SVM finds a model of large margin while allowing some points of the training set to be inside the margin or wrongly classified. This is obtained by

[Solution: Exercise 3]

adding slack variables ξ_i to the constraints of hard-SVM. The constraints for soft-SVM are then:

- $\xi_i \geq 0$ for each $i = 1, \dots, m$
- $y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1 - \xi_i$ for each point $(\vec{x}_i, y_i)$, $i = 1, \dots, m$ in the training set, where $\vec{w}$ and b define the model. The interpretation of ξ_i is the following: if $\xi_i = 0$, then $\vec{x}_i$ is correctly classified and outside the margin; if $0 < \xi_i < 1$, then $\vec{x}_i$ is correctly classified but inside the margin. If $\xi_i \geq 1$, then $\vec{x}_i$ is uncorrectly classified. The function optimized by soft-SVM considers both the margin and the "violation" of the hard-SVM given by ξ_i.

2) See the points circled in blue in the figure. Since they are not correctly classified, the corresponding ξ_i's are > 0.

3) When λ decreases, the term $\frac{1}{m}\sum_{i=1}^{m} \xi_i$ becomes more important in the function minimized by soft-SVM, therefore a model that reduces the values of ξ_i's is sought, even if the margin

---

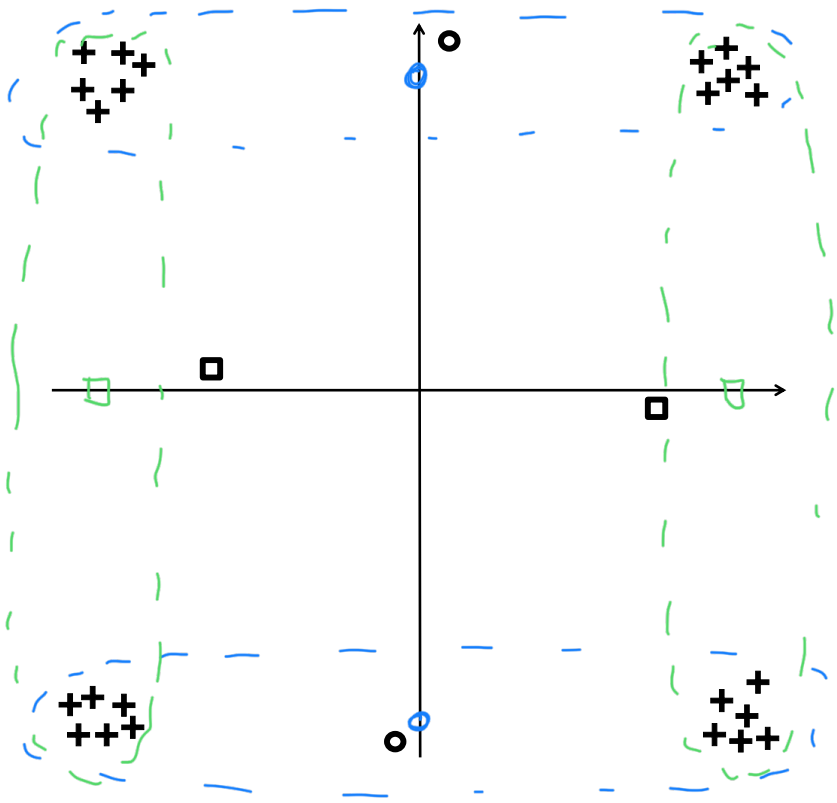
[Solution: Exercise 3]

is smaller. In particular, for the dataset shown in the figure, when $\lambda \approx 0$ the margin is very small, since there is a linear model (shown in green in the figure) of small margin but that correctly classifies all points in the training set, so that $\xi_i = 0 \ \forall i = 1, \dots, m$.

Exercise 4 [8 points]

1. Briefly introduce the clustering problem.
2. Define and explain the cost function used in K-means clustering.
3. Consider the data in the figure below where each point $\mathbf{x} \in \mathbb{R}^2$ is represented by a cross. Show the results (i.e., draw approximately the final centroid locations and the final assignment of the points to the clusters) of clustering into $k = 2$ clusters with K-means when
   (a) the initial centers for the algorithm are the circles;
   (b) the initial centers for the algorithm are the squares.

   Is one solution *significantly* better than the other one? Briefly motivate your answer.



[Solution: Exercise 4]

1) Clustering is the problem of grouping a set of objects such that similar objects end up in the same group and dissimilar objects are separated into different groups.

[Solution: Exercise 4]

More formally, the problem has the following input and output:

Input: set $\mathcal{X}$ of objects and a distance function $d: \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$

Output: a partition of $\mathcal{X}$ into clusters, that is $C = (C_1, C_2, \dots, C_k)$ such that:

- $\bigcup_{i=1}^{k} C_i = \mathcal{X}$
- for all $i \neq j$: $C_i \cap C_j = \emptyset$

(Sometimes the input includes the number k of clusters.)

The partition to be produced in output depends on the specific definition of the problem, and is sometimes captured by defining a cost for a given clustering.

2) Let the input data points be $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m$, with $\vec{x}_i \in \mathbb{R}^d$ for $i = 1, \dots, m$. The k-means cost function is

$$\sum_{i=1}^{k} \sum_{\vec{x} \in C_i} d(\vec{x}, \vec{\mu}_i)^2$$

where $C_1, \dots, C_k$ are the clusters, $d(\cdot,\cdot)$ is the Euclidean distance in $\mathbb{R}^d$, and $\vec{\mu}_i$ is the center

[Solution: Exercise 4]

of cluster C_i for $i = 1, \dots, k$ (the centers are part of the output). Therefore, the cost of a clustering for k-means is defined as the sum of the squares of the distances of each point to the center of the cluster it belongs to.

3) The solutions are shown in the figure. There is no solution that is significantly better than the other one, since the data consists of 4 groups of points that are essentially symmetric with respect to the origin, and the 2 solutions are just 2 different ways to group them into 2 groups. Also, the silhouette coefficient of the 2 solutions is probably very similar.