Student name

MA2823: Foundations of Machine Learning
Final Exam – Solutions
December 16, 2016

Instructor: Chloé-Agathe Azencott

Multiple choice questions

1. (1 point) Taking a bootstrap sample of n data points in p dimensions means:
   - ◯ Sampling p features with replacement.
   - ◯ Sampling $\sqrt{p}$ features without replacement.
   - ◯ Sampling n samples with replacement.
   - ◯ Sampling $k < n$ samples without replacement.

   Solution: Sampling n samples with replacement.

2. (2 points) Which of the following statements are true?
   - ◯ Training a k-nearest-neighbors classifier takes more computational time than applying it.
   - ◯ The more training examples, the more accurate the prediction of a k-nearest-neighbors.
   - ◯ k-nearest-neighbors cannot be used for regression.
   - ◯ A k-nearest-neighbors is sensitive to outliers.

   Solution: False. True. False. True.

3. (4 points) Check all the binary classifiers that are able to correctly separate the training data (circles vs. triangles) given in Figure 1.
   - ◯ Logistic regression
   - ◯ SVM with linear kernel
   - ◯ SVM with RBF kernel
   - ◯ Decision tree
   - ◯ 3-nearest-neighbor classifier (with Euclidean distance).

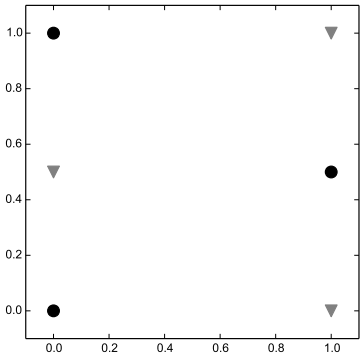


Figure 1: Training data for Question 3.

Solution:

- Logistic regression and linear SVM: linear decision functions, hence no.
- SVM with RBF kernel: yes.
- 3-NN: the 3 nearest neighbors of any point in our training set are 1 of the same class and 2 of the opposite class, hence 3-NN will be systematically wrong.
- DT: yes, you can partition the space with lines orthogonal to the axes in such a way that every sample ends up in a different region.

Short questions

4. (1 point) In a Bayesian learning framework, what is a posterior?

Solution: The updated probability $p(\theta|\mathcal{D})$ of a model, after having seen the data.

5. (1 point) Give an example of a loss function for classification problems.

Solution: cross-entropy; hinge loss; number of errors; etc.

6. (1 point) Give an example of an unsupervised learning algorithm.

Solution: Dimensionality reduction; PCA; clustering; k-means; etc.

7. (1 point) Pearson's correlation between two variables $\boldsymbol{x}$ and $\boldsymbol{z} \in \mathbb{R}^p$ is given by

$$\rho(\boldsymbol{x}, \boldsymbol{z}) = \frac{\sum_{j=1}^{p}(x_j - \bar{\boldsymbol{x}})(z_j - \bar{\boldsymbol{z}})}{\sqrt{\sum_{j=1}^{p}(x_j - \bar{\boldsymbol{x}})^2}\sqrt{\sum_{j=1}^{p}(z_j - \bar{\boldsymbol{z}})^2}},$$

where $\bar{\boldsymbol{x}} = \sum_{j=1}^{p} x_j$. If the data is centered, why is this also referred to as the cosine-similarity?

Solution: If the data is centered,

$$\rho(\boldsymbol{x}, \boldsymbol{z}) = \frac{\langle \boldsymbol{x}, \boldsymbol{z} \rangle}{||\boldsymbol{x}||.||\boldsymbol{z}||} = \cos\theta$$

where θ is the angle between $\boldsymbol{x}$ and $\boldsymbol{z}$.

8. A decision tree partitions the data space $\mathcal{X}$ in m regions $R_1, R_2, \ldots, R_m$. The function f that associates a label to a data point $\boldsymbol{x} \in \mathcal{X}$ can be written as: $f(\boldsymbol{x}) = \sum_{k=1}^{m} c_k I_{\boldsymbol{x} \in R_k}$, where $I_{\boldsymbol{x} \in R_k}$ is an indicator function, i.e. $I_{\boldsymbol{x} \in R_k} = \mathbf{1}_{\boldsymbol{x} \in R_k} = \begin{cases} 1 & \text{if } \boldsymbol{x} \in R_k \\ 0 & \text{otherwise.} \end{cases}$

   Given a training set $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$ where $\boldsymbol{x}^i \in \mathcal{X}$ for $i = 1, \ldots, n$, and assuming we have an algorithm that allows us to define R_k for $k = 1, \ldots, m$, how does one define $c_k (k = 1, \ldots, m)$ for:

   (a) (1 point) a classification problem ($y^i \in \{0, 1\}$)?

   Solution: c_k is the majority class of training points in R_k

   (b) (1 point) a regression problem ($y^i \in \mathbb{R}$)?

   Solution: c_k is the average label of training points in R_k.

9. (2 points) A data scientist runs a principal component analysis on their data and tells you that the percentage of variance explained by the first 3 components is 80 %. How is this percentage of variance explained computed?

   Solution: The overall variance is computed as the sum of the variances of all variables (i.e. the sum of the diagonal terms of the covariance matrix). The variance explained (or accounted for) by one PC is the variance of this PC (i.e. the diagonal term on the corresponding entry of the covariance matrix of the data projected onto its PCs). The variance explained by the first 3 components is **the sum of the tree first values on the diagonal of the covariance matrix of the data projected onto its PCs.**

10. Assume you are given data $\{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^n, y^n)\}$ where $\boldsymbol{x}^i \in \mathcal{X}$ and $y^i \in \mathbb{R}$. You are planning to train an SVM. You define a kernel k and obtain, on your training data, the kernel matrix K presented in Figure 2, where $K_{ij} = k(\boldsymbol{x}^i, \boldsymbol{x}^j)$.

    (a) (1 point) What is the issue here?

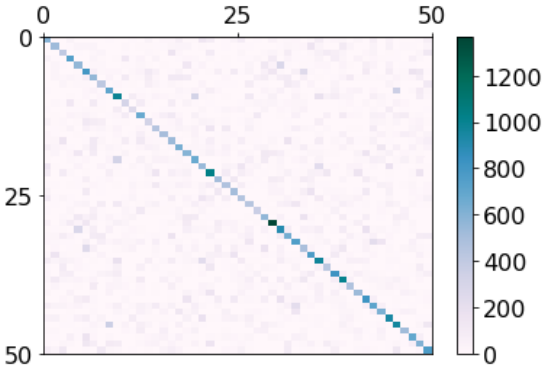


Figure 2: Kernel matrix on the training data for Question 10

Solution: Diagonal dominance: the kernel is equivalent to the identity matrix and the SVM won't learn.

(b) (1 point) How can you address it?

Solution: Normalize the kernel matrix by $K_{ij} \leftarrow \frac{K_{ij}}{\sqrt{K_{ii}K_{jj}}}$, or manipulate a coefficient of your kernel to obtain non-zero off-diagonal terms.

11. (2 points) Assume we are given data $\{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^n, y^n)\}$ where $\boldsymbol{x}^i \in \mathbb{R}^p$ and $y^i \in \mathbb{R}$, and a parameter $\lambda > 0$. We denote by X the $n \times p$ matrix of row vectors $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n$ and $\boldsymbol{y} = (y^1, \ldots, y^n)$. We are also given a graph structure on the features, where vertices are features and edges connect related features. We denote by $\mathcal{E}$ the set of edges of this graph. The graph-Laplacian-regularized linear regression estimator is defined as:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} ||\boldsymbol{y} - X\boldsymbol{\beta}||_2^2 + \lambda \sum_{(u,v) \in \mathcal{E}} (\beta_u - \beta_v)^2.$$

What does the regularizer $\sum_{(u,v) \in \mathcal{E}} (\beta_u - \beta_v)^2$ enforce?

Solution: That connected features get similar weights.

12. Consider a data set described using 1 000 features in total. The labels have been generated using the first 50 features. Another 50 features are exact copies of these features. The 900 remaining features are uninformative. Assume we have 100 000 training data points.

(a) (2 points) How many features will a filtering approach select?

Solution: 100 (50 informative + 50 copies).

(b) (2 points) How many features will a wrapper approach select?

Solution: 50 (only informative features).

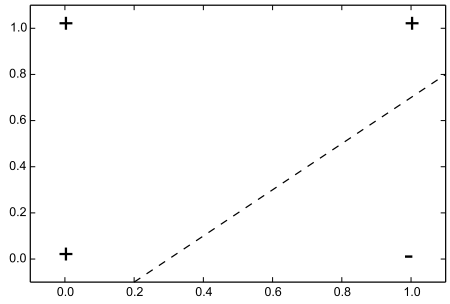
Problems

13. **Perceptron.** Consider the following Boolean function:

x_1	x_2	$y = \neg x_1 \cup x_2$
0	0	1
0	1	1
1	0	0
1	1	1

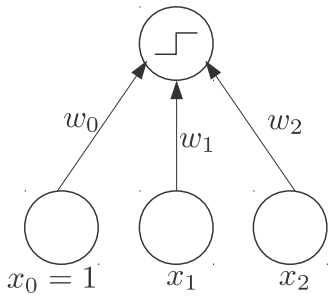
(a) (2 points) Can this function be represented by a perceptron? Explain your answer.

Solution: Yes, because the function is linearly separable.



(b) (4 points) If yes, draw a perceptron that represents it. Otherwise, build a multilayer neural network that will.

Solution: A perceptron has the following architecture:



$w_0 = 1, w_1 = -1, w_2 = 2$

Its output is given by: 1 if $w_0 + w_1 x_1 + w_2 x_2 > 0$ and 0 otherwise.

This is one of many possible solutions. w_0, w_1, w_2 must give the equation of a line that separates (1, 0) from (0, 0), (0, 1) and (1, 1).

14. **Multi-class classification.** Assume p random variables $X_1, \ldots, X_p$, conditionally independent given Y. Y is a discrete random variable that can take one of K values $y_1, \ldots, y_K$, corresponding to K classes. X is boolean.

We suppose that each X_j follows a Bernouilli distribution:

$$P(X_j = u | Y = y_k) = \theta_{jk}^u (1 - \theta_{jk})^{(1-u)}, \ u \in \{0, 1\}.$$

We observe n datapoints $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n$ and their labels $y^1, \ldots, y^n$.

In what follows, you can use the indicator $I_{ik} = \mathbf{1}_{y^i = y_k} = \begin{cases} 1 & \text{if } y^i = y_k \\ 0 & \text{otherwise.} \end{cases}$

We will call n_k the number of training points in class k, and n_{jk} the count of training points in class k for which $x_j = 1$:

$$n_{jk} = \sum_{i=1}^{n} 1_{y^i = y_k} x_j^i.$$

(a) (2 points) What is the likelihood of the parameter θ_{jk}?

Solution: The likelihood of the parameters is given by

$$\begin{aligned} \mathcal{L}(\theta_{jk}) &= \prod_{i=1}^{n} p(X_j = x_j^i | \theta_{jk})^{I_{ik}} \\ &= \prod_{i=1}^{n} \left(\theta_{jk}^{x_j^i} (1 - \theta_{jk})^{(1 - x_j^i)} \right)^{I_{ik}}. \end{aligned}$$

(b) (3 points) What is the maximum likelihood estimator of θ_{jk}?

Solution: The log-likelihood is:

$$l(\theta_{jk}) = \sum_{i=1}^{n} I_{ik} \left[x_j^i \log \theta_{jk} + (1 - x_j^i) \log (1 - \theta_{jk}) \right].$$

Taking the derivative with respect to θ_{jk} and setting it to 0:

$$\frac{\partial l(\theta_{jk})}{\partial \theta_{jk}} = 0,$$

we obtain

$$\frac{1}{\theta_{jk}} \sum_{i=1}^{n} I_{ik} x_j^i + \frac{1}{1 - \theta_{jk}} \sum_{i=1}^{n} I_{ik} (1 - x_j^i).$$

Finally,

$$\hat{\theta}_{jk} = \frac{n_{jk}}{n_k}.$$

For a data point $\boldsymbol{x} = (x_1, \ldots, x_p)$, we can write the Naive Bayes decision rule as:

$$f(\boldsymbol{x}) = \underset{k=1,\ldots,K}{\arg\max} \left(\frac{P(Y = y_k) P(\boldsymbol{x}|Y = y_k)}{\sum_{l=1}^{K} P(Y = y_l) P(\boldsymbol{x}|Y = y_l)} \right).$$

(c) (2 points) When making predictions, we use the rule

$$f(\boldsymbol{x}) = \underset{k=1,\ldots,K}{\arg\max} \left(P(Y = y_k) \prod_{j=1}^{p} P(x_j|Y = y_k) \right).$$

Why?

Solution: Because (i) the independence assumption lets us write

$$P(\boldsymbol{x}|Y = y_k) = \prod_{j=1}^{p} p(x_j|Y = y_k)$$

(ii) the denominator does not depend on k.

(d) (1 point) Given a data point $\boldsymbol{x}$, how can you calculate $P(X = \boldsymbol{x})$ given the parameters estimated by Naive Bayes?

Solution: $P(X = \boldsymbol{x})$ as $\sum_k P(X = \boldsymbol{x}|Y = y_k) P(Y = y_k)$.

15. **Virtual high-througput screening.** Figure 3 presents the performance of several algorithms applied to the problem of classifying molecules in two classes: those that inhibit Human Respiratory Syncytial Virus (HRSV), and those that do not. HRSV is the most frequent cause of respiratory tract infections in small children, with a worldwide estimated prevalence of about 34 million cases per year among children under 5 years of age.

(a) (1 point) Which method gives the best performance?

Solution: Random forests (top line).

(b) (2 points) The goal of this study is to develop an algorithm that can be used to suggest, among a large collection of several million of molecules, those that should be experimentally tested for activity against HRSV. Compounds that are active against HSRV are good leads from which to develop new medical treatments against infections caused by this virus. In this context, is it preferable to have a high sensitivity or a high specificity? Which part of the ROC curve is the most interesting?

Solution: We want a low false positive rate (so as to ensure there are mostly promising compounds among those that will be selected for further development; thera-

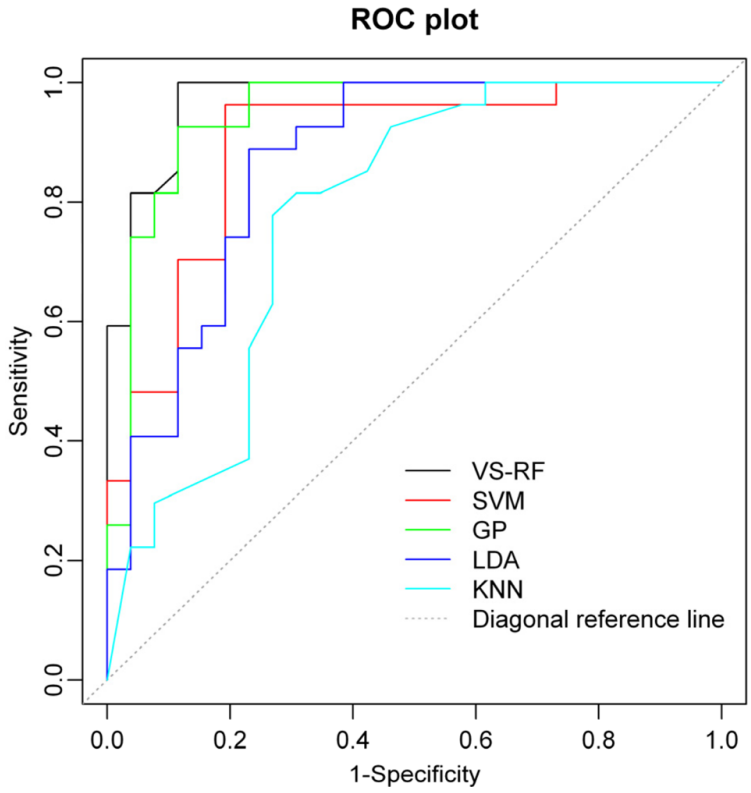


Figure 3: ROC curves for several algorithms classifying molecules according to their action on HRSV, computed on a test set. Sensitivity = True Positive Rate. Specificity = 1 - False Positive Rate. VS-RF : Random Forest. SVM : Support Vector Machine. GP : Gaussian Process. LDA : Linear Discriminant Analysis. kNN : k-Nearest Neighbors. Source: M. Hao, Y. Li, Y. Wang, and S. Zhang, Int. J. Mol. Sci. 2011, 12(2), 1259-1280.

peutic development is costly), i.e. high specificity. We're interested in the left part of the curve: what sensitivity can we get for a fixed specificity?

(c) (1 point) In this study, the authors have represented the molecules based on 777 descriptors. Those descriptors include the number of oxygen atoms, the molecular weights, the number of rotatable bonds, or the estimated solubility of the molecule. They have fewer samples (216) than descriptors. What is the danger here?

Solution: Overfitting.

16. **Kernel ridge regression.** Assume we are given data $\{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^n, y^n)\}$ where $\boldsymbol{x}^i \in \mathbb{R}^p$ is centered and $y^i \in \mathbb{R}$, and a parameter $\lambda > 0$. We denote by X the $n \times p$ matrix of row vectors $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n$ and $\boldsymbol{y} = (y^1, \ldots, y^n)$. The ridge regression estimator is defined as:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} ||\boldsymbol{y} - X\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_2^2.$$

One way to write the solution to this problem is:

$$\hat{\boldsymbol{\beta}} = X^\top (XX^\top + \lambda I)^{-1} \boldsymbol{y}.$$

(a) (1 point) Does this solution always exist? Justify your answer.

Solution: Yes: $(XX^\top + \lambda I)$ can always be inverted if $\lambda > 0$.

(b) (2 points) Write down the value of the prediction for a data point $\boldsymbol{x}' \in \mathbb{R}^p$, as a function of X, $\boldsymbol{y}$ and λ.

Solution:

$$\hat{y} = \hat{\boldsymbol{\beta}}^\top \boldsymbol{x}' = \boldsymbol{y}^\top (XX^\top + \lambda I)^{-1} X \boldsymbol{x}'.$$

(c) (2 points) Let us now replace all data points with their image in a Hilbert space $\mathcal{H}$: $\boldsymbol{x}$ is replaced by $\phi(\boldsymbol{x})$, where $\phi : \mathbb{R}^p \to \mathcal{H}$. Let us define K as the $n \times n$ matrix with entries $K_{ij} = \langle \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}^j) \rangle_{\mathcal{H}}$, and $\boldsymbol{\kappa}$ as the n-dimemsional vector with entries $\kappa_i = \langle \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}') \rangle_{\mathcal{H}}$.
We are now solving the following optimization problem:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} ||\boldsymbol{y} - \Phi\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_2^2,$$

where Φ is the $n \times p$ matrix of row vectors $\phi(\boldsymbol{x}^1), \ldots, \phi(\boldsymbol{x}^n)$.
Write down the value of the prediction for a data point $\boldsymbol{x}' \in \mathbb{R}^p$, as a function of K, κ, $\boldsymbol{y}$ and λ, without using ϕ.

Solution:

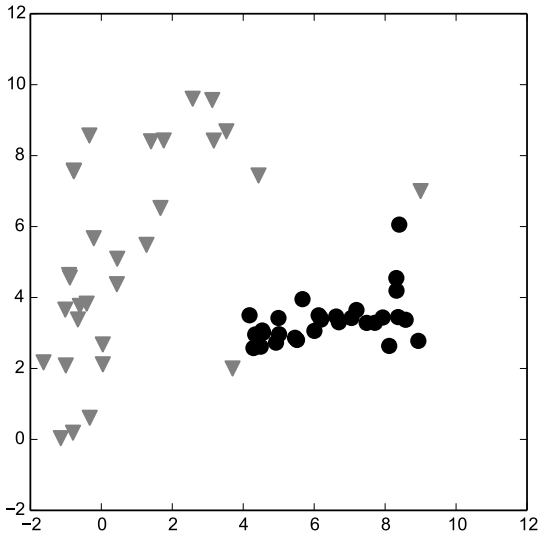
$$\hat{y} = \hat{\boldsymbol{\beta}}^\top \boldsymbol{x}' = \boldsymbol{y}^\top (K + \lambda I)^{-1} \kappa.$$

(d) (2 points) Could the kernel trick be applied in a similar fashion to the l_1-regularized linear regression (Lasso)?

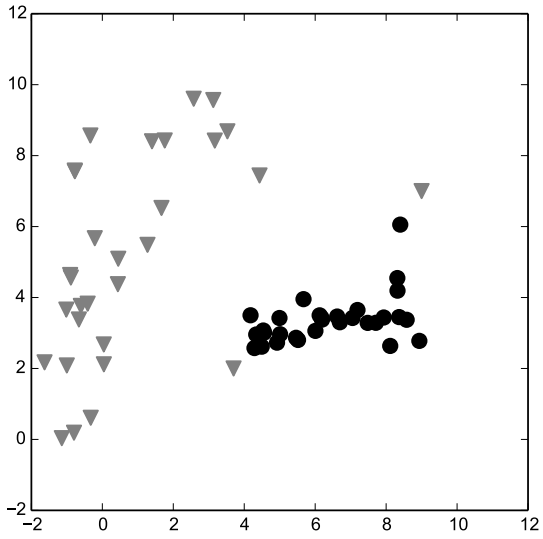
Solution: No, because unlike $||\boldsymbol{w}||_2$, $||\boldsymbol{w}||_1$ cannot be expressed as a dot product.

17. **Quadratic SVM.** We are given the 2-dimensional training data $\mathcal{D}$ shown in Figure 4 for a binary classification problem (circles vs. triangles). Assume we are using an SVM with a **quadratic kernel.** Let C be the cost parameter of the SVM.
Assuming $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$ with $\boldsymbol{x} \in \mathbb{R}^2$ and $y \in \{-1, +1\}$, recall that the SVM is solving the following optimization problem:

$$\begin{aligned} \arg\min_{\boldsymbol{w} \in \mathbb{R}^p, b \in \mathbb{R}} \frac{1}{2}||\boldsymbol{w}||^2 + C \sum_{i=1}^n \xi_i \quad & \text{such that} \\ y^i \left(\langle \boldsymbol{w}, \phi(\boldsymbol{x}^i) \rangle + b \right) &\geq 1 - \xi_i \quad \text{for all } i = 1, \ldots, n \\ \xi_i &\geq 0 \quad \text{for all } i = 1, \ldots, n, \end{aligned}$$



(a) Very large C.



(b) Very small C.

Figure 4: Training data for Question 17.

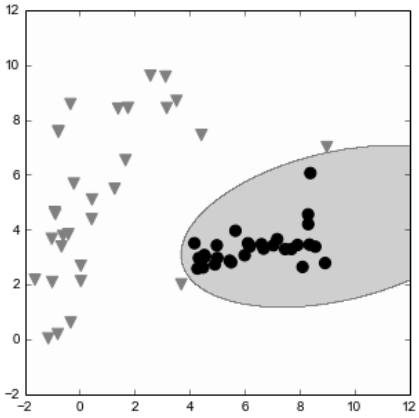
where ϕ is such that $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle = (\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + 1)^2$.

(a) (2 points) On Figure 4 (a), draw the decision boundary for a very large value of C. Justify your answer here.

Solution: The soft-margin formulation of the SVM can be rewritten as

$$\arg\min_f \left(\frac{1}{\text{margin}(f)} + C \times \text{ error}(f) \right).$$

Large C means the classifier makes few errors. Quadratic SVM means the decision boundary is an ellipsoid.

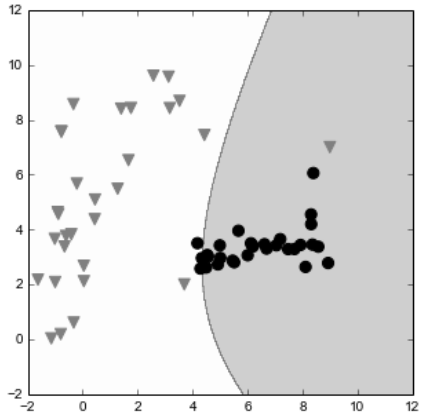


(b) (2 points) On Figure 4 (b), draw the decision boundary for a very small value of C. Justify your answer here.

Solution: The soft-margin formulation of the SVM can be rewritten as

$$\arg\min_f \left(\frac{1}{\text{margin}(f)} + C \times \text{ error}(f) \right).$$

Small C means the classifier has a large margin. Quadratic SVM means the decision boundary is an ellipsoid.



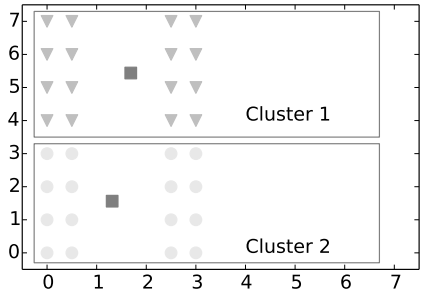
(c) (2 points) Which of the two (large C or small C) do you expect to generalize better and why?

Solution: Small C. The two triangles near the circles are most likely noise/outliers.

18. **K-means clustering.**

(a) (4 points) Consider the unlabeled two-dimensional data represented on Fig. 5. Using the two points marqued as squares as initial centroids, draw (on that same figure) the clusters obtained after one iteration of the k-means algorithm ($k = 2$).

Solution:



(b) (2 points) Does your solution change after another iteration of the k-means algorithm?

Solution: No.

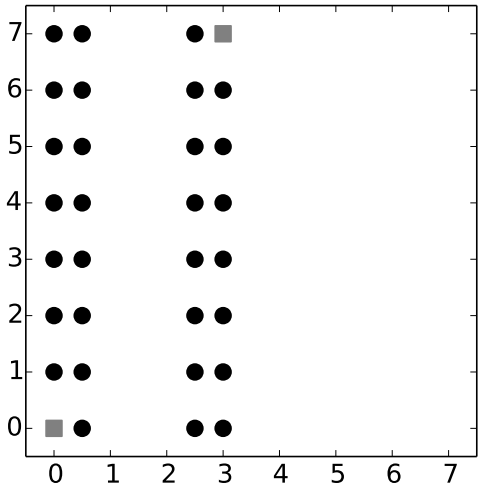


Figure 5: Data for Question 18.

Bonus questions

19. (1 point) In `scikit-learn`, what is the difference between the methods `predict` and `predict_proba` for classifiers?

Solution: `predict` returns a class prediction, while `predict_proba` returns the probabilities to belong to each of the classes.

20. (1 point) Which feature(s) can you use to represent months in such a way that December is equally distant from January and November using the Euclidean distance?

Solution: Map to a circle and use cosine and sine of the angle, i.e. use 2 features $\cos(\frac{\pi k}{6})$ and $\sin(\frac{\pi k}{6})$.