Exam bis MA 2823:
Foundations of Machine Learning

Instructor: Chloé-Agathe Azencott

June 8, 2016

- Exam duration: 3 hours.
- The exam is closed book and notes. No computer, phone, calculator.
- You can answer in English or in French.
- Please write concise (but argumented!) answers.
- No exact numerical computations are required.

Useful formulas:

$\exp(u) = \lim_{n\to\infty} \left(1 + \frac{u}{n}\right)^n$.

The logistic function is defined by $u \mapsto \frac{1}{1+\exp(-u)}$.

The probability density function of the Gaussian distribution $u \sim \mathcal{N}(\mu, \sigma^2)$ is given by:

$$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(u-\mu^2)}{2\sigma^2}\right)$$

This exam has 14 questions, for a total of 42 points. Don't panic!

Question 1 ..... *1 point*

Is the Lasso more likely to overfit if the regularization parameter λ is large or small? Reminder: Given data $\{(x^1, y^1), \ldots, (x^n, y^n)\} \in \mathbb{R}^p \times \mathbb{R}$, Lasso coefficients are computed as the solution to

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} ||y - X\beta||_2^2 + \lambda||\beta||_1.$$

Solution: If λ is small, most of the emphasis is put on the training error and the Lasso is more likely to overfit.

Question 2 ..... *1 point*

What kind of decision boundary can we learn with a decision tree?

Solution: A boundary of which each segment is parallel to the axes.

Question 3 ..... *1 point*

What happens with the predictions of a k-nearest-neighbor algorithm if there are uninformative variables in the data? Why?

Solution: The distance used to determine the nearest neighbors will be wrong (in the sense that it won't relate to the problem) and the k-NN will make mistakes.

Question 4 ..... *1 point*

In which situation would you need to do a *nested* cross-validation, i.e. have a secondary cross-validation inside each train set of a primary cross-validation?

Solution: For a method that has meta-parameters to fit (e.g. The λ parameter of Lasso, the C paramater of SVM, the k parameter of kNN). The inner cross-validation loop is used to set the parameters, the outer cross-validation loop to evaluate the prediction error of the method.

Question 5 ..... *2 points*

What is overfitting, and how can you avoid it?

Solution:

- Model fits the training data too well and cannot generalize.
- Complex model vs. simple model.
- Cross-validation to set model complexity.
- Regularization.

Question 6 ..... *2 points*

What type of strategies can be used to evaluate the quality of the output of a clustering algorithm?

Solution: Three families of strategies:

- Based on the shape of the clusters: cluster tightness/separation, Davies-Bouldin index, silhouette coefficient.
- Based on the stability of the results: under multiple repeats, when perturbating the data with noise or sampling.
- Based on domain knowledge: the clusters match prior knowledge to an extent.

Question 7 ..... *2 points*

You are given an algorithm that can determine whether a person has tuberculosis or not from an X-ray of their lungs. The test is correct 95% of the time. Assume the prevalence of tuberculosis in this population is 1 in $10,000$[1]. If $10,000$ people have been labeled "positive" by this test, how many of them do *not* have tuberculosis?

Solution: Denoting having tuberculosis by T and the test being positive by $+$, let us apply Bayes rule:

$$P(T|+) = \frac{P(T)P(+|T)}{P(+)}$$

[1] In France, the prevalence of tuberculosis in 2013 was of 7.5 cases in 10^5 inhabitants. Note that people who are subjected to a lung X-ray are those that are considered at risk, i.e. come from a population (homeless people, medical staff, recently arrived migrants, ...) with a much higher prevalence.

We can compute

$$\begin{aligned} P(+) &= P(+|T)P(T) + P(+|\bar{T})P(\bar{T}) \\ &= 0.95 * 10^{-4} + (1 - 0.95) * (1 - 10^{-4}) \approx 0.05. \end{aligned}$$

Hence: $P(T|+) = \frac{10^{-4} \times 0.95}{0.05} \approx 0.02$

Finally, $0.98 \times 10^4 = 9800$ of these people do not have tuberculosis.

Question 8 ..... *2 points*

Let us consider n data points, drawn from a normal distribution $\mathcal{N}(\theta, \sigma_0)$, and a prior that $\theta \sim \mathcal{N}(\mu, \sigma)$. If θ_{ML} denotes the maximum likelihood of θ, the Bayes estimator of θ is given by

$$\theta_{\text{Bayes}} = \frac{n/\sigma_0^2}{n/\sigma_0^2 + 1/\sigma^2}\theta_{\text{ML}} + \frac{1/\sigma^2}{n/\sigma_0^2 + 1/\sigma^2}\mu.$$

(a) (1 point) Interpret this formula when the number of data points increases.

Solution: When n increases, the Bayes estimator gets closer to the sample average. It uses more and more information from the sample.

(b) (1 point) What happens if the variance of the prior is low?

Solution: When σ is small, the Bayes estimator gets closer to μ, the mean of the prior. There is little uncertainty about the prior.

Question 9 ..... *3 points*

Suppose you want to build a nearest-neighbor classifier to predict whether a beverage is tea or coffee. You use two features: the volume (in mL) and the caffeine content (in g). You collect the data presented in Table 1.

Volume (mL)	Caffeine (g)	Label
250	0.025	tea
100	0.010	tea
125	0.050	coffee
250	0.100	coffee

Table 1: Tea or coffee? Collected data.

(a) (1 point) Using $k = 1$ and the Euclidean distance, what is the label for a test point with volume=125mL, caffeine=0.015g?

Solution: Calling x our query data point and A, B, C, D the training points,

$$d(x, A)^2 = (250 - 125)^2 + (0.025 - 0.015)^2 = 125^2 + 0.01^2$$
$$d(x, B)^2 = (100 - 125)^2 + (0.010 - 0.015)^2 = 25^2 + 0.005^2$$
$$d(x, C)^2 = (125 - 125)^2 + (0.050 - 0.015)^2 = 0.035^2$$
$$d(x, D)^2 = (250 - 125)^2 + (0.100 - 0.015)^2 = 125^2 + 0.085^2$$

Hence x is closest to C and is labeled as coffee.

(b) (1 point) Why might this be problematic?

Solution: This actually looks more like tea, but volume dominates caffeine content.

(c) (1 point) How would you fix this?

Solution: Normalize the data.

Question 10 ..... *4 points*

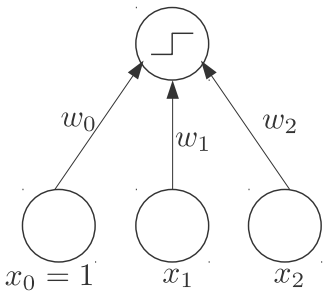
NAND Perceptron. Table 2 gives the decision table for NAND. Give a perceptron that predicts this function.

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

Table 2: Decision table for NAND. x_1 and x_2 are the inputs, and y the output.

(a) (1 point) What is the architecture of a perceptron that predicts NAND?

Solution:



(b) (1 point) What is the form of the output of this perceptron?

Solution: $f(x) = s(w_0 + w_1x_1 + w_2x_2)$ where s is a threshold function.

(c) (2 points) Find weights for this perceptron.

Solution: We are looking for w_0, w_1 and w_2 such that $w_0 + w_1x_1 + w_2x_2 > 0 \Rightarrow y = 1$ and $w_0 + w_1x_1 + w_2x_2 < 0 \Rightarrow y = 0$.
This implies $w_0 > 0$, $w_0 + w_1 > 0$, $w_0 + w_2 > 0$, and $w_0 + w_1 + w_2 < 0$.

$$w_0 = 1.0$$
$$w_1 = -0.75$$
$$w_2 = -0.75$$

is one of many possible solutions.

Question 11 ..... *5 points*

Assume we are given n points $\{x^1, \ldots, x^n\} \in \mathcal{X}^n$. Let us look at PCA in the original space, assuming $\mathcal{X} = \mathbb{R}^p$. Our goal is to find a low-dimensional space such that variance is maximized when the data is projected onto that space. We'll assume the data is centered.

(a) (1 point) Write the projection of $x \in \mathbb{R}^1$ onto the direction $w \in \mathbb{R}^p$. We'll assume w is a unit vector.

Solution: $z = w^\top x$

(b) (1 point) Compute the variance of this projection as a function of w and x.

Solution: Remembering that $\mathbb{E}[x] = 0$ (the data is centered):

$$\begin{aligned}\text{Var}(z) = \text{Var}(w^\top x) &= \mathbb{E}[((w^\top x) - \mathbb{E}[w^\top x])^2] \\ &= \mathbb{E}[(w^\top x)^2] \\ &= w^\top \mathbb{E}[xx^\top] w\end{aligned}$$

(c) (1 point) Denoting by Σ the sample covariance matrix of the data $\{x^1, \ldots, x^n\}$:

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} x^i x^{i\top}$$

write the optimization problem that corresponds to finding the direction w so as to maximize the variance of this projection.

Solution: Σ is an estimator of $\mathbb{E}[xx^\top]$. We can write the problem as finding:

$$\hat{w} = \arg\max_{w \in \mathbb{R}^p} w^\top \Sigma w$$

under the constraint that $||\hat{w}|| = 1$.

(d) (1 point) Prove this problem is equivalent to:

$$\hat{w} = \arg\max_{w} \left(w^\top \Sigma w - \alpha(w^\top w - 1) \right).$$

Solution: The solution lies at a point that is tangent to an iso-contour of $w^\top \Sigma w$ and the surface $||w|| = 1$, i.e. $w^\top w - 1 = 0$. Hence this problem is equivalent to maximizing the Lagrangian $L_\alpha(w) = w^\top \Sigma w - \alpha(w^\top w - 1)$.

(e) (1 point) How can we find $\hat{w}$ now?

Solution: Taking the gradient of the Lagrangian and setting it to 0, we get:

$$2\Sigma\hat{w} - 2\alpha\hat{w} = 0$$

Hence $\Sigma\hat{w} = \alpha\hat{w}$ and $\hat{w}$ is an eigenvector of Σ, with eigenvalue α. For any eigenvector w of Σ, with corresponding eigenvalue α,

$$L_\alpha(w) = w^\top \Sigma w - \alpha(w^\top w - 1) = \alpha.$$

Hence to maximize the Lagrangian we must pick the eigenvector w with the largest eigenvalue α.

Question 12 ..... *5 points*

Assume we are given data $\{(x^1, y^1), \ldots, (x^n, y^n)\}$ where $x^i \in \mathbb{R}^p$ and $y^i \in \mathbb{R}$. We denote by X the $n \times p$ matrix of row vectors $x^1, \ldots, x^n$ and $y = (y^1, \ldots, y^n)$.

Let us assume that the relationship between X and y is linear, that is there exists $\beta \in \mathbb{R}^{p+1}$ such that for all $i \in \{1, \ldots, n\}$:

$$y^i = \sum_{j=1}^{p} \beta_j x_j^i + \beta_0 + \epsilon$$

where ϵ is noise (or error).

(a) (2 points) Assume the noise is Gaussian distributed with mean 0 and standard deviation σ: $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Under this model, write the log-likelihood of β given the data. You can ignore terms that are independent of β (replace them with the notation "Ct" for "constant" if you wish) and replace $\sum_{j=1}^{p} \beta_j x_j^i + \beta_0$ with $f(x|\beta)$.

Solution:

That the noise is Gaussian gives us:

$$y|x \sim \mathcal{N}(f(x|\beta), \sigma^2)$$

We can then write the log-likelihood as:

$$\begin{aligned}
\mathcal{L}(\beta|X) &= \log \textstyle\prod_{i=1}^{n} p(y^i|x^i)p(x^i) \\
&= \log \textstyle\prod_{i=1}^{n} p(y^i|x^i) + \log \prod_{i=1}^{n} p(x^i) \\
&= \log \left(\textstyle\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(y^i - f(x^i|\beta))^2}{2\sigma^2} \right] + \text{Ct} \right) \\
&= \text{Cte} - \tfrac{1}{2\sigma^2} \textstyle\sum_{i=1}^{n} (y^i - f(x^i|\beta))^2
\end{aligned}$$

(b) (2 points) Maximize this log-likelihood. Is the solution unique?

Solution: As per the previous question, maximizing the log-likelihood is equivalent to minimizing the residual sum of squares, which is given by:

$$\begin{aligned}
\text{RSS}(\beta) &= \textstyle\sum_{i=1}^{n} (y^i - f(x^i|\beta))^2 \\
&= \textstyle\sum_{i=1}^{n} \left(y^i - \beta_0 - \sum_{j=1}^{p} x_j^i \beta_j \right)^2 \\
&= (y - X\beta)^\top (y - X\beta)
\end{aligned}$$

This is a quadratic form of β and can be minimized by setting its gradient to zero.

$$\frac{\partial \text{RSS}}{\partial \beta} = -2X^\top (y - X\beta).$$

Hence

$$X^\top y - X^\top X \hat{\beta} = 0$$

If $X^\top X$ can be inverted, the solution is unique and given by

$$\hat{\beta} = (X^\top X)^{-1} X^\top y.$$

Otherwise, the solution is not unique.

(c) (1 point) Why is the setting where there are many more features than samples problematic in linear regression? How can this problem be addressed?

Solution: No closed form solution. Linear regression does not pick out truly important features out of all features. Regularization can be used to address this problem.

Question 13 ..... *6 points*

Let us consider the training data $\{(x^1, y^1), \ldots, (x^n, y^n)\}$ where $x^i \in \mathbb{R}^p$ and $y^i \in \{-1, +1\}$.

The soft-margin SVM uses the decision function

$$f(x) = \langle w, x \rangle + b,$$

and estimates $w \in \mathbb{R}^p$, $b \in \mathbb{R}$ by solving

$$\begin{aligned} \arg \min_{w \in \mathbb{R}^p, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \quad & \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i \\ & \text{s. t. } y^i(\langle w, x^i \rangle + b) \geq 1 - \xi_i \ \ \forall i \in \{1, \ldots, n\} \\ & \xi_i \geq 0 \ \ \forall i \in \{1, \ldots, n\} \end{aligned} \tag{1}$$

(a) (1 point) What is the shape of the decision boundary of the soft-margin SVM?

Solution: A hyperplane.

(b) (1 point) What is the role of the parameter C?

Solution: C is the trade-off between maximizing the margin and making few errors. It can be seen as a bias-variance trade-off, or regularization, parameter.

(c) (1 point) What is the role of the parameters ξ_i?

Solution: The ξ_i are slack variables that quantify the misclassification of training point i. If i is correctly classified, $\xi_i = 0$. If not, ξ_i is the distance from the point to the boundary of the "tube" defining the margin.

To solve Equation 1, we write out its Lagrangian and set its derivative to 0. This yields:

$$w = \sum_i \alpha_i y^i x^i$$

$$\sum_i \alpha_i y^i = 0$$

$$\alpha_i = C - r_i,$$

where α_i, r_i are the Lagrange multipliers.

In addition, the KKT conditions give us that:

1. Either $\alpha_i > 0$ and $y^i(\langle w, x^i\rangle + b) - 1 + \xi_i = 0$
   or $\alpha_i = 0$ and $y^i(\langle w, x^i\rangle + b) - 1 + \xi_i > 0$
2. Either $r_i = 0$ and $\xi_i > 0$
   or $r_i > 0$ and $\xi_i = 0$.

(d) (1 point) Rewrite the decision function using α instead of w.

Solution:

$$f(x) = \sum_{i=1}^{n} \alpha_i y^i \langle x^i, x\rangle + b.$$

(e) (2 points) What does it mean for the training point x^i if

- $\alpha_i = 0$?
- $0 < \alpha_i < C$?
- $\alpha_i = C$?

Solution:

- If $\alpha_i = 0$, then $r_i = C \neq 0$, hence $\xi_i = 0$ and $y^i(\langle w, x^i\rangle + b) > 1$. The training point is correctly classified.
- If $\alpha_i = C$, then $r_i = 0$ and hence $\xi_i > 0$. In addition, $\alpha_i \neq 0$ yields $y^i(\langle w, x^i\rangle + b) = 1 - \xi_i$. The point is incorrectly classified, with a slack of $\xi_i > 0$.
- If $0 < \alpha_i < C$, then neither $r_i = 0$ nor $\alpha_i = 0$. Hence $\xi_i = 0$ and $y^i(\langle w, x^i\rangle + b) - 1 + \xi_i = 0$, resulting in $y^i(\langle w, x^i\rangle + b) = 1$. The point lies on the separating hyperplane, i.e. is a support vector.

Question 14 ..... *7 points*

(a) (1 point) Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel, with the corresponding feature map $\Phi : \mathcal{X} \to \mathcal{H}$. Write $K(x, x')$ as a function of $\Phi(x), \Phi(x')$

Solution: $K(x, x') = \langle \Phi(x), \Phi(x')\rangle_{\mathcal{H}}$.

(b) (2 points) Write $||\Phi(x) - \Phi(x')||^2$ using only $K(x, x)$, $K(x', x')$ and $K(x, x')$.

Solution:

$$\begin{aligned}||\Phi(x) - \Phi(x')||^2 &= \langle \Phi(x) - \Phi(x'), \Phi(x) - \Phi(x') \rangle_{\mathcal{H}} \\ &= \langle \Phi(x), \Phi(x) \rangle_{\mathcal{H}} + \langle \Phi(x'), \Phi(x') \rangle_{\mathcal{H}} - 2\langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}} \\ &= K(x,x) + K(x',x') - 2K(x,x').\end{aligned}$$

(c) (1 point) Assume we are given n points $\{x^1, \ldots, x^n\} \in \mathcal{X}^n$. Assume we are clustering these points in the feature space $\mathcal{H}$, i.e. we are clustering their images $\{\Phi(x_1), \ldots, \Phi(x_n)\}$. For a given cluster $\mathcal{C}$ of these points, let $\mu \in \mathcal{H}$ be its centroid. Write μ as a function of $\{\Phi(x_1), \ldots, \Phi(x_n)\}$.

Solution:

$$\mu = \frac{1}{|\mathcal{C}|} \sum_{\Phi(x) \in \mathcal{C}} \Phi(x).$$

(d) (2 points) Write $||\Phi(x) - \mu||^2$ for any $x \in \mathcal{X}$ using only K (and not Φ).

Solution:

$$\begin{aligned}||\Phi(x) - \mu||^2 &= \langle \Phi(x) - \frac{1}{|\mathcal{C}|} \sum_{\Phi(x') \in \mathcal{C}} \Phi(x'), \Phi(x) - \frac{1}{|\mathcal{C}|} \sum_{\Phi(x') \in \mathcal{C}} \Phi(x') \rangle_{\mathcal{H}} \\ &= \langle \Phi(x), \Phi(x) \rangle_{\mathcal{H}} + \frac{1}{|\mathcal{C}|^2} \sum_{\Phi(x') \in \mathcal{C}} \sum_{\Phi(z) \in \mathcal{C}} \langle \Phi(x'), \Phi(z) \rangle_{\mathcal{H}} - \frac{2}{|\mathcal{C}|} \sum_{\Phi(x') \in \mathcal{C}} \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}} \\ &= K(x,x) + \frac{1}{|\mathcal{C}|^2} \sum_{\Phi(x') \in \mathcal{C}} \sum_{\Phi(z) \in \mathcal{C}} K(x',z) - \frac{2}{|\mathcal{C}|} \sum_{\Phi(x') \in \mathcal{C}} K(x,x').\end{aligned}$$

(e) (1 point) Does the kernel trick apply to the k-means algorithm?

Solution: Yes. One does not need to be able to compute $\Phi(x)$ nor the centroids explicitly in feature space, and it is therefore possible to conduct all computations using only the kernel K.