Exam MA 2823:
Foundations of Machine Learning
(Solution)

Instructor: Chloé-Agathe Azencott

December 18, 2015

- 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:

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

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

$e^{-1} = 0.368$.

This exam has 15 questions, for a total of 42 points. Don't panic!

Question 1 . . . . . *1 point*

What is regularization, and what purpose does it serve?

Solution: Regularization consists in penalizing the training error by a term that depends on the complexity of the model (equivalent to adjusting the bias/variance trade-off). The goal is to avoid overfitting. It is often used for sparsity / feature selection.

Question 2 ..... *1 point*

A decision tree is said to be fully grown if each leaf only contains training points with the same label. Is a decision tree more likely to overfit if it is fully grown or not?

Solution: A fully-grown decision tree has a more complex decision boundary and is hence more likely to overfit.

Question 3 ..... *1 point*

What kind of decision boundary can be learned with a multi-layer perceptron with one hidden layer?

Solution: Any continuous function (on a compact subset of $\mathbb{R}^p$) can be approximated with a multi-layer perceptron with one hidden layer ("universal approximation"). In particular, unlike (single-layer) perceptrons, they can learn non-linear functions (e.g. XOR).

Question 4 ..... *1 point*

You have data you want to cluster. You are considering several algorithms. How can you decide which one is the most appropriate?

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.

Computational complexity (time/space) is another aspect that it can be necessary to take into account.

Question 5 ..... *2 points*

Each tree of a random forest is built on a bootstrap sample of the training data. If n is the total number of training samples, on how many samples, in average, is each tree built?

Solution: Because bootstrap samples are drawn *with replacement*, a sample can be drawn multiple times. The probability p_i that the i-th training instance belongs to one bootstrap iteration is one minus the probability that is not picked among all n instances, n times. All instances have the same probability of being picked, which is $1/n$. Hence

$$p_i = 1 - \left(1 - \frac{1}{n}\right)^n \approx 1 - e^{-1} = 0.632$$

Hence each tree is trained on about 63.2% of the training data.

Question 6 ..... *2 points*

What is the VC-dimension of a circle in $\mathbb{R}^2$?

Solution: The VC-dimension of a circle in $\mathbb{R}^2$ is the maximum number of points in the plane that can be shattered by a circle. Three points that are not aligned can be shattered by a circle: however we assign them + and - labels, we can always find a circle such that the points labeled + are inside this circle and the points labeled - are outside this circle. Hence the VC-dimension of a circle is at least 3. However, there are no configurations of 4 points that can be shattered by a circle. (Proof not asked.) Hence the VC-dimension of a circle is exactly 3.

Question 7 ..... *2 points*

You are given a test that can determine whether a person is a terrorist or not from the emails they've written and received in the past year. The test is correct 95% of the time. Assume the prevalence of terrorists in this population is 1 in $10,000$[1]. If $10,000$ people have been labeled "positive" by this test, how many of them are not terrorists?

Solution: Denoting belonging to the terrorist class 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 Prime Minister evaluates the prevalence of terrorists to at most $1,500$ (out of 64.410^6 inhabitants), so a prevalence of about 0.23×10^{-4}.

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 are not terrorists.

Question 8 ..... *2 points*

Let us consider n data points $\{x_i\}_{i=1,\dots,n}$ in one dimension ($x \in \mathbb{R}$). Assume they are drawn from a normal distribution of mean μ and variance σ^2:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right].$$

Compute the maximum likelihood estimates of μ and σ.

Solution: Likelihood: $l(\mu, \sigma|X) = p(X|\mu, \sigma) = \prod_{i=1}^n p(x_i|\mu, \sigma)$.

Log-likelihood:

$$\begin{aligned} L(\mu, \sigma|X) &= \sum_{i=1}^n \log p(x_i|\mu, \sigma) \\ &= \sum_{i=1}^n \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)\right) \\ &= \sum_{i=1}^n \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{(x_i-\mu)^2}{2\sigma^2} \end{aligned}$$

To get the MLE, take the derivative and set it to 0.

$$\frac{\partial L}{\partial \mu} = \sum_{i=1}^n \left(\frac{-2\mu}{2\sigma^2} + \frac{2x_i}{2\sigma^2}\right) = \frac{1}{\sigma^2} n\mu - \sum_{i=1}^n x_i$$

Hence $\hat{\mu} = \frac{1}{n}\sum_{i=1}^n x_i$

$$\frac{\partial L}{\partial \sigma} = \sum_{i=1}^n \left(\frac{-\sqrt{2\pi}}{\sqrt{2\pi}\sigma} + \frac{(x_i-\mu)^2}{\sigma^3}\right).$$

Hence

$$-\frac{n}{\hat{\sigma}} + \sum_{i=1}^n \frac{(x_i-\mu)^2}{\hat{\sigma}^3} = 0$$

and finally

$$\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2}.$$

Question 9 ..... *2 points*

The volume of a p-dimensional sphere of radius r is given by

$$\frac{2r^p\pi^{p/2}}{p\Gamma(p/2)}.$$

The Γ function is a continuous generalization of the factorial.

(a) (1 point) Consider two p-dimensional spheres (centered at the origin), one (sphere A) of radius 1, and the other (sphere B), slightly smaller, of radius $1-\epsilon$. What is the proportion of the volume of sphere A that lies between sphere A and sphere B?

Solution: This proportion is given by

$$\rho = \frac{\frac{2\pi^{p/2}}{p\Gamma(p/2)} - \frac{2(1-\epsilon)^p\pi^{p/2}}{p\Gamma(p/2)}}{\frac{2\pi^{p/2}}{p\Gamma(p/2)}} = 1 - (1-\epsilon)^p$$

(b) (1 point) What does this have to do with the curse of dimensionality?

Solution: This proportion goes to 0 when p goes to ∞. In other words, in high dimension, all the points within a sphere are concentrated at the boundary of this sphere, and there are virtually no points closer than that to the origin. Thus, hyperspace is very big, everything is far apart, and it is hard to rely on nearby points having similar labels because no points are nearby.

Question 10 ..... *4 points*

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

$$\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} ||y - X\beta||_2^2 \text{ s. t. } ||\beta||_2^2 \leq t.$$

(a) (2 points) Show there exists a unique $\lambda > 0$ such that this formulation is equivalent to:

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

Solution: We are minimizing a quadratic form $f(\beta)$ under the constraint that $g(\beta) \leq 0$, where $g(\beta) = ||\beta||_2^2 - t$.

Either the unconstrained minimum lies in the feasible region: $g(\beta) \leq 0$, or it does not and the solution is at a point where the iso-contour of f and the feasible region are tangent. In this second case, $g(\beta) = 0$ and the gradients of g and f point towards opposite directions. (Indeed, f increases towards the feasible region, while g increases away from the feasible region.)

Hence this problem can be solved by minimizing (in β) the Lagrangian

$$L(\lambda, \beta) = f(\beta) - \lambda g(\beta)$$

under the constraints that $\lambda > 0$ and $\lambda g(\beta) = 0$.

Minimizing the Lagrangian is equivalent to minimizing (in β):

$$||y - X\beta||_2^2 + \lambda||\beta||_2^2 - \lambda t$$

and λt is a constant. QED.

(b) (2 points) Give the explicit form of the solution. Does it always exist?

Solution:

$$\begin{aligned}\hat{\beta}_{\text{ridge}} &= \arg\min_{\beta} ||y - X\beta||_2^2 + \lambda||\beta||_2^2 \\ &= \arg\min_{\beta} (y - X\beta)^\top (y - X\beta) + \lambda\beta^\top\beta.\end{aligned}$$

Taking the gradient and setting it to 0, we obtain:

$$-2X^\top(y - X\hat{\beta}_{\text{ridge}}) + 2\lambda\hat{\beta}_{\text{ridge}} = 0$$

hence

$$(X^\top X + \lambda I)\hat{\beta}_{\text{ridge}} = X^\top y.$$

Finally:

$$\hat{\beta}_{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top y.$$

Note that $(X^\top X + \lambda I)$ can always be inverted if $\lambda > 0$.

Question 11 ..... *4 points*

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

$\{-1, +1\}$. A soft-margin SVM solves

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

(a) (1 point) Is a soft-margin SVM more likely to overfit if C is large or small?

Solution: If C is large, more importance is given to the error on the training set, and the SVM is more likely to overfit.

(b) (1 point) Give one way of choosing C in practice.

Solution: By cross-validation.

(c) (1 point) What does this mean for a feature j if the solution w_j is close to 0?

Solution: That this feature is uninformative. The class won't depend on this feature. (Note: we're talking about a feature weight w_j, not a (support) vector coefficient α_i.)

(d) (1 point) Give an interpretation of the two terms of Equation (1): $||w||^2$ and $\sum_{i=1}^{n} \xi_i$.

Solution: $||w||$ is the inverse of the margin.
$\sum_{i=1}^{n} \xi_i$ is the sum of slacks ξ_i, which quantify the error for each misclassified training point.

Question 12 ..... *6 points*

Figure 1 represents the architecture of a multi-layer perceptron for K classes. The hidden units are logistic units.

We are given training data $\{(x^i, y^i)\}_{i=1,\ldots,n}$, with $x^i \in \mathbb{R}^p$ and $y^i \in \{0, 1\}^K$.

(a) (1 point) Each training point belongs to one class only. What is the value of $\sum_{k=1}^{K} y_k^i$?

Solution: $y_k^i = 1$ if x^i belongs to class k and 0 otherwise. It is meant to be 1.

(b) (1 point) The output units are softmax units. If o_k denotes the linear combination of the signals from the incoming units, the softmax units transform o_k

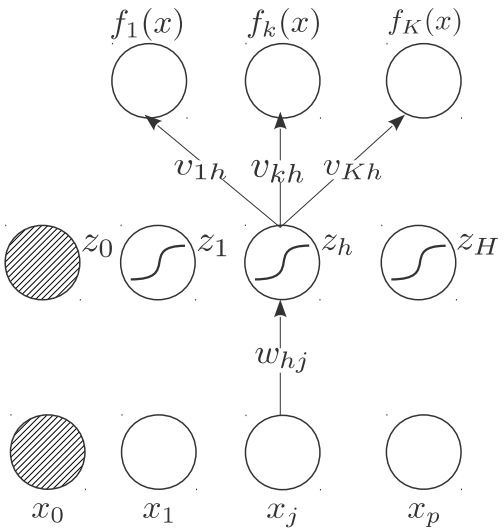


Figure 1: Multi-layer perceptron for K classes. There are $p+1$ input units: $x_1, \ldots, x_p$ and a bias unit $x_0 = 1$, and $H+1$ hidden units: $z_1, \ldots, z_H$ and a bias unit $z_0 = 1$. For clarity, not all connections are represented. Each unit of the input layer is connected to each unit of the hidden layer with a connection weight w_{hj}, and each unit of the hidden layer is connected to each unit of the output layer with a connection weight v_{kh}. The hidden units are logistic units.

as follows:

$$f_k(x) = \frac{\exp(o_k)}{\sum_{l=1}^{K} \exp(o_l)}$$

Why was this function chosen?

Solution: The softmax is a "well-behaved" version of the max. When one class gets a larger output than the others, the softmax will push the ouptut of the corresponding unit towards 1, and push the outputs of the other units towards 0. Unlike the max, it is continuous and derivable, which has mathematical advantages.

(c) (2 points) We will use the cross-entropy error to train this network:

$$\text{Error}(f(x), y) = -\sum_{k=1}^{K} y_k \log f_k(x)$$

Compute the update rule for the weights v_{kh}.

Solution: The output of unit t is given by:

$$f_t(x) = \frac{\exp(v_t^\top z)}{\sum_{l=1}^{K} \exp(v_l^\top z)}$$

The error for training point (x, y) is given by:

$$\text{Error}(f(x), y) = -\sum_{t=1}^{K} y_t \log \frac{\exp(v_t^\top z)}{\sum_{l=1}^{K} \exp(v_l^\top z)}$$

We can show that

$$\frac{\partial f_t}{\partial v_{kh}} = \begin{cases} z_h f_k (1 - f_k) & \text{if } t = k \\ -z_h f_k f_t & \text{if } t \neq k \end{cases}$$

Hence

$$\frac{\partial \log(f_t)}{\partial v_{kh}} = \begin{cases} z_h (1 - f_k) & \text{if } t = k \\ -z_h f_k & \text{if } t \neq k \end{cases}$$

The gradient of the error w.r.t. v_{kh} is given by:

$$\frac{\partial \text{Error}}{\partial v_{kh}} = \left(\sum_{t \neq k} y_t \right) z_h f_k - y_k z_h (1 - f_k)$$

Because if $y_k = 1$, then $\sum_{t \neq k} y_t = 0$ (and conversely), we get:

$$\frac{\partial \text{Error}}{\partial v_{kh}} = \begin{cases} -z_h (1 - f_k) & \text{if } y_k = 1 \\ z_h f_k & \text{if } y_k = 0 \end{cases} = z_h (f_k - y_k)$$

Finally, the update rule (gradient descent) is: $v_{kh} \leftarrow v_{kh} - \eta z_h (f_k - y_k)$

(d) (2 points) Write out the chain rule that enables backpropagation and compute the update rule for the weights w_{hj}.

Solution: Backpropagation chain rule:

$$\frac{\partial \text{Error}}{\partial w_{hj}} = \frac{\partial \text{Error}}{\partial z_h} \frac{\partial z_h}{\partial w_{hj}}$$

$$\frac{\partial \text{Error}}{\partial z_h} = -\sum_{k=1}^{K} y_k \frac{\partial f_k}{\partial z_h} \frac{1}{f_k}$$

And

$$z_h = \frac{1}{1 + e^{-w_h^\top x}}$$

$$\frac{\partial f_k}{\partial z_h} = \frac{v_{kh}\exp(v_k^\top z)\sum_{l=1}^K \exp(v_l^\top z) - \exp(v_k^\top z_h)\sum_{l=1}^K v_{lh}\exp(v_l^\top z)}{\left(\sum_{l=1}^K \exp(v_l^\top z)\right)^2}$$
$$= f_k\left(v_{kh} - \sum_{l=1}^K v_{lh} f_l\right)$$

Hence

$$\frac{\partial \text{Error}}{\partial z_h} = -\sum_{k=1}^K y_k\left(v_{kh} - \sum_{l=1}^K v_{lh} f_l\right) = -\sum_{k=1}^K y_k v_{kh} + \sum_{l=1}^K v_{lh} f_l = \sum_{k=1}^K (f_k - y_k) v_{kh}$$

because $\sum_{k=1}^K y_k = 1$.
Because the derivative of the logistic $\sigma(u)$ can be written as $\sigma(u)(1-\sigma(u))$,

$$\frac{\partial z_h}{\partial w_{hj}} = z_h(1-z_h)x_j$$

Finally:

$$\frac{\partial \text{Error}}{\partial w_{hj}} = \sum_{k=1}^K ((f_k - y_k)v_{kh}) z_h(1-z_h)x_j$$

And hence the update rule (gradient descent) is:

$$w_{hj} \leftarrow w_{hj} + \eta \sum_{k=1}^K ((y_k - f_k)v_{kh}) z_h(1-z_h)x_j.$$

Question 13 ..... *6 points*

Assume we are given n points $\{x^1, \ldots, x^n\} \in \mathcal{X}^n$.

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel, with the corresponding feature map $\Phi : \mathcal{X} \to \mathcal{H}$. We denote by K the $n \times n$ matrix such that $K_{ij} = k(x^i, x^j)$.

We denote by Σ the sample covariance matrix of the images $\{\Phi(x^1), \ldots, \Phi(x^n)\}$ of our data. Let λ and V be an eigenvalue and corresponding eigenvector of Σ.

(a) (1 point) Write $k(x, x')$ as a function of $\Phi(x)$ and $\Phi(x')$.

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

(b) (1 point) Show that V is a linear combination of $\{\Phi(x^1), \ldots, \Phi(x^n)\}$.

Solution:

$$n\lambda V = \sum_{i=1}^{n} \Phi(x^i)\Phi(x^i)^\top V$$

and $\Phi(x^i)^\top V$ is a scalar, hence V spans $\{\Phi(x^1), \ldots, \Phi(x^n)\}$.

(c) (1 point) We can now write V as $V = \sum_{i=1}^{n} \alpha_i \Phi(x^i)$. Let us denote by α the n-dimensional vector $(\alpha_1, \ldots, \alpha_n)$. Show that λ, V are solution to $n\lambda K\alpha = K^2\alpha$.

Solution: We are looking for V, λ such that

$$n\lambda V = \frac{1}{n} \sum_{i=1}^{n} \Phi(x^i)\Phi(x^i)^\top V.$$

As demonstrated above, this implies that V spans $\{\Phi(x^1), \ldots, \Phi(x^n)\}$. Let us write $V = \sum_{i=1}^{n} \alpha_i \Phi(x^i)$ and reinject in the above equation:

$$n\lambda \sum_{i=1}^{n} \alpha_i \Phi(x^i) = \frac{1}{n} \sum_{i=1}^{n} \Phi(x^i)\Phi(x^i)^\top \sum_{j=1}^{n} \alpha_j \Phi(x^j).$$

We can now left-multiply by $\Phi(x^k)^\top$ and sum over k to obtain:

$$n\lambda \sum_{k=1}^{n}\sum_{i=1}^{n} \Phi(x^k)^\top\Phi(x^i)\alpha_i = \frac{1}{n} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \Phi(x^k)^\top\Phi(x^i)\Phi(x^i)^\top\Phi(x^j)\alpha_j.$$

Hence:

$$n\lambda \sum_{k=1}^{n}\sum_{i=1}^{n} k(x^k, x^i)\alpha_i = \frac{1}{n} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} k(x^k, x^i)k(x^i, x^j)\alpha_j.$$

QED.

(d) (1 point) Show how PCA in the feature space can be conducted directly in the original space $\mathcal{X}$, using only k and never computing $\Phi(x)$ for any x.

Solution: All solutions of $n\lambda\alpha = K\alpha$ satisfy the previous equation.
Finding the eigenvectors/eigenvalues of K gives us the eigendecomposition of Σ, hence the PCA in feature space.
The projection of $\Phi(x)$ onto a PC, i.e. an eigenvector V of Σ, can be computed as:

$$\Phi(x)^\top V = \sum_{i=1}^{n} \alpha_i \Phi(x)^\top \Phi(x^i) = \sum_{i=1}^{n} \alpha_i k(x, x^i)$$

and does not require the explicit computation of Φ either.

(e) (1 point) What is the advantage of never having to compute $\Phi(x)$ explicitly?

Solution: This kernel trick means that we don't need to know the form of Φ, and we can compute PCA in spaces for which we don't know an exact mapping (e.g infinite-dimensional spaces with RBF kernel). In other cases (e.g. string kernel, Kendall-tau kernel) this can also afford computational advantages.

(f) (1 point) Can you get an explicit expression of the principal components without using Φ?

Solution: No. $V = \sum_{i=1}^{n} \alpha_i \Phi(x^i)$. This is one drawback of kernel PCA.

Question 14 ..... *4 points*

Show how to apply the kernel trick to the k-means algorithm.

Solution: The key point here is the ability to compute the distance from the image $\Phi(x)$ of a data point x to the centroid of cluster C *in feature space*, which we'll call ν.

By definition,

$$\nu = \frac{1}{|C|} \sum_{z \in C} \Phi(z)$$

Hence:

$$\begin{aligned} ||\Phi(x) - \nu||^2 &= ||\Phi(x) - \frac{1}{|C|} \sum_{z \in C} \Phi(z)||^2 \\ &= K(x,x) - \frac{2}{|C|} \sum_{z \in C} K(x,z) + \frac{1}{|C|^2} \sum_{z,z' \in C} K(z,z') \end{aligned} \tag{2}$$

Hence the re-assignment of data points to clusters can be computed in the initial space, without ever needing to compute Φ explicitly.

The kernel k-means algorithm then proceeds as follows:

1. Randomly assign each data point to one of the k clusters.
2. Until convergence (assignment doesn't change), or a fixed number of iterations is reached, repeat:

(a) Compute the distance of each data point x to the centroid of each of the k clusters, using Eq. 2 (only involving K).

(b) Re-assign each point to the cluster whose centroid it is closest to.

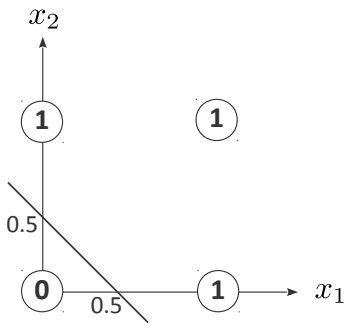
Question 15 ..... *4 points*

Table 1 gives the decision table for OR. Give a perceptron that predicts this function.

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

Table 1: Decision table for OR. x_1 and x_2 are the inputs, and y the output.

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



$$w_0 = -0.5$$
$$w_1 = 1$$
$$w_2 = 1$$

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