UNIVERSITY OF AMSTERDAM

Faculty of Science

Exam

Machine Learning 1

Resit Exam
Date: Februari 13, 2019
Time: 18:00-21:00

Number of pages: 10 (including front page)
Number of questions: 5
Maximum number of points to earn: 46
At each question the number of points you can earn is indicated.

BEFORE YOU START

- As soon as you receive your exam you may start.
- Check if your version of the exam is complete.
- Write down **your name, student ID number**, and if applicable the **version number** on **each sheet** that you hand in. Also **number the pages**.
- Your **mobile phone** has to be switched off and in the coat or bag. Your **coat and bag** must be under your table.
- **Tools allowed**: 1 handwritten double-sided A4-size cheat sheet, pen.
- Multiple choice answers must be indicated on the exam booklet.

PRACTICAL MATTERS

- The first 30 minutes and the last 15 minutes you are not allowed to leave the room, not even to visit the toilet.
- You are obliged to identify yourself at the request of the examiner (or his representative) with a proof of your enrollment or a valid ID.
- During the examination it is not permitted to visit the toilet, unless the proctor gives permission to do so.
- 15 minutes before the end, you will be warned that the time to hand in is approaching.
- Please fill out the evaluation form at the end of the exam.

Good luck!

1 Multiple Choice Questions

/10

Indicate your answers for the multiple choice questions on this exam sheet. For the evaluation of each question note the following: several answers can be correct and at least one is correct. You are granted one point if every correct answer is 'marked' **and** every incorrect answer is 'not marked'. For each mistake 0.5 points are deducted, with the minimum possible number of points per question equal to 0. A box counts as 'marked' if a clearly visible symbol is written in there or if the box is blackened out. In the case you want to change an already marked box write 'not marked' next to the box.

1. Which of the following statements about the bias-variance decomposition are true: **/1**

   - [ ] Complex models are less likely to suffer from high variance than simple models.
   - [X] High variance can be reduced by fitting your model to more data.
   - [ ] Simple models are more likely to have low bias than complex models.
   - [ ] A model that suffers from high variance is not sensitive to overfitting.

2. Which of the following expressions are correct, given no independence assumption and for (non-trivial) discrete random variables? **/1**

   - [ ] $\sum_b P(A|B=b) = 1$.
   - [X] For two values $a_1 \neq a_2$, $P(A = a_1 \text{ or } A = a_2|B) = P(A = a_1|B) + P(A = a_2|B)$.
   - [X] $\sum_a \sum_b P(A = a|B = b)P(B = b) = 1$.
   - [ ] $p(x) = \int p(x, y)p(y)dy$.

3. Consider a neural network with two layers, and 10 hidden units in the hidden layer. Which of the following statements are correct? **/1**

   - [X] For regression with targets $t \in \mathbb{R}$, a suitable activation function for the output unit is $f(x) = x$.
   - [ ] For classification with $K > 2$ mutually exclusive classes we need K output units with activation functions $f(x) = \frac{1}{1+e^{-x}}$.
   - [X] For classification with binary targets we can use 1 output unit with activation function $f(x) = \frac{1}{1+e^{-x}}$.
   - [ ] For regression with targets $t \geq 0$, a suitable activation function for the output unit is $f(x) = \tanh(x)$.

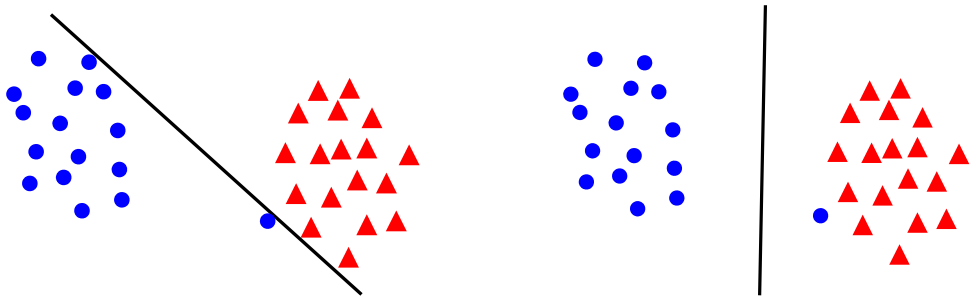
4. Consider regularized linear regression with the error function $E(\mathbf{w}, \lambda) = \frac{1}{2N} \sum_{n=1}^{N} \left(\boldsymbol{\phi}(\mathbf{x}_n)^T \mathbf{w} - t_n\right)^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$ (Note the $1/N$ factor). You want to find the optimal regularization penalty $\lambda \in \{1,\ 0.1,\ 0.01\}$. You will split your data into a training set, a validation set and a test set. You obtain the following validation and training errors $E_{\text{val}}(\mathbf{w}, \lambda)$, and $E_{\text{train}}(\mathbf{w}, \lambda)$:

	$E_{\text{train}}(\mathbf{w}, \lambda)$	$E_{\text{val}}(\mathbf{w}, \lambda)$
$\lambda_1 = 1$	0.25	0.32
$\lambda_2 = 0.1$	0.21	0.39
$\lambda_3 = 0.01$	0.16	0.51

Which of the following statements is the most appropriate? **/1**

- [ ] λ_1: underfitting, λ_2: underfitting, λ_3: best fit.
- [ ] λ_1: underfitting, λ_2: best fit, λ_3: overfitting.
- [ ] λ_1: overfitting, λ_2: best fit, λ_3: underfitting.
- [X] λ_1: best fit, λ_2: overfitting, λ_3: overfitting.

5. Consider the following figures depicting a dataset with datapoints from two classes corresponding to the blue circles and the red triangles:

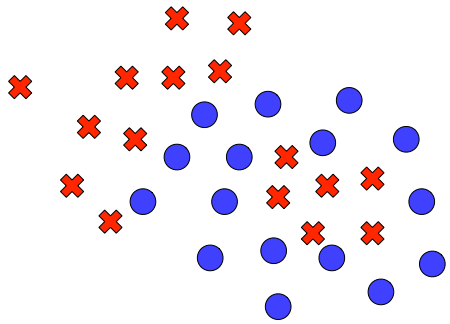


(a) (b)

The black lines correspond to decision boundaries constructed using (different) Maximum Margin classifiers. Which of the following statements about the above figures are correct: **/1**

- [X] It is likely that the decision boundary in (a) is determined by a Maximum Margin classifier with a hard margin.
- [X] The decision boundary in (b) is more likely to lead to a good generalization performance than the decision boundary shown in (a).
- [ ] It is likely that the decision boundary in (a) is determined by a Maximum Margin classifier with a soft margin and a very low penalty for misclassifications.
- [X] It is likely that the decision boundary in (b) is determined by a Maximum Margin classifier with a soft margin and a low penalty for misclassifications.

6. Which of the following classifiers can learn a perfect decision boundary for the dataset shown in the figure below? The blue circles belong to one class, and the red crosses belong to the other class.



/1

- [X] Neural networks.
- [ ] A perceptron algorithm with linear features.
- [ ] Linear Discriminant Analysis with linear features.
- [X] Support Vector Machines with a kernel of the form $k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp\left(-\frac{1}{2\theta_1^2}||\mathbf{x}_n - \mathbf{x}_m||^2\right)$

7. Consider a Gaussian process (GP) with a mean function $m(\mathbf{x}) = 0$ and the following kernel:

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp\left(-\frac{1}{2\theta_1^2}||\mathbf{x}_n - \mathbf{x}_m||^2\right) + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m$$

Indicate which of the following statements are correct: **/1**

- [X] The parameter θ_0 determines the amplitude of the oscillations of a function drawn from this GP.
- [ ] If $\theta_0 = 0$, $\theta_1 = 1$, $\theta_2 = 5$ and $\theta_3 = 1$, then functions drawn from a Gaussian process with this kernel will all be functions of the form $f(\mathbf{x}) = c$, for different values of the constant c. So they are all constant functions at different heights.
- [X] The parameter θ_1 determines the typical length scale over which a function drawn from this GP shows oscillations.
- [X] If $\theta_3 > 0$ and $\theta_0 = 0$, $\theta_1 = 1$, $\theta_2 = 0$, then this kernel corresponds to a feature map $\phi(\mathbf{x})$ which is linear in $\mathbf{x}$.

8. Which of the following statements about ensemble methods are correct? **/1**

- [X] Bootstrap datasets are created by sampling with replacement from one single original dataset.
- [X] The boosting algorithm can be derived with an exponential error function.
- [ ] Several datasets are constructed using feature bagging. For each dataset a new model is trained, and all models are of the same type. If the features are uncorrelated, then the predictions of the different trained models will also be uncorrelated.
- [X] A decision tree divides the input space into rectangular decision regions.

9. Which of the following statements about K-means and Gaussian mixture models are correct? **/1**

- [ ] In K-means the number of clusters is also learned.
- [X] An update step in the K-means algorithm can decrease the loss or leaves it unchanged.
- [X] In Gaussian mixture models, in the E-step the responsibilities $r_{nk} = p(z_n = k|\boldsymbol{x}_n)$ are computed.
- [ ] Gaussian mixture models converge to a global optimum.

10. Consider the case where you have an image consisting of 1×10^6 pixels, with each pixel represented by a vector of size 3 with the RGB values. You want to perform image compression by applying a K-means clustering algorithm to the pixels in the image. **/1**

- [ ] The number of clusters K represents the number of pixels in the compressed image
- [X] The number of clusters K represents the number of colors that remain in the compressed image.
- [X] Each cluster centroid represents an RGB value of one of the colors that remains in the compressed image.
- [ ] One datapoint represents one image.

Grading instructions

The solutions given below, with the corresponding distribution of points, serve as a guideline. If some intermediate steps are left implicit by the student, while still clearly following a derivation, points will not be deducted. The total number of possible points is 46, meaning that the final grade is computed as $10 \times \frac{\#\text{points}}{46}$.

General remarks

The exercises below have subquestions that are not all dependent on each other. If you get stuck at one subquestion, don't stop but try to solve the next ones!

2 Probability theory and Bayes rule

/6

Suppose that you are worried that you have a disease, and you want to get tested. The disease is very rare, it only occurs in one of every 10000 people. The test you are taking is correct 99 percent of the time: if you have the disease the test will lead to a positive result with 99 percent of the time, and if you do not have the disease the test will lead to a negative result 99 percent of the time.

a) List all the random variables and for each random variable list the values it can take on. **/2**

b) If you test positive for the disease, what is the probability that you actually have the disease? You can leave your numerical answer in the form of a fraction if you do not have a calculator with you. **/4**

Solutions

a) $D \in \{d, h\}$ disease status (d = disease, h =healthy) (1pt), $T \in \{+, -\}$ test outcome (positive for disease, negative for healthy) (1pt).

b) We are interested in the probability $p(d|+)$, which according to Bayes rule is equal to

$$p(d|+) = \frac{p(+|d)p(d)}{p(+)} \quad \text{(1pt)}$$

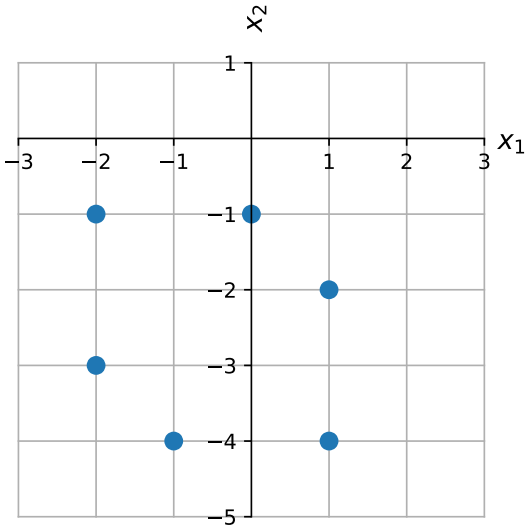
Here $p(+|d) = 0.99$, is the probability of testing positive given that you have the disease. The probability of someone having the disease is equal to $p(d) = 1/10000 = 0.0001$ ((1p) together for $p(+|d)$ and $p(d)$). The marginal probability of being tested positive is given by $p(+) = p(+|d)p(d) + p(+|h)p(h)$ (1p). The probability of testing positive even though you are healthy is given by $p(+|h) = 1 - p(-|h) = 1 - 0.99 = 0.01$. The probability of not having this disease/being healthy is equal to $p(h) = 1 - p(d) = 9999/10000 = 0.9999$ ((1p) together for $p(+|h)$ and $p(h)$). Inserting these numbers gives

$$p(c|+) = \frac{p(+|d)p(d)}{p(+|d)p(d) + p(+|h)p(h)} = \frac{0.99 \times 0.0001}{0.99 \times 0.0001 + 0.01 \times 0.9999} \approx 0.098 \ .$$

3 PCA

/7

Consider the figure below which depicts a dataset of 6 points in 2D.



Answer the following questions about this dataset. You can use the figure above to draw on if that helps you find the solutions.

a) What is the normalized first principal component $\mathbf{u}_1$? Explain how you computed it. You do not need to solve an eigenvalue problem to answer this question. **/1**

b) What is the normalized second principal component $\mathbf{u}_2$? Explain how you computed it. Again, you do not need to solve an eigenvalue problem to answer this question. **/1**

c) The eigenvalues of the covariance matrix for this dataset are 11/6 and 4/3. Which eigenvalue corresponds to principal component $\mathbf{u}_1$, and which corresponds to $\mathbf{u}_2$? **/1**

d) What is the reconstruction error $\frac{1}{N}\sum_{n=1}^{N}\|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2$ if we project each datapoint $\mathbf{x}_n$ onto a 1D line by using the first principal component such that the projected datapoints become

$$\tilde{\mathbf{x}}_n = \bar{\mathbf{x}} + \left((\mathbf{x}_n - \bar{\mathbf{x}})^T \mathbf{u}_1\right)\mathbf{u}_1.$$

Here, $\bar{\mathbf{x}}$ is the average of all the original datapoints: $\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$. **/1**

e) Write down an equation that transforms each datapoint such that the resulting dataset is centered, uncorrelated but not whitened. Indicate explicitly how this transformation depends on the principal components $\mathbf{u}_1$ and $\mathbf{u}_2$ and the eigenvalues of the covariance matrix. **/1**

f) Write an equation that results in a centered, uncorrelated *and whitened* dataset. Again indicate the dependence on the principal components $\mathbf{u}_1$ and $\mathbf{u}_2$ and the eigenvalues of the covariance matrix. Show that for *any dataset* for which you have computed the principal components and the eigenvalues of the covariance matrix, this transformation achieves whitening. **/2**

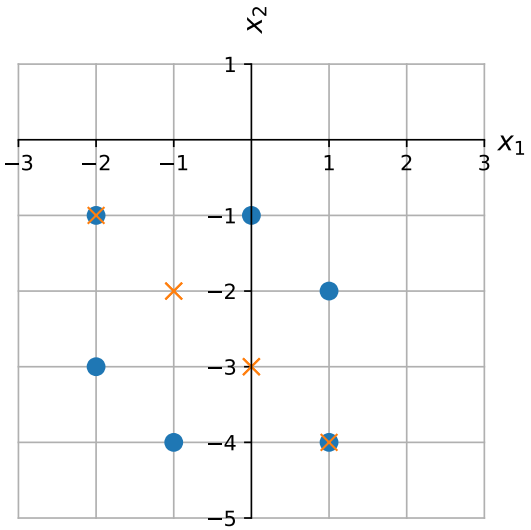
Solutions

a) The first principal component points in the direction of largest variance, which is in the direction $(1,-1)^T$. Normalizing this gives $\mathbf{u}_1 = (1/\sqrt{2}, -1/\sqrt{2})^T$ (1p). Note that the negative of this vector is also valid.

b) The second principal component points in the direction of second largest variance, which is in the direction $(1,1)^T$. Normalizing this gives $\mathbf{u}_2 = (1/\sqrt{2}, 1/\sqrt{2})^T$ (1p). Note that the negative of this vector is also valid.

c) The largest eigenvalue should correspond to $\mathbf{u}_1$, which is $11/6$, and $4/3$ corresponds to $\mathbf{u}_2$ (1p).

d) See figure below: the points will be projected to the four orange crosses, where each pair of two blue datapoints separated along the vector $\mathbf{u}_2$ map to the same cross in between them. For the four blue datapoints making up the square, the squared distance between the crosses and the blue dots along the direction of $\mathbf{u}_2$ is $\|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 = 2$, so the average reconstruction error is $4 \times 2/6 = 4/3$ (1p). Alternatively, the reconstruction error is equal to the variance in the direction of $\mathbf{u}_2$, which is equal to the corresponding eigenvalue: $4/3$.



e) We collect the principal component vectors as columns in the matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2]$. For a centered and uncorrelated but not whitened dataset we use $\mathbf{y}_n = \mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})$ (1p). This transformation is independent of the eigenvalues.

f) We define the matrix $\mathbf{L}$ such that $L_{11} = \lambda_1 = 11/6$ and $L_{22} = \lambda_2 = 4/3$ and $L_{12} = L_{21} = 0$. For a centered, uncorrelated and whitened dataset we use $\mathbf{y}_n = \mathbf{L}^{-1/2}\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})$ (1p). To show that this transformation in achieves whitening for any dataset we need to show that the covariance matrix of the resulting dataset is equal to the identity matrix:

$$\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^T = \frac{1}{N}\sum_{n=1}^{N}\mathbf{L}^{-1/2}\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T\mathbf{U}\mathbf{L}^{-1/2}$$
$$= \mathbf{L}^{-1/2}\mathbf{U}^T\mathbf{S}\mathbf{U}\mathbf{L}^{-1/2} = \mathbf{L}^{-1/2}\mathbf{U}^T\mathbf{U}\mathbf{L}\mathbf{U}^T\mathbf{U}\mathbf{L}^{-1/2} = \mathbf{L}^{-1/2}\mathbf{L}\mathbf{L}^{-1/2} = \mathbf{I}. \quad \text{(1p)}$$

Here we have used that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, and that the eigendecomposition of the original data covariance matrix is $\mathbf{S} = \mathbf{U}\mathbf{L}\mathbf{U}^T$.

4 Mixture of Poisson distributions

/9

Our task is to cluster series of measurements containing the number of decay events per second from a radioactive particle source. We are given an unlabelled dataset $X = \{\boldsymbol{x}_n\}_{n=1}^N$, where each $\boldsymbol{x}_n$ represents a series of measurements of decay counts per second. Each $\boldsymbol{x}_n$ is a vector of size D containing integer numbers that are larger or equal to zero: $x_{ni} \in 0, 1, 2, ...$ for $i = 1, .., D$. D is the total number of measurements of decay counts per second, within one series. We assume that there are K different sources of decaying particles, and that the measurements are generated as follows:

i) The sources are represented by a discrete latent variable $z \in \{1, ..., K\}$ with probability distribution $p(z) = \prod_{k=1}^K \pi_k^{I[z=k]}$ and $\sum_{k=1}^K \pi_k = 1$. Here, $I[z = k]$ is the indicator function. The parameters $\pi_k \geq 0$ represent the prior probabilities for each source k being present, and are unknown, so they need to be learned.

ii) For a vector $\boldsymbol{x}$ with integer elements larger or equal to zero that corresponds to a measurement of source $z = k$, each x_i $(i = 1, ..., D)$ is sampled independently from a Poisson distribution with parameter λ_k:

$$p(x_i|z = k) = e^{-\lambda_k} \frac{(\lambda_k)^{x_i}}{x_i!} \quad .$$

Here $x_i!$ indicates the factorial function such that $x_i! = x_i(x_i - 1)(x_i - 2)...2 \cdot 1$ and $0! = 1$. The parameters $\lambda_k > 0$ are so-called rate parameters, representing the average number of decay events per second, and need to be learned.

Answer the following questions.

a) How many parameters does our model contain? Indicate how this number depends on K, D, N. **/1**

b) Compute the probability of a measurement $\boldsymbol{x}_n$ conditioned on source k: $p(\boldsymbol{x}_n|z_n = k)$. Compute the marginal probability of $\boldsymbol{x}_n$ under this model: $p(\boldsymbol{x}_n)$. Your answers should be functions of the model parameters and the datapoints. **/2**

c) Compute the responsibility (or posterior) $r_{nk} = p(z_n = k|\boldsymbol{x}_n)$ of a measurement with feature vector $\boldsymbol{x}_n$ containing decay counts per second originating from source k. **/1**

d) To derive the EM algorithm for a mixture of Poisson distributions, we will maximize the so-called *expected complete log-likelihood*:

$$\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{X})}\left[\ln p(\boldsymbol{X}, \boldsymbol{Z}|\{\pi_k\}_{k=1}^K, \{\lambda_k\}_{k=1}^K)\right] = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \left(\ln \pi_k - D\lambda_k + \sum_{i=1}^D [x_{ni} \ln \lambda_k - \ln x_{ni}!] \right) \quad . \tag{1}$$

Maximize Eq. (1) with respect to all λ_k for fixed responsibilities r_{nk}. Use this to write down the update rule for λ_k as a function of the responsibilities r_{nk}. Note that we are not looking for a gradient descent update, but a closed-form update that maximizes Eq. (1) with respect to all λ_k for fixed responsibilities r_{nk}. **/2**

e) Similar to d), obtain an update rule for each parameter π_k. *Hint*: do not forget to ensure $\sum_{k=1}^{K} \pi_k = 1$. **/2**

f) Explain in words how you would use the update rules in d) and e) in the EM algorithm. **/1**

Solutions

a) There are two sets of parameters, $\boldsymbol{\pi}$ and $\boldsymbol{\lambda}$. Both $\boldsymbol{\pi}$ and $\boldsymbol{\lambda}$ are vectors of dimension K (one per source). Therefore the total number is $2K$ $(1p)$. There is no dependence on N and D. To be more precise, since $\sum_{k=1}^{K} \pi_k = 1$, we may notice that there is no need to store all K μ_k parameters, but it suffices to have $K-1$; the total is then $2K-1$.

b) Because of independence we write:

$$p(\boldsymbol{x}_n|z=k) = \prod_{i=1}^{D} e^{-\lambda_k} \frac{(\lambda_k)^{x_{ni}}}{x_{ni}!} \quad .(1\text{pt})$$

The marginal:

$$p(\boldsymbol{x}_n) = \sum_{k=1}^{K} \pi_k \prod_{i=1}^{D} e^{-\lambda_k} \frac{(\lambda_k)^{x_{ni}}}{x_{ni}!} \quad .(1\text{pt})$$

c) By Bayes theorem:

$$\begin{aligned} r_{nk} = p(z=k|\boldsymbol{x}_n) &= \frac{p(\boldsymbol{x}_n|z=k)p(z=k)}{p(\boldsymbol{x}_n)} \\ &= \frac{\pi_k \prod_{i=1}^{D} e^{-\lambda_k} \frac{(\lambda_k)^{x_{ni}}}{x_{ni}!}}{\sum_{j=1}^{K} \pi_j \prod_{i=1}^{D} e^{-\lambda_k} \frac{(\lambda_k)^{x_{ni}}}{x_{ni}!}} \quad . \end{aligned}$$

(1pt) for Bayes rule.

d) It is not necessary to show *all* of the steps below; these solutions show them all for the sake of clarity. We can differentiate with respect to λ_k.

$$\begin{aligned} &\frac{\partial}{\partial \lambda_k} \sum_{n=1}^{N} \sum_{k'=1}^{K} r_{nk'} \left(\ln \pi_{k'} - D\lambda_{k'} + \sum_{i=1}^{D} [x_{ni} \ln \lambda_{k'} - \ln x_{ni}!] \right) \\ &= \sum_{n=1}^{N} r_{nk} \frac{\partial}{\partial \lambda_k} \left(-D\lambda_k + \sum_{i=1}^{D} x_{ni} \ln \lambda_k \right) \\ &= \sum_{n=1}^{N} r_{nk} \left(-D + \frac{1}{\lambda_k} \sum_{i=1}^{D} x_{ni} \right) = 0 \quad . \ (1\text{pt}) \end{aligned}$$

Reshuffling, we obtain

$$\lambda_k = \frac{\sum_{i=1}^{D} \sum_{n=1}^{N} r_{nk} x_{ni}}{D \sum_{n=1}^{N} r_{nk}} \quad , (1\text{pt})$$

Note that this makes sense since λ_k is the rate parameter of the Poisson distribution, representing the average number of decay events per second. As such λ_k should have the same units as x_{ni}. The above result shows that λ_k is equal to the weighted mean of the number of decay events per second in each datapoint, with weighting coefficients given by the responsibilities that source k takes for the datapoints.

e) We need to optimize the Lagrangian

$$L = \sum_{n=1}^{N}\sum_{k'=1}^{K} r_{nk'}\left(\ln \pi_{k'} - D\lambda_{k'} + \sum_{i=1}^{D}\left[x_{ni}\ln\lambda_{k'} - \ln x_{ni}!\right]\right) + a\left(\sum_{k=1}^{K}\pi_{k'} - 1\right),$$

with respect to π_k and the Lagrange multiplier a.

$$\begin{aligned}
&\frac{\partial}{\partial \pi_k}\sum_{n=1}^{N}\sum_{k'=1}^{K} r_{nk'}\left(\ln \pi_{k'} - D\lambda_{k'} + \sum_{i=1}^{D}\left[x_{ni}\ln\lambda_{k'} - \ln x_{ni}!\right]\right) + a\left(\sum_{k=1}^{K}\pi_{k'} - 1\right)\\
&= \sum_{n=1}^{N} r_{nk}\frac{\partial}{\partial \pi_k}\ln \pi_k + \frac{\partial}{\partial \pi_k} a\left(\sum_{k=1}^{K}\pi_k - 1\right)\\
&= \sum_{n=1}^{N}\frac{r_{nk}}{\pi_k} + a \quad \rightarrow \quad \pi_k = -\frac{1}{a}\sum_{n=1}^{N} r_{nk} \text{ . (1pt)}
\end{aligned}$$

Use that

$$\sum_{k=1}^{K}\pi_k = -\frac{1}{a}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} \quad \rightarrow \quad 1 = -\frac{1}{a}\sum_{n=1}^{N} 1 \quad \rightarrow \quad a = -N.$$

where we used $\sum_{k=1}^{K} r_{nk} = \sum_{k=1}^{K} p(z=k|x_n) = 1$. So the final answer for the update is

$$\pi_k = \frac{1}{N}\sum_{n=1}^{N} r_{nk} \text{ . (1pt)}$$

f) Updating all of the λ_k is part of the Maximization step in the Expectation-Maximization algorithm, where we update all the parameters and we keep the posterior fixed; the Maximization step would also include an update for $\boldsymbol{\pi}$. In the Expectation-step, we re-compute the posterior probabilities r_{nk}, while keeping all the parameters fixed .(1pt).

5 Maximum margin classifier: diamond shaped decision boundaries

/14

We receive a dataset of N two-dimensional datapoints and their corresponding class values $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$ with $\mathbf{x}_n \in \mathbb{R}^2$, and $t_n \in \{-1, +1\}$. See Figure 2 below for an illustration of an example dataset, where the blue crosses correspond to datapoints for which $t_n = +1$, and the green spheres are datapoints for which $t_n = -1$.
We assume our data is centered around the origin $(0, 0)$, and we expect our data to be *perfectly* separable by a diamond-shaped decision boundary, with equal height and width $h \geq 0$. Our goal is to design a maximum margin classifier with diamond-shaped decision and margin boundaries. The margin size is indicated in Figure 2 with the black curly bracket.

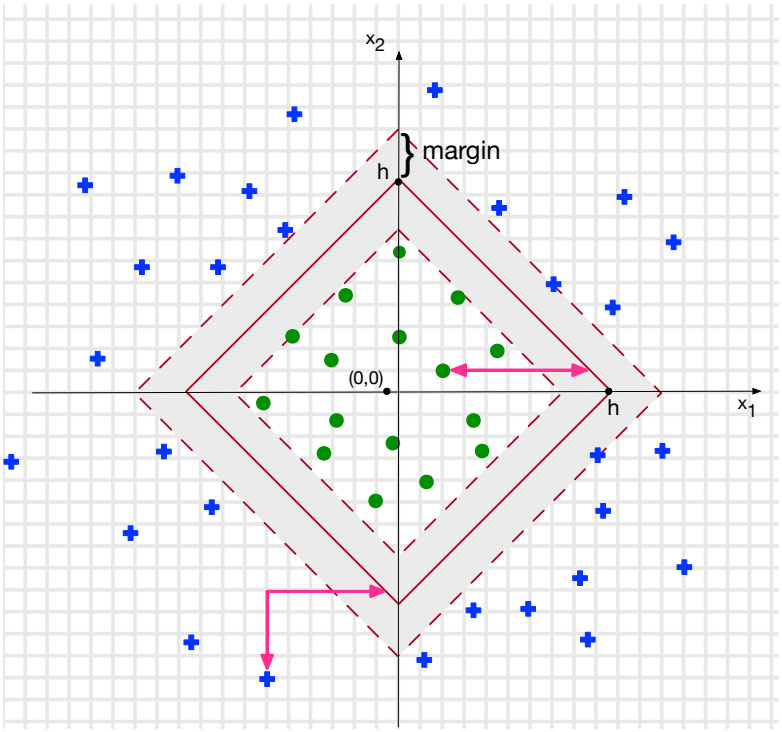


Figure 2

If all datapoints are correctly classified, then the "distance" to the decision boundary for each correctly classified datapoint (indicated by the pink double-headed arrows), is given by

$$t_n(\|\mathbf{x}_n\|_1 - h) = \frac{t_n(a\|\mathbf{x}_n\|_1 - \hat{h})}{a},$$

where $\hat{h} = ah \geq 0$, and $a > 0$. Here, $\|\mathbf{x}_n\|_1 = |x_{n1}| + |x_{n2}|$ is the norm that returns the sum of the absolute values of the elements of $\mathbf{x}_n$. Note that this "distance" to the decision boundary is invariant to a rescaling $a \to \kappa a$. We can use this to set

$$t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) = 1$$

for the datapoints $\mathbf{x}_n$ that are correctly classified and lie on the margin boundary. For a perfect classifier all other points should be further away from the decision boundary, such that

$$t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) \geq 1 \quad \text{for} \quad n = 1, ..., N.$$

This leads to the following primal constrained optimization problem:

$$\min_{a,\hat{h}} a^2 \tag{2}$$

with the following constraints:

$$\textbf{(I)} \quad t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) \geq 1 \quad \text{for all} \quad n = 1, ..., N \tag{3}$$

$$\textbf{(II)} \quad \hat{h} \geq 0 \tag{4}$$

a) What is the size of the margin as indicated in the figure above? **/1**

b) Write down the primal Lagrangian function. Use Lagrange multipliers $\{\lambda_n\}_{n=1}^N$ for constraints (**I**) and δ for constraint (**II**). Which variables are the primal variables? Which variables are the dual variables? **/2**

c) Write down all of the KKT conditions. Do not consider the conditions obtained by optimizing the Lagrangian with respect to the primal variables as KKT conditions. How many KKT conditions do we have in total? **/2**

d) Optimize the Lagrangian with respect to the primal variables. This should give you a set of additional conditions on the Lagrange multipliers. **/2**

e) Derive the dual Lagrangian of the problem. Do not forget to list the conditions on the variables that you need to optimize with respect to in the dual Lagrangian! **/3**

f) We have obtained a dual Lagrangian that is of a form where we can identify which kernel function corresponds to the setup of our problem. What is the explicit form of $\kappa(\mathbf{x}_n, \mathbf{x}_m)$ in your solution to the dual Lagrangian in (e)? If you have not managed to get a sensible answer at (e), you can also argue from the problem setup what the kernel should be. **/1**

g) We now assume that the dataset is *not* perfectly separable with a diamond-shaped decision boundary. Explain how to adjust the optimization problem in Eq. (2) and the constraints in Eq. (3) and Eq. (4), such that we obtain a *soft* margin classifier with diamond-shaped decision boundaries. Using this adjusted optimization problem, write down the corresponding primal Lagrangian. **/3**

Solutions

a) The correctly classified points on the margin boundary satisfy $t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) = 1$, such that for $t_n = +1$, we have

$$a\|\mathbf{x}_n\|_1 - \hat{h} = 1 \rightarrow \|\mathbf{x}_n\|_1 = \frac{1}{a} + \frac{\hat{h}}{a} = h + \frac{1}{a}.$$

So the "distance" to the boundary as measured by the $\|\mathbf{x}_n\|_1$ norm, as indicated in the figure above for a point that lies on the vertex of the diamond, is given by $1/a$ (1p). Note that for points on the vertcies $\|\mathbf{x}_n\|_2 = \|\mathbf{x}_n\|_1$.

b)

$$\mathcal{L}(a, \hat{h}, \{\lambda_n\}_{n=1}^N, \delta) = a^2 - \sum_{n=1}^{N} \lambda_n[t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) - 1] - \delta\hat{h}$$

(1p) for the right collection of terms and signs.
(1p) for naming the primal variables $a, \hat{h}$, and the dual variables $\{\lambda_n\}, \delta$.

c)

$$\lambda_n \geq 0$$
$$t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) - 1 \geq 0$$
$$\lambda_n\{t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) - 1\} = 0$$

for all $n = 1, ..., N$. And

$$\delta \geq 0$$
$$\hat{h} \geq 0$$
$$\delta\hat{h} = 0$$

(1p) for all constraints with correct signs, and (1p) for the number $3N + 3$.

d)

$$\frac{\partial \mathcal{L}}{\partial a} = 2a - \sum_{n=1}^{N} \lambda_n t_n \|\mathbf{x}_n\|_1 = 0 \quad \rightarrow \quad a = \frac{1}{2}\sum_{n=1}^{N} \lambda_n t_n \|\mathbf{x}_n\|_1 \quad \text{(1p)}$$
$$\frac{\partial \mathcal{L}}{\partial \hat{h}} = \sum_{n=1}^{N} \lambda_n t_n - \delta = 0 \quad \rightarrow \quad \delta = \sum_{n=1}^{N} \lambda_n t_n \quad \text{(1p)}$$

One point for each derivative that is computed correctly *and* set equal to zero.

e) Collecting all terms that correspond to each primal variable and using the conditions derived in (d) leads to

$$\begin{aligned}\tilde{\mathcal{L}}(\{\lambda_n\}) =& a\left(a - \sum_{n=1}^{N} \lambda_n t_n \|\mathbf{x}_n\|_1\right) + \hat{h}\left(\sum_{n=1}^{N} \lambda_n t_n - \delta\right) + \sum_{n=1}^{N} \lambda_n \\ =& a\,(a - 2a) + \sum_{n=1}^{N} \lambda_n \\ =& -\frac{1}{4}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_n \lambda_m t_n t_m \|\mathbf{x}_n\|_1 \|\mathbf{x}_m\|_1 + \sum_{n=1}^{N} \lambda_n \quad \text{(1p)}\end{aligned}$$

This dual Lagrangian depends on all λ_n for $n = 1, ..., N$. The resulting conditions for λ_n can be constructed by combining the conditions in c) and d). From the KKT conditions in d) we know that $\lambda_n \geq 0$. (1p). We furthermore found in c) that $\delta = \sum_{n=1}^{N} \lambda_n t_n$, and from d) we know $\delta \geq 0$, so the second condition is $\sum_{n=1}^{N} \lambda_n t_n \geq 0$ (1p).

f) In our solution we have $\kappa(\mathbf{x}_n, \mathbf{x}_m) = \|\mathbf{x}_n\|_1 \|\mathbf{x}_m\|_1$ (1p). Note that this is a valid kernel since $\kappa(\mathbf{x}_n, \mathbf{x}_m) = f(\mathbf{x}_n) f(\mathbf{x}_m)$ is valid for any function f that maps to a real number. In this case f is just a special feature vector of size 1. You can also see from the problem setup that we are using $\phi(\mathbf{x}) = \|\mathbf{x}\|_1$ as a feature vector (actually a feature scalar), so that $\kappa(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)\phi(\mathbf{x}_m) = \|\mathbf{x}_n\|_1 \|\mathbf{x}_m\|_1$.

g) We introduce one slack variable per datapoint: $\xi_n \geq 0$ and allow for points to be on the wrong side of the decision boundary or within the margin boundary. However, we still want to reduce the number of datapoints for which this is the case, so we place a penalty on nonzero ξ_n. This leads to the following primal constrained optimization problem:

$$\min_{a,\hat{h},\{\xi_n\}} \; a^2 + C\sum_{n=1}^{N}\xi_n,$$

with hyperparameter $C > 0$ and the following constraints:

$$\begin{aligned} &(\mathbf{I}) \quad t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) \geq 1 - \xi_n \quad \text{for all} \quad n = 1, ..., N \\ &(\mathbf{II}) \quad \xi_n \geq 0 \quad \text{for all} \quad n = 1, ..., N \\ &(\mathbf{III}) \quad \hat{h} \geq 0 \end{aligned}$$

(1p) for adding the penalty proportional to C.
(1p) for adding the ξ_n correctly into the terms corresponding to modified/new constraints.
The primal Lagrangian then becomes:

$$\mathcal{L}(a, \hat{H}, \xi_n, \{\lambda_n\}_{n=1}^N, \{\mu_n\}_{n=1}^N) = a^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N}\lambda_n[t_n(a\|\mathbf{x}_n\|_1 - \hat{h}) - 1 + \xi_n] - \sum_{n=1}^{N}\mu_n\xi_n - \delta\hat{h}$$

(1p) for the correct corresponding primal Lagrangian.