UNIVERSITY OF AMSTERDAM

Faculty of Science

Exam

Machine Learning 1

Final Exam
Date: December 19, 2018
Time: 9:00-12:00

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

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**

   - [X] Complex models are more 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.
   - [X] A model that suffers from high variance is sensitive to overfitting.

2. We consider linear regression with *maximum likelihood* (ML) and *maximum a posteriori* (MAP) estimates, and Bayesian linear regression. We assume a Gaussian prior over the model parameters $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma_0^2\mathbf{I})$. Choose the correct statements: **/1**

   - [ ] A larger value of σ_0^2, corresponds to a stronger regularization for the MAP optimization of $\mathbf{w}$.
   - [X] In the limit of $\sigma_0^2 \to 0$ the MAP estimate of $\mathbf{w}$ will go to $\mathbf{w}_{\text{MAP}} = \mathbf{0}$.
   - [X] The posterior distribution over $\mathbf{w}$ becomes narrower as more datapoints are observed.
   - [X] In the limit of $\sigma_0^2 \to \infty$ the MAP estimate of $\mathbf{w}$ will be the same as the ML estimate of $\mathbf{w}$.

3. Which of the following statements about classification models are correct? **/1**

   - [ ] In logistic regression we model the class-conditional densities of the input $\mathbf{x}$: $p(\mathbf{x}|\mathcal{C}_k)$.
   - [X] The perceptron algorithm is a discriminant function algorithm.
   - [ ] The decision boundary of a probabilistic generative model with Gaussian class conditional densities $p(\mathbf{x}|\mathcal{C}_k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$ is linear if all the covariance matrices Σ_k are different.
   - [ ] If we apply the Naive Bayes assumption to the class-conditional densities in a probabilisitic generative classification model, then this implies that for each input vector $\mathbf{x} = (x_1, ..., x_D)^T$, the individual features are marginally independent: $p(\mathbf{x}) = p(x_1)...p(x_D)$.

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 $E_{\text{train}}(\mathbf{w}, \lambda)$:

	$E_{\text{train}}(\mathbf{w}, \lambda)$	$E_{\text{val}}(\mathbf{w}, \lambda)$
$\lambda_1 = 1$	0.51	0.55
$\lambda_2 = 0.1$	0.23	0.26
$\lambda_3 = 0.01$	0.12	0.15

Which of the following statements are correct? **/1**

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

5. Which of the following statements about training a neural network are correct? We denote the error function/loss function as $E = \sum_{n=1}^{N} E_n$ for N datapoints. **/1**

- [X] If we do minibatch gradient descent with a batch size M, in the forward propagation step we take M datapoints as input, and then compute all of the activations of the hidden and output units for each datapoint.
- [ ] In the backpropagation step the derivatives of the error function E_n with respect to the weights closest to the input layer are first computed. The derivatives with respect to the weights in the last layer (closest to the output) is computed last.
- [ ] The loss function for neural networks in general has only one global optimum.
- [X] Stochastic gradient descent is less sensitive to get stuck in local minima than full batch gradient descent.

6. 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] If $\theta_0 = 0$, $\theta_1 = 1$, $\theta_2 = 5$ and $\theta_3 = 0$, 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.
- [ ] The parameter θ_0 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 $\mathbf{x}$.
- [ ] The parameter θ_1 determines the amplitude of the oscillations of a function drawn from this GP.

7. Indicate which of the following statements about Gaussian processes and support vector machines are correct: **/1**

- [X] Kernel functions that are positive definite can be used for Gaussian processes and support vector machines.
- [ ] Support vector machines for binary classification are probabilistic models.
- [X] When using a kernel of the form $k(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\frac{1}{2\theta_1^2}||\mathbf{x}_n - \mathbf{x}_m||^2\right)$ with $\theta_1 > 0$ in a Gaussian process, the predictive variance is lower for test points that are closer to training points than for those that are far away from any training point.
- [X] The hard margin limit of a maximum margin classifier can be obtained by enforcing an infinitely large penalty on nonzero slack variables in the case of a soft margin classifier.

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

- [X] Every update step in the K-means algorithm decreases the loss function or leaves it unchanged.
- [ ] In Gaussian mixture models the number of clusters is also learned.
- [ ] In Gaussian mixture models, the covariance matrices of the different clusters have to be the same.
- [ ] The K-means algorithm converges to a global optimum.

9. Which of the following statements about principle component analysis (PCA) are correct? **/1**

- [X] PCA is a linear projection method.
- [ ] Applying PCA to binary data will result in a projected dataset which is also binary.
- [X] PCA can be used to decorrelate a dataset.
- [ ] In PCA the data is projected to a lower dimensional subspace such that the variance of the projected data is minimized.

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

- [X] Boosting performs sequential training of base classifiers.
- [ ] Bootstrap datasets are created by sampling without replacement from one single original dataset.
- [X] 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 highly correlated, then the predictions of the different trained models will also be highly correlated.
- [X] A decision tree divides the input space into rectangular decision regions.

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 $10 \times \frac{\#\text{points}}{46}$.

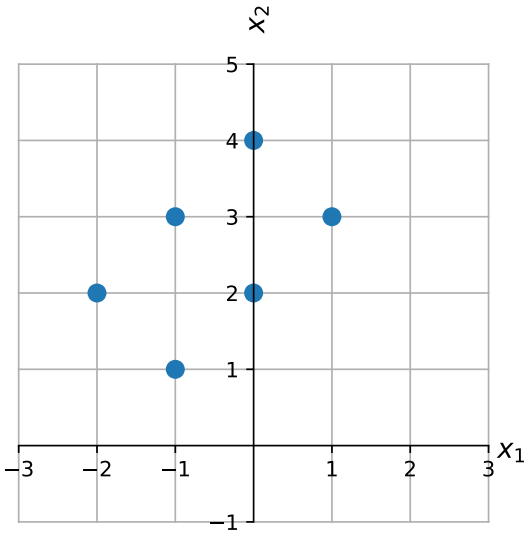
General remarks

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 PCA

/4

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 $1/2$ 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**

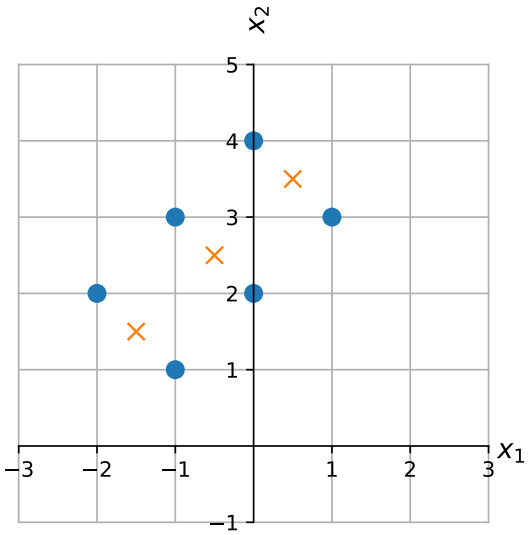
Solutions

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 $4/3$, and $1/2$ corresponds to $\mathbf{u}_2$ (1p).

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



3 Logistic regression for K classes with a generalized Gaussian prior

/9

Consider logistic regression for K classes with N training vectors $\{\mathbf{x}_n\}_{n=1}^N$, each of which is mapped to a different feature vector $\boldsymbol{\phi}(\mathbf{x}_n) = \boldsymbol{\phi}_n = (\phi_0(\mathbf{x}_n), \phi_1(\mathbf{x}_n), ..., \phi_{M-1}(\mathbf{x}_n))^T$ using basis functions $\phi_j(\mathbf{x})$ with $j = 0, ..., M-1$, and $\phi_0(\mathbf{x}) = 1$. Each vector $\mathbf{x}_n$ has a corresponding target vector $\mathbf{t}_n$ of size K: $\mathbf{t}_n = (t_{n1}, t_{n2}, ..., t_{nK})^T$, where $t_{nk} = 1$ if $\mathbf{x}_n \in \mathcal{C}_k$, and $t_{nj} = 0$ for all $j \neq k$. The input data can be collected in a matrix $\mathbf{X}$ with row n given by $\mathbf{x}_n^T$, and the targets can be collected in a target matrix $\mathbf{T}$, with row n equal to $\mathbf{t}_n^T$. The feature vectors can also be collected in a matrix $\boldsymbol{\Phi}$ such that the n-th row of $\boldsymbol{\Phi}$ contains $\boldsymbol{\phi}_n^T$:

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Assuming i.i.d. data, the posterior class probabilities are modeled by

$$p(\mathcal{C}_k|\boldsymbol{\phi}(\mathbf{x}), \mathbf{w}_1, ..., \mathbf{w}_K) = y_k(\boldsymbol{\phi}) = \frac{\exp(a_k(\boldsymbol{\phi}))}{\sum_{j=1}^K \exp(a_j(\boldsymbol{\phi}))},$$

where $a_k(\boldsymbol{\phi}) = \mathbf{w}_k^T \boldsymbol{\phi}$ with $\boldsymbol{\phi} = \boldsymbol{\phi}(\mathbf{x})$, and $\mathbf{w}_k = (w_{k0}, ..., w_{kM-1})^T$. Assume the following generalized Gaussian prior on the parameter vectors $\mathbf{w}_1, ..., \mathbf{w}_K$:

$$p(\mathbf{w}_1, ..., \mathbf{w}_K|\alpha, q) = \prod_{k=1}^K \left(\frac{q}{2\alpha\Gamma(\frac{1}{q})} \right)^M e^{-\|\mathbf{w}_k\|_q^q/\alpha^q}. \tag{1}$$

Here, $\alpha > 0$ is a scale parameter, and $q > 0$ determines the shape of the distribution. Both α and q can be any real number larger than zero. Furthermore, $\|\mathbf{w}_k\|_q^q$ denotes the q-norm of the vector $\mathbf{w}_k$ to the power q, defined as $\|\mathbf{w}_k\|_q^q = \sum_{m=0}^{M-1} |w_{km}|^q$. $\Gamma(\frac{1}{q})$ is a normalization constant.

Answer the following questions:

a) Consider two different weight vectors $\mathbf{w}_k$ and $\mathbf{w}_l$ ($k \neq l$). Are they correlated according to the prior in Eq. (1)? Are two different elements of the same weight vector $\mathbf{w}_k$, such as w_{k1} and w_{k2}, correlated? In both cases explain your answer. **/2**

b) Write down the equation for the posterior $p(\mathbf{w}_1, ..., \mathbf{w}_K|\boldsymbol{\Phi}, \mathbf{T}, \alpha, q)$ in terms of the data likelihood $p(\mathbf{T}|\boldsymbol{\Phi}, \mathbf{w}_1, ..., \mathbf{w}_K)$ and the prior $p(\mathbf{w}_1, ..., \mathbf{w}_K|\alpha, q)$. You do not need to insert the actual distributions. **/1**

c) Compute the log-likelihood $\ln p(\mathbf{T}|\boldsymbol{\Phi}, \mathbf{w}_1, ..., \mathbf{w}_K)$ and the log of the prior $\ln p(\mathbf{w}_1, ..., \mathbf{w}_K|\alpha, q)$. **/2**

d) Using your results in b) and c), show that the optimization problem corresponding to obtaining a Maximum A Posteriori (MAP) estimate for $\mathbf{w}_1, ..., \mathbf{w}_K$ is equivalent to performing regularized logistic regression for K classes, where we minimize the function

$$-\sum_{n=1}^N \sum_{k=1}^K t_{nk} \ln y_k(\boldsymbol{\phi}_n) + \lambda \sum_{k=1}^K \sum_{m=0}^{M-1} |w_{km}|^q$$

with respect to $\mathbf{w}_1, ..., \mathbf{w}_K$, and with λ a regularization penalty parameter. How is λ related to α and q? **/3**

e) Consider the generalized Gaussian distribution for a single stochastic variable w:

$$p(w|\alpha, q) = \frac{q}{2\alpha\Gamma(\frac{1}{q})} e^{-|w|^q/\alpha^q}.$$

In the figure below on the left, this distribution is shown for different values of q, and $\alpha = 1$. On the right you see contours plotted for $|w_1|^q + |w_2|^q = \mu$, for some arbitrary μ, and different values of q. Imagine that you train several logistic regression models with MAP estimates for $\mathbf{w}_1, ..., \mathbf{w}_K$, for different values of $q > 0$ (and $\alpha = 1$). Which trained models will have sparser weight vectors, the ones with $q > 1$ or those with $q \leq 1$? Remember, the sparsity of a weight vector is determined by the number of elements of the vector that are zero (up to numerical precision). Explain your answer. **/1**

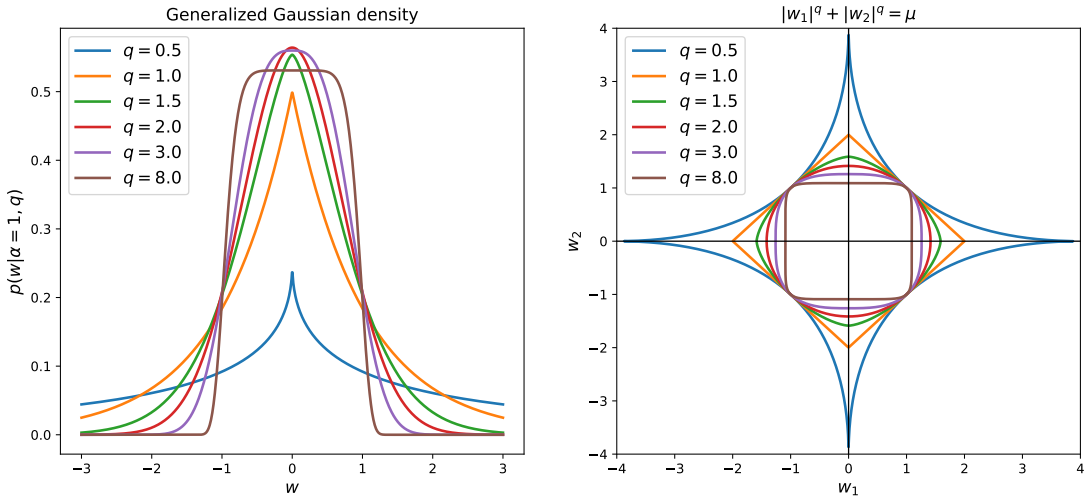


Figure: (left) Generalized Gaussian distribution for a single random variable w. (right) Contours corresponding to $|w_1|^q + |w_2|^q = \mu$ for different q.

Solutions

a) The prior factorizes like $p(\mathbf{w}_1, ..., \mathbf{w}_K|\alpha, q) = p(\mathbf{w}_1|\alpha, q)p(\mathbf{w}_2|\alpha, q)...p(\mathbf{w}_K|\alpha, q)$, where

$$p(\mathbf{w}_k|\alpha, q) = \left(\frac{q}{2\alpha\Gamma(\frac{1}{q})}\right)^M e^{-\|\mathbf{w}_k\|_q^q/\alpha^q}.$$

This means $\mathbf{w}_k$ and $\mathbf{w}_l$ for $(k \neq l)$ are independent, and independent variables are uncorrelated (1p). We can furthermore factorize the distribution of a single weight vector $\mathbf{w}_k$ as

$$p(\mathbf{w}_k|\alpha, q) = \left(\frac{q}{2\alpha\Gamma(\frac{1}{q})}\right)^M e^{-\|\mathbf{w}_k\|_q^q/\alpha^q} = \left(\frac{q}{2\alpha\Gamma(\frac{1}{q})}\right)^M e^{-\sum_{m=0}^{M-1}|w_{km}|^q/\alpha^q}$$
$$= \prod_{m=0}^{M-1} \frac{q}{2\alpha\Gamma(\frac{1}{q})} e^{-|w_{km}|^q/\alpha^q} = \prod_{m=0}^{M-1} p(w_{km}|\alpha, q)$$

This means that two different individual elements of the same weight vector $\mathbf{w}_k$, such as w_{k1} and w_{k2}, are independent, and independent variables are uncorrelated (1p).

b) The posterior distribution over the parameters is given by (1p)

$$p(\mathbf{w}_1, ..., \mathbf{w}_K|\mathbf{\Phi}, \mathbf{T}, \alpha, q) = \frac{p(\mathbf{T}|\mathbf{\Phi}, \mathbf{w}_1, ..., \mathbf{w}_K)p(\mathbf{w}_1, ..., \mathbf{w}_K|\alpha, q)}{p(\mathbf{T}|\mathbf{\Phi}, \alpha, q)}.$$

c) The log-likelihood is given by

$$\ln p(\mathbf{T}|\mathbf{\Phi}, \mathbf{w}_1, ..., \mathbf{w}_K) = \ln \prod_{n=1}^{N} \prod_{k=1}^{K} y_k(\boldsymbol{\phi}_n)^{t_{nk}} = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\boldsymbol{\phi}_n). \text{ (1 pt)}$$

The log of the prior is given by

$$\begin{aligned}
\ln p(\mathbf{w}_1, ..., \mathbf{w}_K|\alpha, q) =& \ln \prod_{k=1}^{K} \left(\frac{q}{2\alpha\Gamma(\frac{1}{q})} \right)^M e^{-\|\mathbf{w}_k\|_q^q/\alpha^q} \\
=& KM \ln q - KM \ln 2\alpha - KM \ln \Gamma(\frac{1}{q}) - \frac{1}{\alpha^q} \sum_{k=1}^{K} \|\mathbf{w}_k\|_q^q \\
=& KM \ln q - KM \ln 2\alpha - KM \ln \Gamma(\frac{1}{q}) - \frac{1}{\alpha^q} \sum_{k=1}^{K} \sum_{m=0}^{M-1} |w_{km}|^q \text{ (1 pt)}
\end{aligned}$$

d) The MAP estimate for $\mathbf{w}_1, ..., \mathbf{w}_K$ is obtained by maximizing the posterior distribution with respect to the parameters $\mathbf{w}_1, ..., \mathbf{w}_K$:

$$\begin{aligned}
\mathbf{w}_1^{\text{MAP}}, ..., \mathbf{w}_K^{\text{MAP}} &= \arg\max_{\mathbf{w}_1,...,\mathbf{w}_K} p(\mathbf{w}_1, ..., \mathbf{w}_K|\mathbf{\Phi}, \mathbf{T}, \alpha) = \arg\min_{\mathbf{w}_1,...,\mathbf{w}_K} -\ln p(\mathbf{w}_1, ..., \mathbf{w}_K|\mathbf{\Phi}, \mathbf{T}, \alpha) \\
&= \arg\min_{\mathbf{w}_1,...,\mathbf{w}_K} -\ln p(\mathbf{T}|\mathbf{\Phi}, \mathbf{w}_1, ..., \mathbf{w}_K) - \ln p(\mathbf{w}_1, ..., \mathbf{w}_K|\alpha), \text{ (2 pt)}
\end{aligned}$$

where we have used the fact that $\frac{\partial}{\partial \mathbf{w}_j} - \ln p(\mathbf{T}|\mathbf{\Phi}, \alpha) = 0$ for $j = 1, ..., K$. Noting that the first three terms of the log-prior are independent of $\mathbf{w}_1, ..., \mathbf{w}_K$, we obtain

$$\mathbf{w}_1^{\text{MAP}}, ..., \mathbf{w}_K^{\text{MAP}} = \arg\min_{\mathbf{w}_1,...,\mathbf{w}_K} -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\boldsymbol{\phi}_n) + \frac{1}{\alpha^q} \sum_{k=1}^{K} \sum_{m=0}^{M-1} |w_{km}|^q,$$

such that $\lambda = 1/\alpha^q$. (1 pt)

e) The generalized Gaussian distributions for $q \leq 1$ have sharper peaks around $w = 0$ and heavier tails, as compared to those for $q > 1$. This means that models with $q \leq 1$ result in more sparse weight vectors (1p).

4 Mixture of geometric distributions

/9

The geometric distribution is a discrete probability distribution, where the discrete random variable describes the number of Bernoulli trials that are needed to get one success. The random variable can thus take on any positive integer value. In this exercise we will consider a dataset $\boldsymbol{X} = \{x_n\}_{n=1}^N$, where each $x_n \in \{1, 2, 3, ...\}$ is a discrete random variable that represents the number of throws required to throw the number 2 with a six-sided die. The success event thus corresponds to throwing a 2, and an unsuccessful event corresponds to throwing $1, 3, 4, 5$ or 6. We assume that there are K different dice, and that the datapoints are generated as follows:

i) The dice 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 die k, and are unknown, so they need to be learned.

ii) For a datapoint x that was obtained by throwing die $z = k$, $x \in \{1, 2, 3, ...\}$ is sampled from a geometric distribution with parameter p_k:

$$p(x|z = k) = (1 - p_k)^{x-1} p_k \ .$$

The parameters p_k correspond to the probability of throwing a 2 with die $z = k$, and need to be learned.

Answer the following questions.

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

b) Compute the joint probability of a datapoint x_n and the corresponding latent variable z_n that represents the identity of a dice, $p(x_n, z_n)$, as a function of x_n and z_n. Compute the marginal probability of x_n under this model, denoted with $p(x_n)$. /2

c) Compute the responsibility (or posterior) $r_{nk} = p(z = k|x_n)$ of die k having generated the datapoint x_n. /1

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

$$\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{X})} \left[\ln p(\boldsymbol{X}, \boldsymbol{Z}|\{\pi_k\}_{k=1}^K, \{p_k\}_{k=1}^K)\right] = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \left(\ln \pi_k + (x_n - 1) \ln(1 - p_k) + \ln p_k\right) \ . \tag{2}$$

Maximize Eq. (2) with respect to all p_k for fixed responsibilities r_{nk}. Use this to write down the update rule for p_k as a function of the responsibilities r_{nk}. You may assume $0 < p_k < 1$ for all $k = 1, ..., K$. /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, $\{\pi_k\}_{k=1}^K$ and $\{p_k\}_{k=1}^K$. 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) The joint is given by

$$p(x_n, z_n) = p(x_n|z_n)p(z_n) = \prod_{k=1}^K \left[p(x_n|z_n=k)p(z_n=k)\right]^{I[z_n=k]} = \prod_{k=1}^K \left[(1-p_k)^{x_n-1}p_k\pi_k\right]^{I[z_n=k]} \quad .(1\text{pt})$$

The marginal yields

$$p(x_n) = \sum_{k=1}^K \pi_k (1-p_k)^{x_n-1} p_k \quad .(1\text{pt})$$

c) By Bayes theorem:

$$\begin{aligned} r_{nk} = p(z=k|x_n) &= \frac{p(x_n|z=k)p(z=k)}{p(x_n)} \\ &= \frac{\pi_k(1-p_k)^{x_n-1}p_k}{\sum_{j=1}^K \pi_j(1-p_j)^{x_n-1}p_j} \quad . \end{aligned}$$

(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 p_k.

$$\begin{aligned} &\frac{\partial}{\partial p_k} \sum_{n=1}^N \sum_{k=1}^K r_{nk} \left[\ln \pi_k + (x_n - 1)\ln(1-p_k) + \ln p_k\right] \\ &= \sum_{n=1}^N r_{nk} \frac{\partial}{\partial p_k} \left[(x_n-1)\ln(1-p_k) + \ln p_k\right] \\ &= \sum_{n=1}^N r_{nk} \left(\frac{x_n - 1}{1-p_k} \cdot (-1) + \frac{1}{p_k}\right) \\ &= \sum_{n=1}^N r_{nk} \left[\frac{1 - p_k x_n}{p_k(1-p_k)}\right] = 0 \quad . \ (1\text{pt}) \end{aligned}$$

Assuming $0 < p_k < 1$ for all $k = 1, ..., K$,

$$\sum_{n=1}^N r_{nk}\left(1 - p_k x_n\right) = 0 \rightarrow p_k = \frac{\sum_{n=1}^N r_{nk}}{\sum_{n=1}^N r_{nk} x_n} \quad .(1\text{pt})$$

Note that this is a sensible answer if we consider the case where $r_{nk} = 1$ for all datapoints for one particular k, and look at the values of p_k for different x_n. If almost all $x_n \geq 1$ are equal to 1, then the resulting p_k above will be close to 1, so throwing the die k will very often yield 2 (success), which is consistent with x_n almost always returning 1.

e) We need to optimize the Lagrangian

$$L = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\left[\ln \pi_k + (x_n - 1)\ln(1-p_k) + \ln p_k\right] + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right),$$

with respect to π_k and the Lagrange multiplier λ.

$$\begin{aligned}
&\frac{\partial}{\partial \pi_k}\left[\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\left[\ln \pi_k + (x_n - 1)\ln(1-p_k) + \ln p_k\right] + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)\right] \\
&= \sum_{n=1}^{N} r_{nk}\frac{\partial}{\partial \pi_k}\ln \pi_k + \frac{\partial}{\partial \pi_k}\lambda\left(\sum_{k=1}^{K}\pi_k - 1\right) \\
&= \sum_{n=1}^{N}\frac{r_{nk}}{\pi_k} + \lambda \quad \rightarrow \quad \pi_k = -\frac{1}{\lambda}\sum_{n=1}^{N} r_{nk} \text{ . (1pt)}
\end{aligned}$$

Use that

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

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

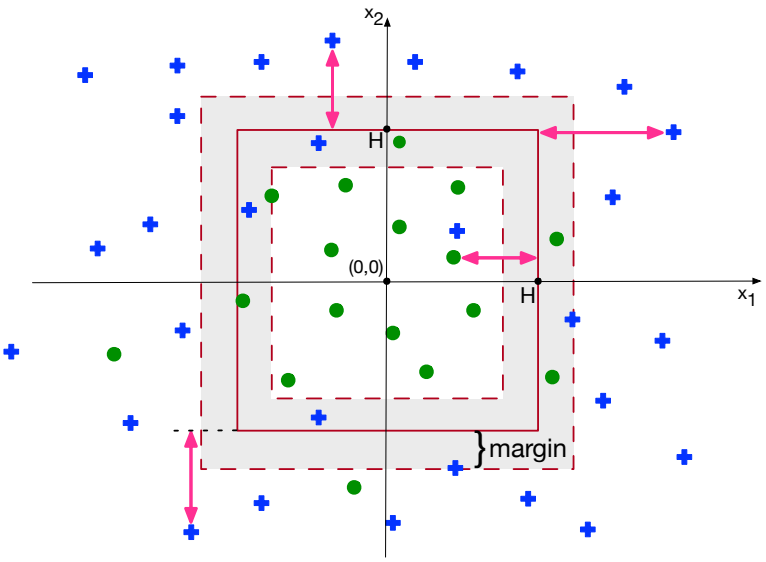
$$\pi_k = \frac{1}{N}\sum_{n=1}^{N} r_{nk} \text{ . (1pt)}$$

f) Updating all of the p_k and π_k is the Maximization step in the Expectation-Maximization algorithm, where we update all the parameters and we keep the posterior fixed. In the Expectation-step, we re-compute the posterior probabilities r_{nk}, while keeping all the parameters fixed. (1pt)

5 Maximum margin classifier: square 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 the figure 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 almost perfectly separable by a square decision boundary, with equal height and width $H \geq 0$. Our goal is to design a maximum margin classifier with square decision and margin boundaries. The margin size is indicated in the figure with the black curly bracket.



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

$$t_n(\|\mathbf{x}_n\|_\infty - H) = \frac{t_n(\alpha\|\mathbf{x}_n\|_\infty - \hat{H})}{\alpha},$$

where $\hat{H} = \alpha H \geq 0$, and $\alpha > 0$. Here, $\|\mathbf{x}_n\|_\infty = \max(|x_{n1}|, |x_{n2}|)$ is the norm that returns the maximum of the absolute values of the elements of $\mathbf{x}_n$. Note that the "distance" to the decision boundary is invariant to a rescaling $\alpha \to \kappa\alpha$. We can use this to set

$$t_n(\alpha\|\mathbf{x}_n\|_\infty - \hat{H}) = 1$$

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. However, we know that our data is not perfectly separable by a square decision boundary, so we introduce slack variables $\xi_n \geq 0$ for each datapoint, such that

$$t_n(\alpha\|\mathbf{x}_n\|_\infty - \hat{H}) \geq 1 - \xi_n \quad \text{for} \quad n = 1, ..., N.$$

We introduce a penalty for all datapoints for which $\xi_n > 0$. This leads to the following primal constrained optimization problem:

$$\min_{\alpha, \hat{H}, \{\xi_n\}} \alpha^2 + C \sum_{n=1}^{N} \xi_n,$$

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

(**I**) $t_n(\alpha\|\mathbf{x}_n\|_\infty - \hat{H}) \geq 1 - \xi_n$ for all $n = 1, ..., N$

(**II**) $\xi_n \geq 0$ for all $n = 1, ..., N$

(**III**) $\hat{H} \geq 0$

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**), $\{\mu_n\}_{n=1}^N$ for constraints (**II**), and δ for constraint (**III**). Which variables are the primal variables? Which variables are the dual variables? **/3**

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? **/3**

d) Optimize the Lagrangian with respect to the primal variables. This should give you a set of additional conditions on the Lagrange multipliers. **/3**

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**

Solutions

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

$$\alpha\|\mathbf{x}_n\|_\infty - \hat{H} = 1 \rightarrow \|\mathbf{x}_n\|_\infty = \frac{1}{\alpha} + \frac{\hat{H}}{\alpha} = H + \frac{1}{\alpha}.$$

So the "distance" to the boundary as indicated in the figure is given by $1/\alpha$ (1p).

b)

$$\mathcal{L}(R, \boldsymbol{\mu}, \xi_n) = \alpha^2 + C\sum_{n=1}^N \xi_n - \sum_{n=1}^N \lambda_n(t_n(\alpha\|\mathbf{x}_n\|_\infty - \hat{H}) - 1 + \xi_n) - \sum_{n=1}^N \mu_n\xi_n - \delta\hat{H}$$

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

c)

$$\lambda_n \geq 0$$
$$t_n(\alpha\|\mathbf{x}_n\|_\infty - \hat{H}) - 1 + \xi_n \geq 0$$
$$\lambda_n\{t_n(\alpha\|\mathbf{x}_n\|_\infty - \hat{H}) - 1 + \xi_n\} = 0$$

$$\mu_n \geq 0$$
$$\xi_n \geq 0$$
$$\mu_n\xi_n = 0$$

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

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

(2p) for all constraints with correct signs, and (1p) for the number $2 \cdot 3N + 3 = 6N + 3$.

d)

$$\frac{\partial \mathcal{L}}{\partial \alpha} = 2\alpha - \sum_{n=1}^{N} \lambda_n t_n \|\mathbf{x}_n\|_\infty = 0 \quad \rightarrow \quad \alpha = \frac{1}{2} \sum_{n=1}^{N} \lambda_n t_n \|\mathbf{x}_n\|_\infty$$
$$\frac{\partial \mathcal{L}}{\partial \hat{H}} = \sum_{n=1}^{N} \lambda_n t_n - \delta = 0 \quad \rightarrow \quad \delta = \sum_{n=1}^{N} \lambda_n t_n$$
$$\frac{\partial \mathcal{L}}{\partial \xi_n} = C - \lambda_n - \mu_n = 0 \quad \rightarrow \quad \lambda_n = C - \mu_n.$$

(1p) 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

$$\begin{aligned}\tilde{\mathcal{L}}(\{\lambda_n\}, \gamma) =& \alpha \left(\alpha - \sum_{n=1}^{N} \lambda_n t_n \|\mathbf{x}_n\|_\infty \right) + H \left(\sum_{n=1}^{N} \lambda_n t_n - \delta \right) + \sum_{n=1}^{N} \xi_n (C - \lambda_n - \mu_n) + \sum_{n=1}^{N} \lambda_n \\ =& \alpha (\alpha - 2\alpha) + \sum_{n=1}^{N} \lambda_n \\ =& -\frac{1}{4} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n \lambda_m t_n t_m \|\mathbf{x}_n\|_\infty \|\mathbf{x}_m\|_\infty + \sum_{n=1}^{N} \lambda_n \quad \text{(1p)}\end{aligned}$$

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 $\lambda_n \geq 0$. In c) we also found that $\lambda_n = C - \mu_n$, and in d) we saw that $\mu_n \geq 0$, hence $0 \leq \lambda_n \leq C$ (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 $\kappa(\mathbf{x}_n, \mathbf{x}_m) = \|\mathbf{x}_n\|_\infty \|\mathbf{x}_m\|_\infty$ (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 or size 1. You can also see from the problem setup that we are using $\phi(\mathbf{x}) = \|\mathbf{x}\|_\infty$ as a feature vector (actually a feature scalar), so that $\kappa(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)\phi(\mathbf{x}_m) = \|\mathbf{x}_n\|_\infty \|\mathbf{x}_m\|_\infty$.