Which of the following statements are true? Check all that apply

1.0 maximum point · Multiple choice · 4 choices

☐	Higher complexity models are more prone to overfitting and typically have lower variance	-1.0
☑	**If we only add more training samples for training a learner with high bias, the test error may not decrease.**	0.5
☑	**Overfitting may arise when relevant features are missing in the data**	0.5
☐	Increasing the depth of a neural network will always reduce the test error.	-1.0

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

25 Getting Ill During a Pandemic

4.0 points • 2 questions

Your boss is demanding you to come to the office during a pandemic caused by a dangerous virus, despite everyone knowing the chances of contracting the disease during an office visit. It turns out that you have a chance of 20% to contract the virus during a regular workday and 10% during the weekend. Your boss tries to be generous and decides to let you work during the weekend every now and then. He decides on a weekly basis what days you should work, but for now only says, "you'll be working 70% of your days during the weekend".

Description

a You have been working so much lately that you don't even know anymore what day it is, the only thing you know is that your alarm went of and you have to go to the office. Without knowing what day it is, what is your chance of contracting the virus?

Fill in the answer as a percentage, rounded to the nearest integer (so fill in an integer between 0 and 100).

2.0 maximum points · Numerical question · _pVirus = 13

Grading description

Question is autograded and is solved by marginalizing out the day-type out of the joint distribution which is obtained via the product rule:

$$p(virus) = p(virus,day=work\text{-}day) + p(virus, day = weekend)$$

with

$$p(virus,day) = p(virus \mid day)*p(day)$$

Answer

_pVirus

Margin

1.0 Absolute

Feedback

Feedback if the question is completely correct

Feedback if the question is completely incorrect

b Bad luck strikes, you get ill, but luckily you recover well under the doctor's treatment. The doctor would like to know when you have contracted the disease. You have no idea but you could give him the odds that it happened during the weekend. What is the chance that you contracted the virus at work during the weekend?

Fill in the answer as a percentage, rounded to the nearest integer (so fill in an integer between 0 and 100).

2.0 maximum points · Numerical question · _pWeekendGivenVirus = 54

Grading description

Question is autograded and is solved via Bayes rule:

p(day | virus) = p(virus|day)p(day)/p(virus)

Answer

_pWeekendGivenVirus

Margin

1.0 Absolute

Feedback

Feedback if the question is completely correct

Feedback if the question is completely incorrect

26 K-Means Iterations

13.0 points • 5 questions

Consider the unlabeled dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of 2D points $x_n \in \mathbb{R}^2$. We want to learn something about the structure of the data and decide to apply the K-means algorithm to split the data in three separate groups ("1", "2", and "3").

Before we do so, let's recap what we know about the K-means algorithm.

Description

a Which of the following statements is true?

1.0 maximum point · Multiple choice · 3 choices

○ K-means is a supervised learning algorithm	0.0
◉ **K-means is an unsupervised learning algorithm**	1.0
○ K-means is a semi-supervised learning algorithm	0.0

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

b Which of the following statements is true?

1.0 maximum point · Multiple choice · 4 choices

☑ **In K-means the data is modeled by a discrete latent variable model**	0.5
☐ In K-means the data is modeled by a continuous latent variable model	-1.0
☑ **The latent variables in K-means are the cluster identities of each data point.**	0.5
☐ The latent variables in K-means are the centroids of the clusters to which a data point can belong.	-1.0

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

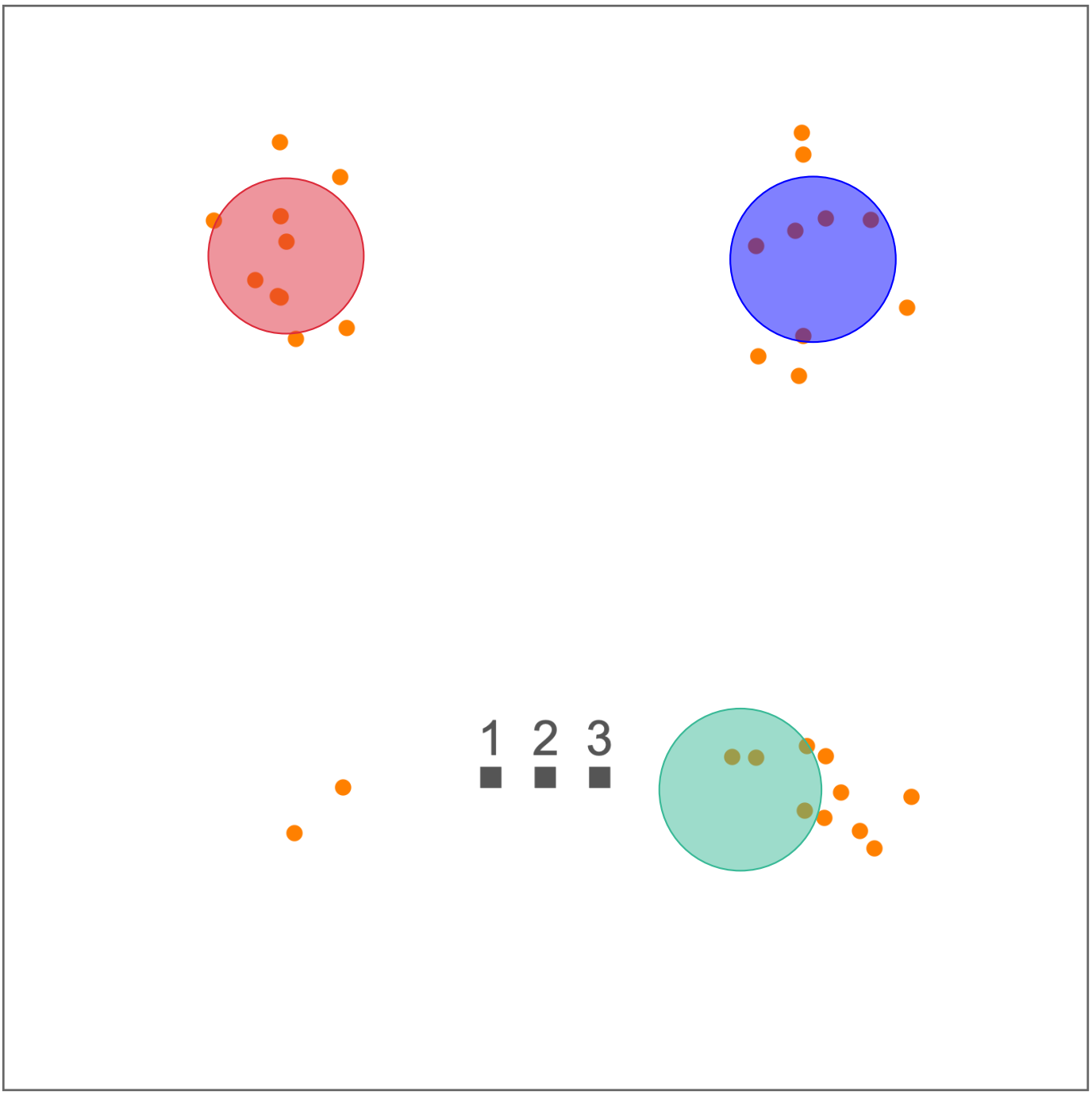
c In the figure below, indicate where you expect the cluster centers ("1", "2" and "3") to be after 1000 K-means update iterations, provided that the cluster centers are initialized at the points indicated by the gray rectangle markers.

3.0 maximum points · Hotspot

1 red - Area 1

2 blue - Area 3

3 green - Area 2



Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

d The K-means algorithm can be seen as a version of the E-M algorithm. Explain what happens in the E step and what happens in the M step.

2.0 maximum points · Open question

+1 point

In the E-step data points are assigned to one of the K latent classes. The class that is assigned is the one that corresponds to then nearest cluster center.

+1 point

In the M-step the cluster centers are updated by taking the mean over all datapoints within the same cluster.

e In EM algorithms the E stands for "Expectation" and the M for "Maximization" and the terminology is typically used in the probabilistic setting when optimizing Mixture Models. Explain why the Expectation/Maximization terminology is also sensible in the K-means clustering algorithm. Do so by indicating the correspondences between steps in the probabilistic and K-means setting.

(It is not necessary to explain the steps in terms of mathematical formulas)

6.0 maximum points · Open question

+1 point

E-step, probabilistic setting: In the expectation step the expected responsibilities, or posterior probabilities, are computed.

+1 point

E-step, K-means: Even though K-means is a non-probabilistic method, the cluster assignments can be thought of as determining the probabilities (which is either 1 or 0) of a datapoint for each class with a one-hot encoding.

+1 point

M-step, probabilistic setting: In the probabilistic setting maximizing refers to maximizing the likelihood of the mixture model, where maximization is done w.r.t. the model parameters.

+1 point

M-step, probabilistic setting: in the maximization step the model parameters of the individual distributions in the mixture (such as e.g. means of Gaussians) are the parameters that are optimized over.

+1 point

M-step, K-means: Instead of minimizing the log-likelihood, K-means minimizes a "K-means objective" (not necessary to specify it) which is similar to the log-likelihood in the probabilistic setting. (important in this answer is mentioning that K-means minimizes an objective).

+1 point

M-step, K-means: In the M step, the cluster centers are the model parameters that are optimized, where as in the probabilistic setting the model parameters are those of the distributions in the mixture.

27 Kernel Ridge Regression

18.0 points • 13 questions

We are given the data set $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^N$, where each $\mathbf{x}_i \in \mathbb{R}^d$ and $t_i \in \mathbb{R}$. We are also given a collection of feature functions $\{\phi_a(\cdot)\}_{a=1}^K$ that we can use to generate a new feature vector $\boldsymbol{\phi}(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \ldots, \phi_K(\mathbf{x}_i))^T \in \mathbb{R}^K$. Now consider the following optimization problem:

$$\min_{\mathbf{w}} \ \sum_{i=1}^N (\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - t_i)^2,$$

$$\text{subject to } \ \|\mathbf{w}\|^2 \leq C.$$

Description

[Latex support]

Some of the math expressions used in this exercise are generated in latex code as follows:

- $\min_{\mathbf{w}} \ \sum_{i=1}^N (\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - t_i)^2,$

\underset{\mathbf{w}}{\operatorname{min}} \;\; \sum_{i=1}^N (\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - t_i)^2,

- $\|\mathbf{w}\|^2 \leq C.$

\lVert \mathbf{w} \rVert^2 \leq C.

- β

\beta

- Remember to place latex code in one line between dollar signs via `$$code in one line$$`

/[Latex support]

Description

a Provide an expression for the Lagrangian L. Use β as a symbol to denote the Lagrange multiplier.

(Hint: check that the sign of the Lagrange multiplier is correct)

2.0 maximum points · Open question

Grading description

For primal problems of the form

$$\min_{\mathbf{w}} \; f(\mathbf{w}) \quad \text{subject to} \; \; g(\mathbf{w}) \geq 0$$

the Lagrangian is given by

$$L(\mathbf{w}, \beta) = f(\mathbf{w}) - \beta \, g(\mathbf{w}).$$

We recognize that the constraint $g(\mathbf{w}) \geq 0$ can be written as $C - \|w\|^2 \geq 0$. Filling this in gives

$$L(\mathbf{w}, \beta) = \sum_{i=1}^{N} (\mathbf{w}^T \phi(\mathbf{x}_i) - t_i)^2 - \beta(C - \|w\|^2),$$

or written slightly differently

$$L(\mathbf{w}, \beta) = \sum_{i=1}^{N} (\mathbf{w}^T \phi(\mathbf{x}_i) - t_i)^2 + \beta(\|w\|^2 - C),$$

with $\beta \geq 0$.

+1 point

One point for the general form of the Lagrangian

+1 point

One point for having the sign correct.

b Is minimizing the Lagrangian L with respect to $\mathbf{w}$ a convex optimization problem?

1.0 maximum point · Multiple choice · 2 choices

- (●) Yes. 1.0
- ( ) No. 0.0

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

c Explain why it is or isn't a convex optimization problem.

1.0 maximum point · Open question

+0.34 points

The primal variable $\mathbf{w}$ appears in the Lagrangian both in the first term, which is convex, ...

+0.33 points

... as well as in the second term (the constraint) which is also convex.

+0.33 points

The problem is convex because the sum of these convex terms is also convex.

[Latex support]

- $\mathbf{\Phi}$

\boldsymbol{\Phi}

/[Latex support]

Description

d

Let us define the design matrix $\mathbf{\Phi} = \begin{pmatrix} - & \boldsymbol{\phi}(\mathbf{x}_1)^T & - \\ & \vdots & \\ - & \boldsymbol{\phi}(\mathbf{x}_N)^T & - \end{pmatrix}$ which is a $[N \times K]$ matrix. And let us store all target values in the vector $\mathbf{t} = (t_1, \ldots, t_N)^T$. Write down the Lagrangian in terms of the design matrix $\mathbf{\Phi}$ and target vector $\mathbf{t}$.

1.0 maximum point · Open question

Grading description

$L(\mathbf{w}, \beta) = \mathbf{w}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w} - 2\mathbf{w}^T\mathbf{\Phi}^T\mathbf{t} + \mathbf{t}^T\mathbf{t} + \beta(\mathbf{w}^T\mathbf{w} - C)$

+0.5 points

For writing the inner products in matrix vector form with $\mathbf{\Phi}$, $\mathbf{w}$ and $\mathbf{t}$

+0.5 points

For then also writing out the "square".

e Derive a closed form expression for the minimizer (with respect to $\mathbf{w}$) of the primal Lagrangian L for a fixed β. Call this minimizer $\mathbf{w}^*(\beta)$.

3.0 maximum points · Open question

+1 point

We need to solve:

$\nabla_{\mathbf{w}} L(\mathbf{w}, \beta) = 0$

+2 points

Two points for the actual computation:

$$\nabla_{\mathbf{w}} L(\mathbf{w}, \beta) = 0$$
$$\Leftrightarrow$$
$$\nabla_{\mathbf{w}} (\mathbf{w}^T \mathbf{\Phi}^T \mathbf{\Phi} \mathbf{w} - 2\mathbf{w}^T \mathbf{\Phi}^T \mathbf{t} + \mathbf{t}^T \mathbf{t} + \beta(\mathbf{w}^T \mathbf{w} - C)) = 0$$
$$\Leftrightarrow$$
$$\mathbf{\Phi}^T \mathbf{\Phi} \mathbf{w} - \mathbf{\Phi}^T \mathbf{t} + \beta \mathbf{w} = 0$$
$$\Leftrightarrow$$
$$(\mathbf{\Phi}^T \mathbf{\Phi} + \beta \mathbf{I}_K) \mathbf{w} = \mathbf{\Phi}^T \mathbf{t}$$
$$\Leftrightarrow$$
$$\mathbf{w} = (\mathbf{\Phi}^T \mathbf{\Phi} + \beta \mathbf{I}_K)^{-1} \mathbf{\Phi}^T \mathbf{t}$$

Since there is only one solution $\mathbf{w}^*$ is given by

$$\mathbf{w}^*(\beta) = (\mathbf{\Phi}^T \mathbf{\Phi} + \beta \mathbf{I}_K)^{-1} \mathbf{\Phi}^T \mathbf{t}$$

Note in the derivative above I took the convention that $\nabla_{\mathbf{w}}$ returns a column vector instead of a row vector. The derivation is the same if you resort to the row vector convention as you can always take the transpose on both sides of the equation.

f Write down all KKT equations (including the optimality condition).

4.0 maximum points · Open question

+1 point

Primal feasibility: $\|\mathbf{w}\|^2 - C \leq 0$ or $C - \|\mathbf{w}\|^2 \geq 0$.

+1 point

Dual feasibility: $\beta \geq 0$.

+1 point

Complimentary slackness: $\beta(C - \|\mathbf{w}\|^2) = 0$

+1 point

Optimality (see closed form solution for $\mathbf{w}^*(\beta)$):

$$\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi} + \beta\mathbf{I}_N)^{-1}\mathbf{\Phi}^T\mathbf{t}$$

g Assume you are given the minimizer $\mathbf{w}^*(\beta)$ of the Lagrangian L with respect to $\mathbf{w}$ for fixed β. Write down the dual optimizaton problem in terms of $\mathbf{w}^*(\beta)$.

2.0 maximum points · Open question

+1 point

One point for computing the dual Lagrangian.

The dual Lagrangian is given by

$$L^*(\beta) = L(\mathbf{w}^*(\beta), \beta).$$

If the dual is fully written out then this also gives full points. I.e., when

$$L^*(\beta) = \sum_{i=1}^N (\mathbf{w}^*(\beta)^T\phi(\mathbf{x}_i) - t_i)^2 - \beta(C - \|w^*(\beta)\|^2)$$

or with the different convention

$$L^*(\beta) = \sum_{i=1}^N (\mathbf{w}^*(\beta)^T\phi(\mathbf{x}_i) - t_i)^2 + \beta(\|w^*(\beta)\|^2 - C).$$

Note that

$$L^*(\beta) = \min_{\mathbf{w}} L(\mathbf{w}, \beta) = L(\mathbf{w}^*(\beta), \beta)$$

with

$$\mathbf{w}^*(\beta) = \arg\min_{\mathbf{w}} L(\mathbf{w}, \beta).$$

+1 point

One point for writing out the dual problem.

Then the dual optimization problem is given by

$$\max_{\beta} L^*(\beta) \quad \text{subject to} \quad \beta \geq 0.$$

h Is the dual problem convex?

1.0 maximum point · Multiple choice · 2 choices

Grading description

The dual formulation of convex problems are always concave.

○ yes	0.0
◉ no	1.0

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

i Why can the provided constrained minimization problem be thought of as a ridge regression problem?

2.0 maximum points · Open question

+1 point

Ridge regression is solving a linear regression problem with an added squared weight loss penalty. The solutions to ridge regression are of the form $\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi} + \beta\mathbf{I}_N)^{-1}\mathbf{\Phi}^T\mathbf{t}$.

+1 point

Solving the constraint optimization problem involves also solving for β. Once this is done the solution is given by $\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi} + \beta\mathbf{I}_N)^{-1}\mathbf{\Phi}^T\mathbf{t}$. This is the same as in ridge regression, but in ridge regression β is a hyperparameter and in our case it is determined by the constraint C.

j Let us apply the kernel trick by taking on the dual viewpoint. Using a matrix inversion lemma the solution for $\mathbf{w}^*(\beta)$ can be written as $\mathbf{w}^*(\beta) = \mathbf{\Phi}^T\mathbf{b}$, with $\mathbf{b}$ a dual variable defined by $\mathbf{b} = (\mathbf{K} + \beta\mathbf{I}_N)^{-1}\mathbf{t}$.

Here $\mathbf{K}$ would then be kernel in matrix form. How is it defined? Given an expression for $\mathbf{K}$ in terms of $\mathbf{\Phi}$.

1.0 maximum point · Open question

+1 point

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$

k Assume we have a test case $\mathbf{x}^*$ and you are given a kernel $K(\cdot,\cdot)$ that corresponds to a particular choice of $\boldsymbol{\phi}$. Provide an expression for the predicted value of t using the above ridge regression model. The expression may only involve kernel evaluations (instead of feature evaluations) since you don't know how to compute the features.

2.0 maximum points · Open question · Bonus

+0.5 points

A linear model is given by
$\mathbf{t}^* = \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}^*)$.

+0.5 points

Substituting the expression for $\mathbf{w}$ in terms of $\mathbf{b}$ (defined above) gives
$\mathbf{t}^* = \mathbf{b}^T\boldsymbol{\Phi}\boldsymbol{\phi}(\mathbf{x}^*)$

+1 point

Which explicitly written out gives (note $K(\mathbf{x}_i,\mathbf{x}^*) = \boldsymbol{\phi}(\mathbf{x}_i)^T\boldsymbol{\phi}(\mathbf{x}^*)$)
$\mathbf{t}^* = \sum_{i=1}^{N} b_i K(\mathbf{x}_i,\mathbf{x}^*)$

l Assume the kernel does not have any free parameters that you can tune. Now consider a situation where you suspect overfitting. What would you do to reduce this overfitting?

2.0 maximum points · Open question · Bonus

+2 points

Lower C which prevents particular weights becoming too large (it indirectly controls the β of ridge regression), or gather more data.

m Now consider the case of 2D data points, and you choose to work with a kernel that is defined by $k(\mathbf{y}, \mathbf{z}) = (\mathbf{y}^T \mathbf{A} \mathbf{z})^2$, with $\mathbf{A} = \begin{pmatrix} 3 & 0 \\ 0 & 7 \end{pmatrix}$, with $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^2$. What would the corresponding feature transform $\phi(\mathbf{x})$ be?

2.0 maximum points · Multiple choice · Bonus · 4 choices

○	$\phi(x) = (7x_1^2, \sqrt{3}\sqrt{7}x_1x_2, 3x_2^2)^T.$	0.0
○	$\phi(x) = (3x_1^2, 37x_1x_2, 7x_2^2)^T.$	0.0
○	$\phi(x) = (9x_1^2, 37x_1x_2, 49x_2^2)^T.$	0.0
◉	$\phi(x) = (3x_1^2, \sqrt{3}\sqrt{7}x_1x_2, 7x_2^2)^T.$	2.0

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

28 Image Content Classification

21.5 points • 8 questions

Consider a database of N images that contain scenes with objects (items, people, animals, ...) in them. For each image and for each object class we know whether or not it is present and we have annotated the images with a binary vector $\mathbf{t}_n \in \{0,1\}^K$ in which the k^{th} element is 1 if that object corresponding to that index is present and 0 otherwise. An example vector would be $\mathbf{t}_n = (0,1,0,0,1,0,\dots)^T \in \{0,1\}^K$.

We have access to a table of the K expected classes and their class indices. The first 5 classes are listed in the table below.

Class id	Class name
k=1	Adult
k=2	Child
k=3	Cat
k=4	Dog
k=5	Car
...	...

For each image we have access to feature vectors that are precomputed via deep neaural networks. The feature vector for the n^{th} image is denoted with $\mathbf{x}_n \in \mathbb{R}^M$.

All features vectors $\mathbf{x}_n$ are stored in a $N \times M$ matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$. All class indicator vectors $\mathbf{t}_n$ are stored in the $N \times K$ matrix $\mathbf{T} \in \{0, 1\}^{N \times K}$.

In this exercise we are interested in retrieving the image content (which object classes are present) for new input images via a probabilistic approach. We model the posterior class probabilities via

(1) $$p(C_k|\mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_K) = y_k(\mathbf{x}) = \frac{1}{1+e^{-a_k(\mathbf{x})}}$$

with $a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x}$, and for each k the weights $\mathbf{w}_k = (w_{k1}, \ldots, w_{kM})^T \in \mathbb{R}^M$ are considered to be model parameters.

Description

a We refer to $p(C_k|\mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_K)$ as the *posterior class* probability (as opposed to prior, likelihood, class conditional, naive, ...). Why do we call the above probability the *posterior class* probability?

2.0 maximum points · Open question

+1 point

It defines a probability for the class C_k

+1 point

After (posterior) observing input image $\mathbf{x}$.

b Suppose you have already optimized the model weights $\mathbf{w}_1, \ldots, \mathbf{w}_K$ and inspect the posterior class probabilities for a new input image $\mathbf{x}$. You find that $y_3 \approx 0.5$ and $y_4 \approx 0.5$ and all other posterior class probabilities are close to zero, i.e. $\forall_{k \neq 3, k \neq 4} : y_k \approx 0$. What does this tell you about the content of the image?

2.0 maximum points · Open question

Grading description

It means that the image probably contains a cat **and** a dog. Note, the model is not mutually exclusive like it would have been when y_k was given by the logistic loss function. So, an answer stating the image contains **either** a cat **or** a dog is not correct.

+1.5 points

It means that the image probably contains a cat **and** a dog. Possibly it contains only one cat or dog, whose appearance is equally close to both class types.

+0.5 points

Note, the model is not mutually exclusive like it would have been when y_k was given by the logistic loss function. So, an answer just stating the image contains **either** a cat **or** a dog is not correct.

+1 point

1 if only *or* is mentioned. Though this would imply a bad model because a good model would be able to see the difference between a cat and dog, and therefore would only assign a high prob to one of them. But yeah, maybe something really looks like a half cat-half dog.

For the weights we assume a prior distribution which is given by a generalized Gaussian as follows

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

where $\gamma > 0$ is a scale parameter, $q > 0$ determines the shape of the distribution, $\Gamma(\frac{1}{q})$ is some normalization constant, and the q-norm of a vector $\mathbf{w}_k$ (to the power q) is defined as

$$\|\mathbf{w}_k\|_q^q = \sum_{m=1}^{M} |w_{km}|^q.$$

Description

[Latex support]

Some of the math expressions used in this exercise are generated in latex code as follows:

- $p(C_k|\mathbf{x}, \mathbf{w}_1, \dots, \mathbf{w}_K) = y_k(\mathbf{x}) = \frac{1}{1+e^{-a_k(\mathbf{x})}}$

p(C_k| \mathbf{x}, \mathbf{w}_1, \dots, \mathbf{w}_K) = y_k(\mathbf{x}) = \frac{1}{1 + e^{-a_k(\mathbf{x})}}

- $p(\mathbf{w}_1, \dots, \mathbf{w}_K|\gamma, q) = \prod_{k=1}^K \left(\frac{q}{2\,\gamma\,\Gamma(\frac{1}{q})} \right)^M e^{-\lVert \mathbf{w}_k \rVert_q^q / \gamma^q}$

p(\mathbf{w}_1,\dots,\mathbf{w}_K | \gamma, q) = \prod_{k=1}^K \left(\frac{q}{2 \, \gamma\, \Gamma(\frac{1}{q})} \right)^M e^{-\lVert \mathbf{w}_k \rVert_q^q / \gamma^q}

- $\lVert \mathbf{w}_k \rVert_q^q = \sum_{m=1}^M |w_{km}|^q$

\lVert \mathbf{w}_k \rVert_q^q = \sum_{m=1}^{M} | w_{km}|^q

- Remember to place latex code in one line between dollar signs.

[/Latex support]

Description

c Consider two different weight vectors $\mathbf{w}_k$ and $\mathbf{w}_l$ ($k \neq l$).

1. Are they correlated according to the prior in Eq. (2)?
2. Are two different elements of the same weight vector $\mathbf{w}_k$, such as w_{k1} and w_{k2}, correlated?

For both cases, explain your answer.

4.0 maximum points · Open question

+1 point

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

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

(no points deducted if the formula for $p(\mathbf{w}_k|\gamma, q)$ is omitted)

+0.5 points

1. This means $\mathbf{w}_k$ and $\mathbf{w}_l$ for $k \neq l$ are independent variables.

+0.5 points

1. Independent variables are uncorrelated.

+1 point

2. The probabilities distribution for a single weight vector $\mathbf{w}_k$ itself can be factorized because the $\|\mathbf{w}_k\|_q^q$ is a sum of the individual elements in $\mathbf{w}_k$ (to the power q). The exponential of a sum can be factorized. Thus we can factorize as follows

$$\begin{aligned} p(\mathbf{w}_1|\gamma, q) &= \left(\frac{q}{2\,\gamma\,\Gamma(\frac{1}{q})}\right)^M e^{-(\sum_{m=1}^M |w_{km}|^q)/\gamma^q} \\ &= \prod_{m=1}^M \frac{q}{2\,\gamma\,\Gamma(\frac{1}{q})} e^{-|w_{km}|^q/\gamma^q} = \prod_{m=1}^M p(\mathbf{w}_{km}|\gamma, q). \end{aligned}$$

(again, no points deducted if no explicit formula is provided, as long as the answer contains a proper explanation in words.)

+0.5 points

2. This means all individual weights w_{km} are independently and (identically) distributed.

+0.5 points

2. and thus they are uncorrelated.

[Latex support]
Some of the math expressions used in this exercise are generated in latex code as follows:

- $p(\mathbf{w}_1, \dots, \mathbf{w}_K | \mathbf{X}, \mathbf{T}, \gamma, q)$

p(\mathbf{w}_1, \dots, \mathbf{w}_K | \mathbf{X}, \mathbf{T}, \gamma, q)

- $p(\mathbf{T} | \mathbf{X}, \mathbf{w}_1, \dots, \mathbf{w}_K)$

p(\mathbf{T} | \mathbf{X}, \mathbf{w}_1, \dots, \mathbf{w}_K)

- $p(\mathbf{w}_1, \dots, \mathbf{w}_K | \gamma, q)$

p(\mathbf{w}_1, \dots, \mathbf{w}_K | \gamma, q)
[/Latex support]
Description

d Write down the expression for the posterior $p(\mathbf{w}_1, \dots, \mathbf{w}_K | \mathbf{X}, \mathbf{T}, \gamma, q)$ in terms of the data likelihood $p(\mathbf{T} | \mathbf{X}, \mathbf{w}_1, \dots, \mathbf{w}_K)$ and the prior $p(\mathbf{w}_1, \dots, \mathbf{w}_K | \gamma, q)$. You do not need to explicitly write out the actual distributions.

2.0 maximum points · Open question

+1 point

The posterior distribution is obtained via Bayes rule

+1 point

As follows:

$$p(\mathbf{w}_1, \dots, \mathbf{w}_K | \mathbf{X}, \mathbf{T}, \gamma, q) = \frac{p(\mathbf{T} | \mathbf{X}, \mathbf{w}_1, \dots, \mathbf{w}_K) p(\mathbf{w}_1, \dots, \mathbf{w}_K | \gamma, q)}{p(\mathbf{T} | \mathbf{X}, \gamma, q)}.$$

e Give the expressions for

1. the log-likelihood $\ln p(t_n | \mathbf{x}_n, \mathbf{w}_1, \dots, \mathbf{w}_K)$ of a single training example ($\mathbf{x}_n$ with corresponding $\mathbf{t}_n$) given the probabilistic model of equation (1) **in terms of** y_k,
2. the log-likelihood for the full dataset $\ln p(\mathbf{T} | \mathbf{X}, \mathbf{w}_1, \dots, \mathbf{w}_K)$ **in terms of** y_k and
3. the log of the prior $\ln p(\mathbf{w}_1, \dots, \mathbf{w}_K | \gamma, q)$.

3.5 maximum points · Open question

+1 point

1.

The single datapoint likelihood is given by
$p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w}_1, \ldots, \mathbf{w}_K)$
$= \prod_{k=1}^K p(t_{nk}|\mathbf{x}_n, \mathbf{w}_1, \ldots, \mathbf{w}_K)$
$= \prod_{k=1}^K y_k(\mathbf{x}_n)^{t_{nk}}(1 - y_k(\mathbf{x}_n))^{1-t_{nk}}.$

+0.5 points

1.

and thus the log-likelihood is given by
$p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{k=1}^K t_{nk} \ln y_k(\mathbf{x}_n) + (1 - t_{nk}) \ln(1 - y_k(\mathbf{x}_n))$

+0.5 points

2.

The likelihood is given by
$$\prod_{n=1}^N p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w}_1, \ldots, \mathbf{w}_K)$$

+0.5 points

2.

and thus the log-likelihood as
$\sum_{n=1}^N \ln p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w}_1, \ldots, \mathbf{w}_K)$
which given the the previous answer is given by
$p(\mathbf{T}|\mathbf{X}, \mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^N \sum_{k=1}^K t_{nk} \ln y_k(\mathbf{x}_n) + (1 - t_{nk}) \ln(1 - y_k(\mathbf{x}_n))$

+1 point

The log of the prior is given by

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

(in the answer at least the front factor and the term that depends on $\mathbf{w}_k$ needs to be split for full points. The front factor does not necessarily need to be split in the sub-parts)

[Latex support]

- $L(\mathbf{w}_1, \dots, \mathbf{w}_K, \mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} \left[-t_{nk} \ln y_k(\mathbf{x}_n) - (1 - t_{nk}) \ln(1 - y_k(\mathbf{x}_n)) + \mu \|\mathbf{w}_k\|_q^q \right]$

L(\mathbf{w}_1, \dots, \mathbf{w}_K, \mu) = \sum_{n=1}^N \sum_{k=1}^K \left[\, - t_{nk} \ln y_k(\mathbf{x}_n) - (1 - t_{nk})\ln(1 - y_k(\mathbf{x}_n)) + \mu\lVert \mathbf{w}_k \rVert_q^q \, \right]

[/Latex support]

Description

f Show that the optimization problem corresponding to obtaining a Maximum A Posteriori (MAP) estimate for $\mathbf{w}_1, \dots, \mathbf{w}_K$ is equivalent to minimizing the corresponding loss function

$$L(\mathbf{w}_1, \dots, \mathbf{w}_K, \mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} \left[-t_{nk} \ln y_k(\mathbf{x}_n) - (1 - t_{nk}) \ln(1 - y_k(\mathbf{x}_n)) + \mu \|\mathbf{w}_k\|_q^q \right]$$

with respect to $\mathbf{w}_1, \dots, \mathbf{w}_K$, and with μ a regularization penalty parameter. Also explain how μ is related to γ and q?

3.0 maximum points · Open question

+0.5 points

Optimizing the posterior equals optimizing the log-posterior.

+1 point

Which is equivalent to maximizing (log-likelihood + log-prior):
$\sum_{n=1}^{N} \sum_{k=1}^{K} \left[t_{nk} \ln y_k(\mathbf{x}_n) + (1 - t_{nk}) \ln(1 - y_k(\mathbf{x}_n)) - \frac{1}{\gamma^q} \|\mathbf{w}_k\|_q^q \right]$,
where we omitted the terms that do not depend on the weights $\mathbf{w}_k$.

+1 point

Maximizing some criterion is the same as *minimizing the negative of this criterion*. And thus the expressions are the same for $\mu = \frac{1}{\gamma^q}$.

+0.5 points

Parameter q is not related to μ.

g After reinspecting the dataset we discover that, quite unexpectedly, and perhaps disturbingly, many of the images feature clowns. We decide to go over all images again and add the "clown" class to all the target vectors $\mathbf{t}_n$, such that we can retrain our model and make it capable of detecting the presence of clowns in images. The new target vectors $\mathbf{t}_n \in \{0, 1\}^{K+1}$ are now thus of dimension $K + 1$. Do we need to retrain the entire model? Explain your answer.

3.0 maximum points · Open question

Grading description

Also full points if answered in words along the line:

The functions y_k are logistic sigmoid functions that only depend on $\mathbf{w}_k$. The problem is actually a problem of fitting $K + 1$ logistic regression classifiers independently of one another. The class conditional distributions can therefore be trained independently from one another.

+1 point

Since each y_k only depends on $\mathbf{w}_k$ ...

+1 point

... the MAP objective, or the regularized logistic regression loss can be split in to parts that only depend on that particular $\mathbf{w}_k$. I.e.,
$L(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{k=1}^{K+1} L_k(\mathbf{w}_k)$
with
$L_k(\mathbf{w}_k) = \sum_{n=1}^{N} t_{nk} \ln y_k(\mathbf{x}_n) + (1 - t_{nk}) \ln(1 - y_k(\mathbf{x}_n)) - \frac{1}{\alpha} \|\mathbf{w}_k\|_q^q.$

+1 point

Thus optimizing $L(\mathbf{w}_1, \ldots, \mathbf{w}_{K+1})$ with respect to weights $\mathbf{w}_k$ does not depend on the other weights, but only the particular loss $L_k(\mathbf{w}_k)$ needs to be optimized.

+1 point

One point if just the answer is correct, but the reasoning is missing or incorrect.

h Let us finally take a look again at the prior on the weights $\mathbf{w}_k$ as given in equation (2), and see what it looks like for one particular weight vector $\mathbf{w}$ in the 1D and 2D case, for which $\mathbf{w} \in \mathbb{R}$ or $\mathbf{w} \in \mathbb{R}^2$ respectively. In 1D, the generalized Gaussian is given by

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

In the figure below it is plotted for the 1D case (left) and the 2D case (right). The left plot directly shows the prior probabilities for some value w given the model parameters $\gamma = 1$ and several values for q. In the right figure you see the contour plot for the 2D distributions, showing the level set $|w_1|^q + |w_2|^q = C$ for some arbitrary value C.

Imagine that you train several models with MAP estimates for $\mathbf{w}_1, \ldots, \mathbf{w}_K$ for different values of $q > 0$ (and $\gamma = 1$).

1. Which trained model will have sparse weight vectors, the ones with $q > 1$ or those with $q \leq 1$? Explain your answer.
2. Given the correct choice for q, how should the γ parameter be changed in order to increase the sparsity of the solutions for the weights $\mathbf{w}_k$?

(Remember, the more elements in a vector take on the value 0, the sparser we consider it to be).

2.0 maximum points · Open question

+0.5 points

The distributions $q \leq 1$ are more peaked around the $w = 0$. This means that weights $w = 0$ are a-priori more likely to be sampled. This idea generalizes to 2D as we see in the right figure, the mass of the distribution concentrates around the axes. We assume this generalizes to higher dimensions as well.

+0.5 points

Thus, the models with $q \leq 1$ are a-priori more likely to be sparse than the other models because they are more likely to sample vectors of which the components $w_m = 0$.

+1 point

This prior assumption about the weights is propagated to the MAP estimates via the γ parameter by making it smaller.

Equivalently we can approach this via the loss function of exercise (e) where the weight penalty λ should be set higher to promote sparsity. In that exercise we found that $\mu = \frac{1}{\gamma}$.