Final Exam

Machine Learning 1 52041MAL6Y 21/22 (Period 1) · 14 exercises · 50.0 points

1 ML vs MAP

1.0 point · 1 question

Given a dataset $\mathcal{D} = \{x_n\}_{n=1}^{N}$. The data is normally distributed $\mathcal{N}(x_n|\mu, \sigma^2)$ and we assume a Gaussian prior over $\mu : \mathcal{N}(\mu|0, \sigma_0^2)$. Furthermore, the variance σ^2 is assumed to be known. Let μ_{ML} and μ_{MAP} respectively be the ML and MAP estimates for μ. How does $|\mu_{ML} - \mu_{MAP}|$ change as (i) $\sigma_0 \to 0$, (ii) $\sigma_0 \to \infty$, (iii) $N \to \infty$.

1.0 point · Multiple choice · 4 choices

○ (i) decrease (ii) increase (iii) decrease. 0.0

○ (i) decrease (ii) increase (iii) increase. 0.0

● **(i) increase (ii) decrease (iii) decrease.** 1.0

○ (i) increase (ii) decrease (iii) increase. 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

2 Changing decision boundaries

1.0 point · 1 question

Consider a binary classification problem. Suppose I have trained a model on a linearly separable training set, and now I get a new labeled data point which is correctly classified by the model, and far away from the decision boundary. If I now add this new point to my earlier training set and re-train, in which cases is the learned decision boundary likely to change?

1.0 point · Multiple choice · 4 choices

☑	When my model is a perceptron.	0.34
☑	When my model is logistic regression.	0.34
□	When my model is an SVM.	-0.5
☑	When my model is Gaussian discriminant analysis.	0.34

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

3 Misclassification error

1.0 point · 1 question

After fitting a Logistic Regression model with two classes to the training data we calculate the confusion matrix of the model on the test data with N observations: $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$. Which of the following formulas could you use to estimate the misclassification error:

1.0 point · Multiple choice · 4 choices

☐ $\frac{1}{N}(A+D)$. -0.5

☑ $1-\frac{1}{N}(A+D)$. 0.5

☑ $\frac{1}{N}(B+C)$. 0.5

☐ $1-\frac{1}{N}(B+C)$. -0.5

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

4 Probabilistic models

1.0 point · 1 question

In classification, there are three approaches, discriminant functions, probabilistic generative models and probabilistic discriminative models. The following are statements about probabilistic generative models and probabilistic discriminative models. Which of the following statements is true?

1.0 point · Multiple choice · 4 choices

☑	**Logistic regression is a probabilistic discriminative model.**	0.34
☑	**In probabilistic discriminative models, the posterior class probabilities $p(C\|\mathbf{x})$ are modeled directly.**	0.34
□	In probabilistic discriminative models, the prior probability of class $p(C)$ is modeled.	-0.5
☑	**In generative models, the class conditional probability $p(\mathbf{x}\|C)$ are modeled directly.**	0.34

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

5 Neural networks: representation power

1.0 point · 1 question

Consider a neural network with L layers, and H hidden units per hidden layer, and O output units.

1.0 point · Multiple choice · 4 choices

☑ Given a fixed number of network paramers (say fix the number $L\,H$) and activation functions $h(x) = \max(0, x)$ for the hidden units, one most effectively increases the network complexity by increasing L rather than H. 0.34

☑ A 5 layer neural network with activation functions $h(x) = \frac{1}{1+e^{-x}}$ for the hidden layers is able to approximate any continuous function with compact support to arbitrary precision via suitable choice for H. 0.34

□ A 5 layer neural network with activation functions $h(x) = x^3$ for the hidden layers is able to approximate any continuous function with compact support to arbitrary precision via suitable choice for H. -0.5

☑ With the same amount of output units O, a linear regression model is at least as expressive as a deep neural network with $L = 100$ layers, any choice of of H, and activation functions $h(x) = x$. 0.34

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

6 The kernel trick

1.0 point · 1 question

What is the advantage of using kernels in methods such as support vector machines?

1.0 point · Multiple choice · 4 choices

☑	They can simulate an infinite dimensional feature space.	0.5
□	They will always reduce the number of support vectors.	-0.5
□	They reduce the risk of getting stuck in local minima.	-0.5
☑	They make it possible to model non-linear decision boundaries.	0.5

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

7 SVM and underfitting

1.0 point · 1 question

Suppose you are given the following binary dataset and trained a SVM that solves

$$\underset{\mathbf{w},\mathbf{b},\{\xi_n\}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \quad \text{subject to} \quad \begin{array}{lll} \forall_{n=1,...,N}: & t_n y_n & \geq 1-\xi_n \\ \forall_{n=1,...,N}: & \xi_n & \geq 0 \end{array}$$

using a Gaussian kernel $k(\mathbf{x},\mathbf{x}) = \exp(-\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{x}'\|^2)$ that gave the following decision boundary:

You suspect that the SVM is underfitting your dataset. Should you try to increase or to decrease the C parameter? Increase or decrease σ^2?

1.0 point · Multiple choice · 4 choices

● Increase C, decrease σ^2 1.0

○ Increase C, increase σ^2 0.0

○ Decrease C, increase σ^2 0.0

○ Decrease C, decrease σ^2 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

8 Local vs global optimality

1.0 point · 1 question

In which of the following can we expect to find the globally optimal solution for the model parameters? In all cases assume i.i.d. assumption on the dataset $D = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$ and assume all (undefined) parameters/functions to be known.

1.0 point · Multiple choice · 5 choices

☑ ○ $\underset{\mathbf{w},\mathbf{b},\{\xi_n\}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \quad \text{subject to} \quad \begin{matrix} \forall_{n=1,\ldots,N}: & t_n y_n & \geq 1 - \xi_n \\ \forall_{n=1,\ldots,N}: & \xi_n & \geq 0 \end{matrix}$ (

with $y_n = \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) + \mathbf{b}$ and with $\boldsymbol{\phi}(\mathbf{x})$ some non-linear feature transformation of input $\mathbf{x}$.

☑ ○ $\underset{\mathbf{w},\mathbf{b},\beta}{\text{maximize}} \ p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)}$ (

using Gaussian predictive distribution $p(t|\mathbf{x}, \mathbf{w}) = \mathcal{N}(t|\mathbf{w}^T\mathbf{x} + \mathbf{b}, \beta^{-1})$ and prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \mathbf{S})$.

□ ○ $\underset{\{\pi_k\},\{\boldsymbol{\mu}_k\},\sigma}{\text{maximize}} \ p(D|\mathbf{w}, \{\pi_k\}, \{\mu_k\}, \sigma)$ -

with $p(\mathbf{x}) = \sum_{k=1}^{5}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma^2\mathbf{I})\pi_k$, with $\sum_{l=1}^{5}\pi_l = 1$.

□ ○ $\underset{\mathbf{w}}{\text{minimize}} \ \sum_{n=1}^{N}(NN_{\mathbf{w}}(\mathbf{x}_n) - t_n)^2$ using *Gradient Descent* -

with $NN_{\mathbf{w}}$ a deep neural network with ReLU activations and parametrized by weights $\mathbf{w}$.

□ ○ $\underset{\mathbf{w}}{\text{minimize}} \ \sum_{n=1}^{N}(NN_{\mathbf{w}}(\mathbf{x}_n) - t_n)^2$ using *Stochastic Gradient Descent* -

with $NN_{\mathbf{w}}$ a deep neural network with ReLU activations and parametrized by weights $\mathbf{w}$.

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

9 Gaussian process and its kernel

1.0 point · 1 question

The values $f(\mathbf{x}_i)$ of a Gaussian process f evaluated at a fixed set of N points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, with each $\mathbf{x}_i \in \mathbb{R}^D$ can be described by a multi-variate Gaussian. Which of the following statements are true?

1.0 point · Multiple choice · 4 choices

☑	In order to compute the covariance matrix of the multi-variate Gaussian one needs to evaluate the kernel for all data point pairs.	0.5
□	The kernel makes its appearance in both the covariance matrix as well as the mean of the multi-variate Gaussian.	-0.5
☑	The kernel describes how two function values $f(\mathbf{x})$ and $f(\mathbf{y})$ covariate for any choice of $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$.	0.5
□	The covariance matrix of the multi-variate Gaussian is a $D \times D$ matrix.	-0.5

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

10 SVM: decision boundary

1.0 point · 1 question

Consider the following two-class SVM optimization problem with data set $\mathcal{D} = \{(\mathbf{x}_n, t_n)\}_{n=1}^N$ with $\mathbf{x}_n \in \mathbb{R}^2$ and $t_n \in \{-1, 1\}$:

$$\underset{\mathbf{w},\mathbf{b},\{\xi_n\}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \quad \text{subject to} \quad \begin{array}{lrl} \forall_{n=1,...,N}: & t_n(\mathbf{w}^T\mathbf{x}_n - \mathbf{b}) & \geq 1 - \xi_n \\ \forall_{n=1,...,N}: & \xi_n & \geq 0 \end{array}$$

The data set is depicted in the figure below. The 11 gray dots correspond to data points with labels $t_n = +1$, and the 9 black stars to data points with $t_n = -1$.

Instead of solving the above problem directly you consider solving it using the kernel trick and try out 2 kernels: namely $k_A(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$ and $k_B(\mathbf{x}, \mathbf{x}') = \exp(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}'\|^2)$ with some choice for σ.

Which of the following statements are correct?

1.0 point · Multiple choice · 4 choices

□ The solid line (decision bounday I) could represent the learned decision boundary using k_A and with the setting $C \to \infty$. -0.5

☑ **The solid line (decision bounday I) could represent the learned decision boundary for kernel k_A and with the setting $C = 1$.** 0.34

☑ **The dashed line (decision bounday II) could represent the learned decision boundary for kernel k_B and with the setting $C \to \infty$.** 0.34

☑ **The dashed line (decision bounday II) could represent the learned decision boundary for kernel k_B and with the setting $C = 1$.** 0.34

Feedback

Feedback if the question is completely correct

Feedback if the question is not completely correct

Feedback if the question is completely incorrect

11 ML1 student on a date (Probability theory)

5.5 points · 4 questions

A friend set you up for date with someone called Alex. You are afraid of getting in a sequence of dates going nowhere, so you decide to adopt a Bayesian approach to dating; you will setup a probabilistic model to help you decide whether or not a second date would be worthwhile.

You heard that successful couples have a shared taste in music, so you focus on this feature. You pick your favorite album and assume that, if Alex were indeed to be your soulmate, then there's an 80% chance of you both liking the album. You also assume that you have a very sophisticated taste and that only 1 in 1.000 people like this album. So you reason, there is still a 1/1.000 chance that Alex likes the album even if you don't turn out to be soulmates. Finally, you guess that the chance for finding a match is 1% in the first place.

You stick to the following conventions:

- Define M to be the binary random variable of finding a match ($M = 1$) with the date.
- Let A be the random variable for someone liking the album ($A = 1$) or not ($A = 0$).
- You decide for a second date if the probability of Alex being a match is larger than 25%.
- Your definition of an *enjoyable evening* is one where you can talk all night with a likeminded person about your favorite album.

Description

a What is the probability of you having an *enjoyable evening* with Alex?

Give your answer in % in one decimal precision (xx.x%) or provide it as a fraction.

1.5 points · Open question · 1/4Page

+0.5 points

Give formula or mention that the probability $p(A)$ can be computed via the sum rule of probabilities, or via the marginalization over the joint distribution $p(A, M)$.

+1 point

Write out the marginalization:

$$p(A=1) = p(A=1|M=1) * p(M=1) + p(A=1|M=0)p(M=0) = 0.8 * 0.01 + 0.001 * 0.99 = 0.00899$$

+0.5 points

Students who did the above get .5 bonus point for doing the computations dispite the ambiguity in the text.

Give the answer in the asked format: In term of percentages you'll have a chance of **0.9%** for having an enjoyable evening. Answering a fraction, such as **899/100,000** is also fine.

+1.5 points

Due to ambiguity in the assignment we also accept directly providing p(A) = 1/1000 .

b So you go out with Alex and things are going smoothly. You just discovered that Alex really likes your favorite album too! 😍 Wow, did you find your soulmate? Compute the probability for a match!

Give your answer in % in one decimal precision (xx.x%) or provide it as a fraction.

2.0 points · Open question · 1/4Page

+1 point

Give formula or mention that the posterior probability is obtained via Bayes rule. Which is given by

$$p(M=1|A=1)=\frac{p(A=1|M=1)p(M=1)}{p(A=1)}$$

+0.75 points

Fill this in:

$$p(M=1|A=1)=\frac{0.8*0.01}{0.00899}=0.88988$$

+0.25 points

Provide answer as a percentage or fraction. So $p(M=1|A=1)=89.0\%$. (or 88.9% if compute with rounded result of p(A)=0.009). Otherwise, either $\frac{800}{899}$ or $\frac{800}{900}$.

c **[BONUS]** Alex tells you that the album is in all the "all-time-favorite" album charts and it is actually not that special that you both like it; around 1 in 50 people like it. Your heart sinks... 😱 This greatly reduces the probability of being soulmates. Recompute it by updating your initial assumption for the probability for $A=1$ for non-soulmates ($M=0$) with the new fact. Will you go for a second date?

1.0 point · Multiple choice · Bonus · 2 choices

Grading description

Recomputing with P(A=1|M=0)=1/50 will change the posterior to 28.8%

● Yes, few it's just above 25%! 1.0

○ No way, not with less than 25% chance of Alex being your soulmate! 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 d "Wait! Are you taking notes on a date!?!" Alex asks. Whoops, awkward.... 😅 You explain you were computing the posterior for having just found your soulmate". "You did what?!?", Alex says, "You should have told me, I'm quite the Bayesian myself you know! With friends were building such a soulmate prediction model based on our collective experiences!"

Alex explains their model is parameterized by weights $\mathbf{w}$ and predicts the probability of a match M given some experience vector $\mathbf{x}$. They decided on a prior for $\mathbf{w}$ and update their choice for $\mathbf{w}$ given a collected dataset $D_N = \{\mathbf{x}_i\}_{i=1}^N$ of N experiences. They recompute the most probable model parameters, given the data, every time a new experience $\mathbf{x}_{N+1}$ is added (creating D_{N+1}). "The dataset is getting too large and it is increasingly demanding to compute the most probable $\mathbf{w}$. Super annoying..." Alex says.

OK, you got this! Propose to Alex... a solution to this problem!

2.0 points · Open question · 3/5Page

+1 point

A solution is obtained via sqential Bayesian learning

+1 point

Mention either this: In this case the posterior after observing data point $\mathbf{x}_n$ is given by

$$p(\mathbf{w}|D_{N+1}) = \frac{p(\mathbf{x}_{N+1}|\mathbf{w})p(\mathbf{w}|D_N)}{p(x_{N+1})}$$

+1 point

Or this: Where the posterior $p(\mathbf{w}|D_N)$ after N observations is used as prior for the posterior after observing $\mathbf{x}_{N+1}$.

-0.25 points

If forgetting the evidence in the equation for the posterior. (or when writing = instead of $\propto$ when not dividing by the evidence)

+0.5 points

Some points for alternative answer based on SGD. SGD is an efficiency improvement in finding the optimum, for large datasets at least. But it doesn't solve the problem of having to retrain every time a new data point comes in.

12 Discrete latent variable modeling

14.0 points · 9 questions

We have access to a dataset $D = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of N observations which we assume to be independently and identically distributed and generated by a process that depends on a discrete latent variable $z \in \{C_1, \dots, C_K\}$. I.e., we assume $\mathbf{x} \sim p(\mathbf{x}|z)$, K discrete latent classes, and that $z \sim p(z)$. We will denote with $\mathbf{z}$ the one-hot encoding of z. I.e., $\mathbf{z} = (z_1, \dots, z_k)^T \in \{0, 1\}^K$ where $z_k = 1$ if $z = C_k$ and 0 otherwise.

We thus believe that each datapoint $\mathbf{x}_i$ is associated with a latent $\mathbf{z}$, however, there is no way of observing it. So, instead we try to infer the latent variables using a generative model. We will model the latent conditionals with a normal distribution with parameters σ and $\boldsymbol{\mu}_k$ for each class via

$$p(\mathbf{x}|z = C_k, \boldsymbol{\mu}_k, \sigma) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{I}\,\sigma^2).$$

We will model the prior on the latent with a Bernoulli distribution

$$p(\mathbf{z}|\{\pi_k\}) = \prod_{k=1}^{K} \pi_k^{z_k},$$

with model parameters $\{\pi_k\}_{k=1}^K$ that satisfy $\sum_{k=1}^K \pi_k = 1$. We want to derive the optimal model parameters $\{\pi_k\}$, $\{\boldsymbol{\mu}_k\}$ and σ, and intend to do so via the Expectation Maximization (EM) algorithm.

Description

a Suppose we have fully optimized the model and found the optimal parameters $\{\pi_k\}$, $\{\boldsymbol{\mu}_k\}$ and σ. How could we generate/sample new data points with this model?

1.0 point · Open question · 1/4Page

+0.5 points

First sample z from the prior $p(z)$.

+0.5 points

Then sample $\mathbf{x}$ from the corresponding latent conditional $p(\mathbf{x}|z)$

+0.25 points

If describing marginalization without explaining *how* to sample from it.

b Give an expression for the marginal $p(\mathbf{x}|\{\pi_k\},\{\boldsymbol{\mu}_k\},\sigma)$ *and* derive the likelihood $p(D|\{\pi_k\},\{\boldsymbol{\mu}_k\},\sigma)$ for D being generated by our model. Please mention if you relied on any assumptions.

2.5 points · Open question · 1/4Page

+1 point

The marginal:
$p(\mathbf{x}|\{\pi_k\},\{\boldsymbol{\mu}_k\},\sigma) = \sum_{k=1}^{K} p(\mathbf{x}|z=C_k)p(z=C_k)$
$= \sum_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k,\sigma^2\mathbf{I})\pi_k$

+1 point

The likelihood

$$p(D|\{\pi_k\},\{\boldsymbol{\mu}_k\},\sigma) = \prod_{n=1}^{N}\sum_{k=1}^{K}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k,\sigma^2\mathbf{I})\pi_k$$

+0.5 points

Where we used the i.i.d. assumption

-0.5 points

Mistake for $p(z=C_k)$

c Write down the expression for the posterior latent class probability $p(z = C_k|\mathbf{x}, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \sigma)$.

2.0 points · Open question · 1/5Page

+1.5 points

Using Bayes rule we get:

$$p(z = C_k|\mathbf{x}, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \sigma) = \frac{p(\mathbf{x}|z = C_k)p(z = C_k)}{p(\mathbf{x}|\{\pi_k\}, \{\boldsymbol{\mu}_k\}, \sigma)}$$

[This point is also given if the explicit expression below is correct.]

+0.5 points

Explicitly fill in the distributions

$$p(z = C_k|\mathbf{x}, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \sigma) = \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma)\pi_k}{\sum_{l=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l, \sigma)\pi_l}$$

-0.25 points

If the sum uses the same index k. This cannot be the case as k is already reserved for the queried class and is therefore kept fixed.

d Let us refer to the posteriors as responsibilities, denoted with $r_{nk} = p(z = C_k|\mathbf{x}_n, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \sigma)$. We then maximize the log-likelihood with respect to parameters $\{\pi_k\}$, $\{\boldsymbol{\mu}_k\}$, and σ, while keeping the responsibilities r_{nk} fixed. For example, the update rule for the σ in terms of r_{nk} then becomes

$$\sigma^2 = \frac{1}{N}\frac{1}{d}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x} - \boldsymbol{\mu}_k\|^2.$$

Derive the update rule for $\boldsymbol{\mu}_k$ in terms of r_{nk} and explain in words what this update rule does.

Hint: Make use of the fact that $\frac{\partial}{\partial \mu_k}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma^2) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma^2)\frac{1}{\sigma^2}(\mathbf{x} - \boldsymbol{\mu}_k)^T$.

2.5 points · Open question · 3/5Page

+0.5 points

Our objective is to solve for $\boldsymbol{\mu}_k$ in the following

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} \ln p(D) = 0$$

(Points if it is implicitly clear the student used this)

+0.5 points

Correctly compute the dertive of the log-likelihood:

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} \ln p(D) = \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\mu}_k} \ln \sum_{l=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l, \sigma^2\mathbf{I})\pi_l$$

$$= \sum_{n=1}^{N} \frac{1}{\sum_{l=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l,\sigma^2\mathbf{I})\pi_l} \frac{\partial}{\partial \boldsymbol{\mu}_k} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma^2\mathbf{I})\pi_k$$

$$= \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k,\sigma^2\mathbf{I})\pi_k}{\sum_{l=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l,\sigma^2\mathbf{I})\pi_l} \frac{1}{\sigma^2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

+0.5 points

Recognize the posterior distribution and replace it with r_{nk}

$$= \sum_{n=1}^{N} r_{nk} \frac{1}{\sigma^2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

+0.5 points

Set derivative equal to zero (or $\mathbf{0}^T$) and solve for μ_k, no points deducted for not sticking to correct row vector convention.

$$= \sum_{n=1}^{N} r_{nk} \frac{1}{\sigma^2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T = 0$$

$$\overset{\sigma>0}{\Rightarrow} \sum_{n=1}^{N} r_{nk} \mathbf{x}_n = \sum_{n=1}^{N} r_{nk} \boldsymbol{\mu}_k$$

$$\Leftrightarrow \sum_{n=1}^{N} r_{nk} \mathbf{x}_n = N_k \boldsymbol{\mu}_k$$

$$\Leftrightarrow \boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \mathbf{x}_n$$

with $N_k = \sum_{n=1}^{N} r_{nk}$.

+0.5 points

The update rule thus recomputes the means $\boldsymbol{\mu}_k$ as a weighted average of the points for that class C_k. The points are weighted with the current posteriors/responsibilities r_{nk}

e Derive the update rule for π_k in terms of r_{nk} and explain in words what this update rule does.

3.0 points · Open question · 13/20Page

+0.5 points

We have to solve the constrained optimization problem, of maximizing the log-likelihood whilst respecting the costraint that $\sum_{l=1}^{K} \pi_l = 1$. Thus, we want to find the optimizer of the Lagrangian

$$L(\{\pi_k\}, \lambda) = \ln p(D) + \lambda \left(\sum_{l=1}^{K} \pi_l - 1 \right)$$

with $\lambda \geq 0$ a Lagrange multiplier.

Or write something like finding the stationary points (that includes the Lagrange multiplier term)

$$\frac{\partial}{\partial \pi_k} \left(\ln p(D) + \lambda \left(\sum_{l=1}^{K} \pi_l - 1 \right) \right) = 0,$$

+0.5 points

Correctly compute the derivative w.r.t. π_k:
$\frac{\partial}{\partial \pi_k} \ln p(D) = \sum_{n=1}^{N} \frac{\partial}{\partial \pi_k} \ln \sum_{l=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l, \sigma^2 \mathbf{I})\pi_l + \lambda$

$= \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma^2 \mathbf{I})}{\sum_{l=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l, \sigma^2 \mathbf{I})\pi_l} + \lambda$

+0.5 points

Write in terms of posterior r_{nk}:

$= \sum_{n=1}^{N} \frac{1}{\pi_k} r_{nk} + \lambda$

+0.5 points

Set to zero and solve for π_k:

$\pi_k = -\frac{1}{\lambda} \sum_{n=1}^{N} r_{nk} = -\frac{N_k}{\lambda}$,
with $N_k = \sum_{n=1}^{N} r_{nk}$.

Thus the update rule, that still depends on λ is

$$\pi_k = -\frac{N_k}{\lambda}$$

+0.5 points

Optimize Lagrangian w.r.t. λ:

$\sum_{l=1}^{K} \pi_l = 1$

(plug in the solution for π_k, make N_k explicit)

$\Leftrightarrow \sum_{n=1}^{N} \sum_{l=1}^{K} -\frac{r_{nl}}{\lambda} = 1$

(note that $\sum_{l=1}^{K} r_{nl} = 1$ - probs sum to 1)

$\Leftrightarrow \sum_{n=1}^{N} 1 = -\lambda$

$\Leftrightarrow \lambda = -N$

+0.5 points

Thus, the update rule is

$$\pi_k = \frac{N_k}{N}$$

and it recomputes the (weighted) fraction of points belonging to class C_k, given the current parameters.

f Describe the EM algorithm and how the update rules are used in it.

1.0 point · Open question · New page · 1/2Page

+0.34 points

The EM algorithm iteratively updates the model parameters by alternatiting between an expectation (E) and maximization (M) step.

+0.34 points

In the E step the expected posterior/responsibilities are update based on the current model parameters

+0.34 points

In the M-step the log-likelihood is maximized whilst keeping r_{nk} fixed. The derived update rules are used in this step.

g Will the EM algorithm converge to a globally optimal solution?

1.0 point · Multiple choice · 2 choices

○ 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

h The K-means clustering algorithm shows strong resemblance to the EM algorithm. In fact, we can turn the above EM algorithm into K-means by directly modeling the posterior probabilities r_{nk} with a function $q(z|\mathbf{x})$, instead of modeling it with $p(z|\mathbf{x}, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \sigma)$ using using $p(\mathbf{x}|z = C_k, \boldsymbol{\mu}_k, \sigma)$ and $p(\mathbf{z}, \{\pi_k\})$. We will refer to this function as the encoder. How should you define $q(z|\mathbf{x})$ to obtain K-means?

Write down the expression for the encoder (i.e. write $q(z|\mathbf{x}) = \ldots$) or describe in words what this function does. Indicate on which of the parameters $\{\pi_k\}$, $\{\boldsymbol{\mu}_k\}$ and/or σ it depends.

1.0 point · Open question · 1/4Page

+0.75 points

$$q(z = C_k|\mathbf{x}) = \begin{cases} 1 & \text{if} \;\; k = \arg\min_l \|\mathbf{x} - \boldsymbol{\mu}_l\| \\ 0 & \text{otherwise} \end{cases} .$$

+0.25 points

So it depends on the model parameters $\{\boldsymbol{\mu}_l\}$

i **[BONUS]** K-means clustering suffers from the fact that the cluster sizes are equal for each latent class. Can you think of an adaptation based on the above that allows to handle clusters with different sizes? What changes to the encoder and/or variable update rules should be made?

1.5 points · Open question · Bonus · 1/2Page

Grading description

Use different σ_k per cluster.

+0.75 points

The update rule for σ_k then becomes

$$\sigma_k^2 = \frac{1}{N_k}\frac{1}{d}\sum_{n=1}^{N} r_{nk}\|\mathbf{x} - \boldsymbol{\mu}_k\|^2.$$

+0.75 points

The encoder then changes to

$$q(z|\mathbf{x}) = \begin{cases} 1 & \text{if } \; k = \arg\min_l \frac{1}{\sigma_k}\|\mathbf{x} - \boldsymbol{\mu}_l\| \\ 0 & \text{otherwise} \end{cases},$$

so the distance is scaled with a factor $\frac{1}{\sigma_k^2}$ when computing the nearest cluster center.

+0.75 points

For alternative answer based on Gaussian Mixture models

13 Continuous latent variable modeling

5.5 points · 5 questions

We now consider a dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of observations $\mathbf{x}_n \in \mathbb{R}^d$ which you assume come from a multi-variate normal distribution $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{S})$. You assume that the observations are deterministically associated with a continuous latent variable $\mathbf{z} \in \mathbb{R}^M$ via the model

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}.$$

Description

a Through the above model, the latent $\mathbf{z}$ is also a random variable. What is the expectation and covariance of $\mathbf{z}$? Compute $\mathbb{E}[\mathbf{z}]$ and $\mathrm{Cov}[\mathbf{z}]$.

2.5 points · Open question · 3/4Page

+0.75 points

$\mathbb{E}[\mathbf{z}] = \mathbb{E}[\mathbf{W}\mathbf{x} + \mathbf{b}]$
$= \mathbf{W}\mathbb{E}[\mathbf{x}] + \mathbf{b}$

+0.25 points

Finally, since $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$,
$\mathbb{E}[\mathbf{z}] = \mathbf{W}\boldsymbol{\mu} + \mathbf{b}$

+0.5 points

$\mathrm{Cov}[\mathbf{z}] = \mathbb{E}[(\mathbf{z} - \mathbb{E}[\mathbf{z}])(\mathbf{z} - \mathbb{E}[\mathbf{z}])^T]$
$= \mathbb{E}[(\mathbf{W}\mathbf{x} + \mathbf{b} - \mathbf{W}\boldsymbol{\mu} - \mathbf{b})(\mathbf{W}\mathbf{x} + \mathbf{b} - \mathbf{W}\boldsymbol{\mu} - \mathbf{b})^T]$
$= \mathbb{E}[(\mathbf{W}\mathbf{x} - \mathbf{W}\boldsymbol{\mu})(\mathbf{W}\mathbf{x} - \mathbf{W}\boldsymbol{\mu})^T]$
$= \mathbf{W}(\mathbb{E}[\mathbf{x}\mathbf{x}^T] - \mathbb{E}[\mathbf{x}]\boldsymbol{\mu}^T - \boldsymbol{\mu}\mathbb{E}[\mathbf{x}]^T + \boldsymbol{\mu}\boldsymbol{\mu}^T)\mathbf{W}^T$
$= \mathbf{W}(\mathbb{E}[\mathbf{x}\mathbf{x}^T] - 2\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T)\mathbf{W}^T$
$= \mathbf{W}(\mathbb{E}[\mathbf{x}\mathbf{x}^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T)\mathbf{W}^T$

+0.5 points

Next, we note that $\mathbb{E}[\mathbf{x}\mathbf{x}^T]$ can be derived from the covariance of $\mathbf{x}$ via
$\mathrm{Cov}[\mathbf{x}] = \mathbf{S} \Leftrightarrow$
$\mathbb{E}[\mathbf{x}\mathbf{x}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T = \mathbf{S} \Leftrightarrow$
$\mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{S} + \boldsymbol{\mu}\boldsymbol{\mu}^T.$

+0.5 points

Substituting this in the expression for the covariance of $\mathbf{z}$, we obtain
$\mathrm{Cov}[\mathbf{z}] = \mathbf{W}\mathbf{S}\mathbf{W}^T$

+1.5 points

Alternative solution:

$$\begin{aligned}\mathbf{S} = \mathrm{Cov}[\mathbf{x}] &= \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^T] \\ &= \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T]\end{aligned}$$

$\mathrm{Cov}[\mathbf{z}] = \mathbb{E}[(\mathbf{z} - \mathbb{E}[\mathbf{z}])(\mathbf{z} - \mathbb{E}[\mathbf{z}])^T]$
$= \mathbb{E}[(\mathbf{W}\mathbf{x} + \mathbf{b} - \mathbf{W}\boldsymbol{\mu} - \mathbf{b})(\mathbf{W}\mathbf{x} + \mathbf{b} - \mathbf{W}\boldsymbol{\mu} - \mathbf{b})^T]$
$= \mathbb{E}[(\mathbf{W}\mathbf{x} - \mathbf{W}\boldsymbol{\mu})(\mathbf{W}\mathbf{x} - \mathbf{W}\boldsymbol{\mu})^T]$
$= \mathbb{E}[(\mathbf{W}(\mathbf{x} - \boldsymbol{\mu}))(\mathbf{W}(\mathbf{x} - \boldsymbol{\mu}))^T]$
$= \mathbf{W}\mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T]\mathbf{W}^T$
$= \mathbf{W}\mathbf{S}\mathbf{W}^T$

b Provide an expression for the distribution $p(\mathbf{z})$ in terms $\mathbf{W}$, $\mathbf{b}$, $\boldsymbol{\mu}$ and $\mathbf{S}$.

1.0 point · Open question · 2/5Page

+0.25 points

A linear transformation of a Gaussian random variable is again normally distributed.
(Or an answer of this kind where it is mentioned that $\mathbf{z}$ is normaly distributed)

+0.25 points

The expectation and covariance of a Gaussian random variable are directly given by the mean and covariance matrix of the Gaussian.

+0.5 points

Thus $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{W}\boldsymbol{\mu} + \mathbf{b}, \mathbf{W}\mathbf{S}\mathbf{W}^T)$

c The individual elements in the latent vector $\mathbf{z} = (z_1, \ldots, z_M)^T \in \mathbb{R}^M$ will generally be correlated. What if we were interested in finding a $\mathbf{W}$ such that $\mathbf{z}$ will consist of uncorrelated elements, what would be a suitable algorithm/method for obtaining such a $\mathbf{W}$?

1.0 point · Open question · 2/5Page

+1 point

$\mathbf{W}$ could be obtained via principal component analysis.

+1 point

Alternative. We can do an eigendecomposition of $\mathbf{S}$ and use M (doesn't matter which one as it is about independency) eigenvectors for $\mathbf{W}$. I.e., $\mathbf{W} = \mathbf{U}_M$, with $\mathbf{U}_M$ a matrix with M eigenvectors as colums.

d Suppose we only have access to a sampling algorithm that can sample from normal distributions $N(\mathbf{0}, \mathbf{I})$ with zero mean and unit diagonal covariance matrix. Which trick could you use to still sample variables that are distributed according to $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu, \mathbf{S})$ for arbitrary $\boldsymbol{\mu}$ and $\mathbf{S}$? Describe the steps.

1.0 point · Open question · New page · 2/5Page

+0.34 points

Via the reparametrization trick.

+0.34 points

This requires to first decompose $\mathbf{S}$ into $\mathbf{S} = \tilde{\mathbf{W}}^T \tilde{\mathbf{W}}$, e.g. via eigen- or cholesky-decomposition.

+0.34 points

Then a sample $\tilde{\mathbf{x}}$ drawn from $\mathcal{N}(0, \mathbf{I})$ is mapped to $\mathbf{x}$ via

$$\mathbf{x} = \tilde{\mathbf{W}}\tilde{\mathbf{x}} + \boldsymbol{\mu}.$$

($\mathbf{x}$ will then be a random variable with mean $\boldsymbol{\mu}$ and covariance $\mathbf{S}$)

e **[BONUS]** Proof that with your choice for $\mathbf{W}$ in exercise **(c)** the elements z_i are indeed uncorrelated.

1.0 point · Open question · Bonus · 2/5Page

+1 point

Any answer following roughly this structure will be OK.

In principal component analysis we obtain M orthonormal eigenvectors from $\mathbf{S}$, stored as colums in $\mathbf{U}_M \in \mathbb{R}^{M \times d}$ and with eigenvalues stored in a diagonal matrix Λ_M.
(or alternative if W was given by any set of eigenvectors (does not have to be sorted) is also fine)

The covariance matrix $\mathbf{S}$ can be diagonalized via $\mathbf{U}^T \Lambda \mathbf{U}$ with U the eigenvectors and Λ a diagonal matrix containing the eigenvalues.

Then, if we set $\mathbf{W} = \mathbf{U}_M^T$, with $\mathbf{U}_M$ any set of eigenvectors we have

$$\mathrm{Cov}[\mathbf{z}] = \mathbf{WSW} = \mathbf{U}_M \mathbf{U}^T \Lambda \mathbf{U} \mathbf{U}_M^T = \Lambda_M,$$

and thus the covariance matrix will be diagonal.

14 SVM regression

15.0 points · 8 questions

***Figure 1**: **(a)** The tolerance error function E_ϵ. **(b)** Illustration of SVM regression.*

Consider the problem of fitting a function $y(\mathbf{x}; \mathbf{w}, \mathbf{b}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + \mathbf{b}$ to a dataset of point pairs $D = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$, with $\mathbf{x}_n \in \mathbb{R}^d$ and $t_n \in \mathbb{R}$. For notational convenience we drop the dependency on parameters $\mathbf{w}$ and $\mathbf{b}$ of the model and simply write $y(\mathbf{x})$. Furthermore, we may simply denote $y(\mathbf{x}_n)$, the prediction for the n^{th} data point, with y_n. We fit by solving the following problem

$$\min_{\mathbf{w},\mathbf{b}} \quad C \sum_{n=1}^{N} E(y_n, t_n) + \frac{1}{2}\|\mathbf{w}\|^2, \qquad (1)$$

in which $E(y(\mathbf{x}), t)$ penalizes errors made by the model. Instead of using the usual squared error loss (see Figure 1(a)), we allow for a tolerance of ϵ. This is done by giving a 0 penalty if the prediction lies within the tolerance interval, and linearly penalize proportional to the distance towards the tolerance region using the following function (see also Figure 1(a))

$$E_\epsilon(y(\mathbf{x}), t) = \begin{cases} 0 & \text{if } \ |y(\mathbf{x}) - t| < \epsilon \\ |y(\mathbf{x}) - t| - \epsilon & \text{otherwise} \end{cases}.$$

The penalty-free region is then given by $y(\mathbf{x}) - \epsilon \leq t_n \leq y(\mathbf{x}_n) + \epsilon$ and will be referred to as the ϵ-tube.

Akin to the soft-margin support vector machine case, we allow for some slack and adjust the tolerance interval by possibly adding $\xi_n \geq 0$ at the top of the interval, or subtract $\hat{\xi}_n \geq 0$ at the bottom of the interval. We then obtain an equivalent inequality constraint optimization problem:

$$\min_{\mathbf{w},\mathbf{b},\{\xi_n\},\{\hat{\xi}_n\}} \quad C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n) + \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad \begin{array}{lr} \xi_n \geq 0, & \text{(I)} \\ \hat{\xi}_n \geq 0, & \text{(II)} \\ t_n \leq y_n + \epsilon + \xi_n, & \text{(III)} \\ t_n \geq y_n - \epsilon - \hat{\xi}_n. & \text{(IV)} \end{array}$$

Note that this is equivalent to (1) as the slack variables ξ_n and $\hat{\xi}_n$ capture the error beyond the tolerance ϵ. This problem is visualized in Figure 1b.

Description

a Write down the primal Lagrangian for this inequality constrained optimization problem. Use the following symbols for the Lagrange multipliers: for constraint (I) use μ_n, for (II) use $\hat{\mu}$, for (III) use a_n , and for (IV) use $\hat{a}_n$.

2.0 points · Open question · 1/4Page

Grading description

The Lagrangian is given by

$$L(\mathbf{w}, \mathbf{b}, \{a_n\}, \{\hat{a}_n\}, \{\mu_n\}, \{\hat{\mu}_n\}) = C\sum_{n=1}^{N}(\xi_n + \hat{\xi}_n) + \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N}(\mu_n\xi_n + \hat{\mu}_n\hat{\xi}_n)$$

$$-\sum_{n=1}^{N} a_n(\epsilon + \xi_n + y_n - t_n) - \sum_{n=1}^{N} \hat{a}_n(\epsilon + \hat{\xi}_n - y_n + t_n).$$

Grading: subtract half a point if all is OK except for a minus sign at one location.

+1 point

One point if all the terms are included

+1 point

Another point if all the signs are also correct.

b In the lectures we considered the stationarity conditions separately from the KKT conditions; in this exam we do the same. Write down the KKT conditions (not including the stationarity conditions).

3.0 points · Open question · 13/20Page

+0.5 points

(bonus/compensation for mistakes) For indicating up front that $n = 1, \ldots, N$ or denoting it per constraint.

-1 points

If everything else is correct, except for the signs in the constraints for (III) and (IV)

+1 point

For writing down all 4 cases of primal feasibilities:

$$\xi_n \geq 0$$

$$\hat{\xi}_n \geq 0$$

$$t_n \leq y_n + \epsilon + \xi_n \quad \text{or} \quad y_n + \epsilon + \xi_n - t_n \geq 0$$

$$t_n \geq y_n - \epsilon - \hat{\xi}_n \quad \text{or} \quad -y_n + \epsilon + \hat{\xi}_n + t_n \geq 0$$

+1 point

For writing down all 4 cases of dual feasibilities:

$$\mu_n \geq 0$$

$$\hat{\mu}_n \geq 0$$

$$a_n \geq 0$$

$$\hat{a}_n \geq 0$$

+1 point

For writing down all 4 cases of complementary slackness:

$$\xi_n \mu_n = 0$$

$$\hat{\xi}_n \hat{\mu}_n = 0$$

$$a_n(\epsilon + \xi_n + y_n - t_n) = 0$$

$$\hat{a}_n(\epsilon + \hat{\xi}_n - y_n + t_n) = 0$$

c How many KKT conditions do we have?

1.0 point · Open question · 1/20Page

+1 point

3*4*N = 12N

d In order to obtain the dual Lagrangian we need to compute the stationarity conditions. When optimizing the Lagrangian with respect to primal variables ξ_n and $\hat{\xi}_n$ respectively we obtain the following extra constraints on a_n and $\hat{a}_n$

$$\begin{aligned} &\text{for all } n = 1, \ldots, N: \quad & a_n &= C - \mu_n, \\ &\text{for all } n = 1, \ldots, N: \quad & \hat{a}_n &= C - \hat{\mu}_n. \end{aligned} \tag{2}$$

Complete the set of stationarity conditions by deriving them for the primal variables $\mathbf{w}$ and $\mathbf{b}$.

2.0 points · Open question · 13/20Page

+1 point

$$\frac{\partial L}{\mathbf{w}} = 0 \Rightarrow$$

$$\mathbf{w} - \sum_{n=1}^{N} a_n \boldsymbol{\phi}(\mathbf{x}_n) + \sum_{n=1}^{N} \hat{a}_n \boldsymbol{\phi}(\mathbf{x}_n) \Leftrightarrow$$

$$\mathbf{w} = \sum_{n=1}^{N} (a_n - \hat{a}_n) \boldsymbol{\phi}(\mathbf{x}_n)$$

+1 point

$$\frac{\partial L}{\mathbf{b}} = 0 \Rightarrow$$

$$-\sum_{n=1}^{N} a_n + \sum_{n=1}^{N} \hat{a}_n = 0 \Leftrightarrow$$

$$\sum_{n=1}^{N} (a_n - \hat{a}_n) = 0 \quad \text{(or with a } - \text{ sign)}$$

0 points

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow$$

$$C - \mu_n - a_n = 0 \Leftrightarrow$$

$$C = a_n + \mu_n$$

0 points

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow$$

$$C - \hat{\mu}_n - \hat{a}_n = 0 \Leftrightarrow$$

$$C = \hat{a}_n + \hat{\mu}_n$$

e The new constraints for the dual variables a_n and $\hat{a}_n$ (see Eq. (2) in exercise **(d)**), together with the original KKT conditions, define box constraints (min-max intervals) for a_n and $\hat{a}_n$. Write them down.

1.0 point · Open question · 1/10Page

+0.5 points

$$0 \leq a_n \leq C$$

+0.5 points

$$0 \leq \hat{a}_n \leq C$$

f Without actually solving the dual problem, we can already tell that there will be different type of solutions for the values of the dual variables. In the following three figures, we ask you to encircle the requested instances:

1. In subfigure (i) encircle the points for which both $a_n = 0$ and $\hat{a}_n = 0$.
2. In subfigure (ii) encircle the points for which $0 < a_n < C$.
3. In subfigure (iii) encircle the points for which $\hat{a}_n = C$.

Hint: The KKT and optimality conditions allow you to reason about the solution.

3.0 points · Image · 1/4Page

+1 point

Figure (i): all points inside (not on the boundary) of the ϵ-tube should be marked.

+1 point

Figure (ii): all points at the **top of the boundary** should be marked. Zero points if also the bottom points are marked (the provided condition is for a_n, not $\hat{a}_n$).

+1 point

Figure (iii): The outlier at the bottom should be marked.

g The stationarity conditions that you computed in question **(d)** should give you an expression for $\mathbf{w}$ in terms of the dual variables. Use it to derive a dual formulation for the predictive model y that no longer depends on $\mathbf{w}$.

In case you can't rely on the result of question (d), use the following expression for $\mathbf{w} = \sum_{n=1}^{N} f(a_n, \hat{a}_n, \mu_n, \hat{\mu}_n)\boldsymbol{\phi}(\mathbf{x}_n)$, *where* $f(a_n, \hat{a}_n, \mu_n, \hat{\mu}_n)$ *is introduced to indicate that* $\mathbf{w}$ *may still depend on the Lagrange multipliers.*

1.0 point · Open question · 1/5Page

+1 point

Substitute the expression for $\mathbf{w}$ into $y(\mathbf{x}) = \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})$ gives

$$y(\mathbf{x}, \{a_n\}\{\hat{a}\}) = \sum_{n=1}^{N}(a_n - \hat{a}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T\boldsymbol{\phi}(\mathbf{x})$$

-0.5 points

For missing a transpose or forgetting indexing with n of $\mathbf{x}_n$.

h Consider the following. You have access to a solver that gives you the optimizer of the dual problem, and thereby the optimal solution for the primal problem as well. It turns out that you have a hard time choosing an appropriate set of basis functions to create the feature vectors $\boldsymbol{\phi}(\mathbf{x})$ that are used in the predictive model. What can you do to circumvent the problem of having to make specific choices for $\boldsymbol{\phi}$? Explicitly state what changes you make to the predictive model and/or the optimization steps.

2.0 points · Open question · 2/5Page

+1 point

You can apply the kernel trick. I.e., replace all instances of $\boldsymbol{\phi}(\mathbf{x}_n)^T\boldsymbol{\phi}(\mathbf{x})$ with a kernel $k(\mathbf{x}_n, \mathbf{x})$.

+0.5 points

Adjust predictive model (akin to 14g)

+0.5 points

Adjust also the dual Lagrangian prior to solving the dual problem.

+1 point

For answer based on Gaussian Processes. This is also a non-parametric method. However the answer does not connect to the exercise and what is asked, hence not full points.

+0.5 points

Some points for NN based answer as it indeed circumvents the need to actually choose basis functions, they are learned instead. Additionally, the model class is however still parametric and the answer does not connect to the exercise and what is asked. Hence only a few points are awarded for this answer.