Started on	Saturday, 27 July 2024, 9:57 PM
State	Finished
Completed on	Saturday, 27 July 2024, 10:09 PM
Time taken	11 mins 55 secs
Marks	50/50
Grade	**10** out of 10 (**100**%)

Question 1

Correct

Mark 13 out of 13

Which of the following are models for linear classification, i.e., classifies inputs into categorical output classes by means of linear decision boundaries?

- [ ] 1. SVM with polynomial kernel
- [ ] 2. QDA
- [ ] 3. linear Gaussian naive Bayes model on continuous features
- [ ] 4. single-hidden-layer DNN with ReLU activation
- [x] 5. LDA
- [x] 6. naive Bayes model on binary features — This rewrites to a special type of logistic regression model, see, e.g. this paper or Sec. 2.4 in these lecture notes.
- [x] 7. DNN with linear activation — This is essentially a linear function similar to a SVM.
- [x] 8. SVM with linear kernel
- [ ] 9. polynomial regression
- [x] 10. logistic regression
- [ ] 11. linear regression
- [x] 12. single-hidden-layer DNN with linear activation — This is simply a linear function.
- [x] 13. SVM — Vanilla support vector machines are linear models.

Your answer is correct.

Correct

Marks for this submission: 13/13.

Question **2**

Correct

Mark 8 out of 8

Match the following binary classification models $\mathbb{R}^n \to \{y_+, y_-\}, x \mapsto \big(t < \mathrm{discr}_\theta(x)\big)$ to their discriminator definitions. The models to assign are:

- **SVM**: The distance of to the decision hyperplane exceeds a margin
- **Logistic regression**: The probability of class y_+ is greater than 0.5
- **QDA**: The (normal) conditional probability of class y_+ is greater than that of y_-
- **LDA**: as QDA, but with equal covariance matrices
- **Linear Gaussian Naive Bayes**: The conditional probability of class y_+ is greater than that of y_-

$f(x) = y_+$ if ...

$1 < \frac{P(y_+)\exp(\frac{1}{2}(x-\mu_+)^T\Sigma_+^{-1}(x-\mu_+))}{P(y_-)\exp(\frac{1}{2}(x-\mu_-)^T\Sigma_-^{-1}(x-\mu_-))}$	QDA
$1 \leq \frac{\exp\left(\frac{1}{2}(x-\mu_+)^T\Sigma^{-1}(x-\mu_+)\right)}{\exp\left(\frac{1}{2}(x-\mu_-)^T\Sigma^{-1}(x-\mu_-)\right)}$	LDA
\(0.5 < P(\pm y \mid x) = \tfrac{\exp(w^Tx+b)}{1 + \exp(w^Tx + b)} \)	Logistic regression
\(1 < \tfrac{P(y_+\mid x)}{P(y_-\mid x)}= \tfrac{\prod_i P(x_i\mid y_+)P(y_+)}{\prod_i P(x_i\mid y_-)P(y_-)}= \tfrac{P(y_+)}{P(y_-)}\cdot \tfrac{\prod_i \exp(\frac{1}{2\sigma_{+,i}^2}(x_i-\mu_{+,i})^2)}{\prod_i \exp(\frac{1}{2\sigma_{-,i}^2}(x_i-\mu_{-,i})^2)} \)	Linear Gaussian Naive Bayes
\(1\leq \big(w^Tx + b \big) \)	SVM

Your answer is correct.

Correct

Marks for this submission: 8/8.

Question 3

Correct

Mark 3 out of 3

Match the following optimization objectives to their respective dual formulation.

(A) $\text{argmin}_x x^T c \quad\text{ s.t. } Ax\leq b$
(B) $\text{argmin}_w \frac{1}{2} w^Tw \quad\text{ s.t. } Aw + b \leq 0$
(C) $\text{argmin}_w \frac{1}{2} w^Tw \quad\text{ s.t. } 1 - by - y^TXw = 0$

$\text{argmax}_{\lambda\geq0} \sum_i\lambda_i - \frac{1}{2} \lambda^TX'^TX'\lambda \quad\text{ s.t. } \lambda^Ty=0$, where $X':=(y_1x_1,\dots, y_nx_n)$	(C)
$\text{argmax}_{\lambda\geq 0} - \lambda A^TA\lambda + \lambda^T b$	(B)
$\text{argmax}_\lambda -\lambda^T b \quad\text{ s.t. } A^T\lambda + c = 0$	(A)

Your answer is correct.

Find more examples and derivations of some of the dualities here.

These are the relevant intermediate steps:

For (A): $L^*(\lambda) = \min_x ((c^T+\lambda^TA)x - \lambda^Tb) = -\infty$ if we do not constrain $A^T\lambda + c = 0$.

For (B): $L(w,A,b,\lambda) = \frac{1}{2} w^Tw + \lambda^T A w + \lambda^T b$ and $0 = \frac{dL}{dw} = w + \lambda^T A$, so, $\lambda^* = \text{argmax}_{\lambda\geq 0} -\frac{1}{2}\lambda A^TA\lambda + \lambda^T b$

For (C): note that this is a reformulation of the SVM objective.

Correct

Marks for this submission: 3/3.

Question 4

Correct

Mark 10 out of 10

What we have seen so far

Recall that in the lecture we saw how the optimization objective

$$ \text{argmin}_w \|w\|^2 \quad\text{ s.t. } \forall i: y_iw^Tx_i+b\geq 1 $$

of a support vector machine can be reformulated into its dual

$$ \text{argmax}_\lambda \text{argmin}_{w,b} L(w,b,\lambda),\quad L(w,b,\lambda) = \frac{1}{2}\|w\|^2 + \sum_i \lambda_i (1-y_i(w^T x_i +b)) .$$

With \(0 \overset{!}{=} \frac{dL}{dw} = w-\sum_i\lambda_iy_ix_i \) and \(0 \overset{!}{=} \frac{dL}{db} = \sum_i\lambda_iy_i \) we showed how to find the optimal \(\lambda_i \) as

$$ \lambda^* = \sum_i\lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i\lambda_jy_iy_j x_i^Tx_j .$$

Applying the **kernel trick**, i.e., replacing the input \(x\) by its transformed version \(\phi(x)\), we can express the \(\lambda^* \) in terms of a kernel \(k(x,x') = \phi(x)^T\phi(x') \):

$$ \lambda^* = \sum_i\lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i\lambda_jy_iy_j k(x_i,x_j) .$$

Kernelizing the remaining formulas

To express the SVM inference condition via kernels, we first need to find the formulas for \(w \) and \(b \). For this, use the equation \(\frac{dL}{dw}=0 \), and the equality \(y_b = w^T x_b + b \) which holds for any training sample \((x_b,y_b)\) on the margin's boundary.

This can then be inserted into the inference condition \(f(x) = \left(w^T x + b \right) \geq 0 \).

How does this reformulate in terms of only using \(k\) instead of any mention of \(\phi\)?

- [ ] a. \(f(x) = 0\leq \sum_i \lambda_iy_i k(x_i, x) + y_b - \sum_i\lambda_iy_i x_i^Tx_b \)
- [ ] b. \(f(x) = 0\leq \sum_i y_i k(x_i, x) + y_b - \sum_i y_ik(x_i, x_b) \)
- [ ] c. \(f(x) = 0\leq \sum_i \lambda_iy_i k(x_i, x - x_b) + y_b \)
- [x] d. \(f(x) = 0\leq \sum_i \lambda_iy_i(k(x_i, x) - k(x_i,x_b)) + y_b \)
- [x] e. \(f(x) = 0\leq \sum_i \lambda_iy_i k(x_i, x) + y_b - \sum_i\lambda_iy_ik(x_i, x_b) \)

Your answer is correct.

Correct

Marks for this submission: 10/10.

Question 5

Correct

Mark 8 out of 8

A couple of the models we have seen so far can be reformulated in terms of kernels, thus summarizing calculations of any transformation into the kernel function.

*Note: A new regression model we can define is the **Naydayara-Watson kernel regression**: It simply sets the regression outcome |(f(x) |) to the weighted mean of training data outcomes |(y_i |). The weight for |(y_i|) is the similarity between the new input |(x|) and the respective training input |(x_i|).*

Match the formulations based on kernel \(k \) to the models from above or the lecture.

Tipp: What is the formula for |(f(x) |) for the respective models from the lecture? Can you find occurrences of some distance measurement like |(|/x-x'|/|) (Euclidean) or |(x^Tx'|) (cosine similarity / measuring "angle) in these formulas that may be replaced by a kernel? Which of the formulas below could match them? Use a process of elimination to determine the remaining matches.

\(f(x) = \sum_i k(x_i,x)\tilde{k}_iy_i \) where \(K := \left(k(x_i,x_j)\right)_{i,j}, \tilde{K}=\left(\tilde{k}_1,\dots,\tilde{k}_n\right) := K^{-1} \)	linear regression
\(f(x) = \frac{\sum_{i\leq n} k(x, x_i)y_i}{\sum_{i\leq n} k(x, x_i)} \) for some kernel s.t. \(k(\cdot, x')\) suffices the constraints of a probability (integrates to 1).	Naydayara-Watson kernel regression
\(f(x) = \text{argmax}_{y_i} k(x, x_i) \)	k-nearest neighbors
\(f(x) = \big(0\leq k(w, x) + b \big)\)	SVM

Your answer is correct.

Correct

Marks for this submission: 8/8.

Question 6

Correct

Mark 8 out of 8

Given below constraints from the problem formulation, which kernel is a good choice for the start?

Let's consider the following selection of (pretty standard) kernels:

- **Linear**: $k(x,z) = x^Tz + c$
- **Polynomial**: $k(x,z) = (x^Tz + c)^d$
- **Radial basis function (RBF)**: $k(x,z) = \exp(-\gamma\|x-z\|_2^2)$
- **Fisher**: $k(x, z) = g(x,\theta)F^{-1}g(z,\theta)$ for the Fisher score g with respect to a joint probability distribution model specifying $P(X, \theta)$

We need to avoid costly **hyperparameter tuning**.	linear kernel
One only has **few data points** with a **large number of features**; the problem is prone to **overfitting**.	linear kernel
The features have complex **non-linear dependencies**.	RBF kernel
The input features are **polynomially dependent**.	polynomial kernel
We need a formulation that allows for **fast training**.	linear kernel
The data for classification is **linearly separable**.	linear kernel
We want to do binary classification and assume an **elliptical decision boundary**.	RBF kernel
The goal is to measure the **similarity of events** wrt. to a statistical model and provided evidence.	Fisher kernel

Your answer is correct.

Correct

Marks for this submission: 8/8.

◄ 06. Quiz - Linear Models

Jump to...

08. Quiz - Version Space Learning ►