Resit Exam

52041MAL6Y Machine Learning 1 23/24 (Period 1.1) · 5 exercises · 47.5 points

Multiple Choice #28376409

8 pts · Last updated 8 Jan, 2024, 16:22 · Saved in ML1_2023_exams

Which of the following statements about linear models is **incorrect**?

1 pt · Multiple choice · 4 alternatives

- ○ Linear discriminant analysis and logistic regression have the same expressive power (i.e. in the types of decision boundaries that they can represent). 0.0
- ○ For logistic regression, the decision boundary lies perpendicular to the parameter vector. 0.0
- ◉ **Logistic regression models the difference in the class probabilities, i.e. $P(C_1|X) - P(C_0|X)$, where C_k denotes the class and X the features.** 1.0

  Feedback

  Logistic regression models the difference in **log probabilities:** $$\log \frac{P(C_1 | X)}{P(C_0 | X)} = \log {P(C_1 | X) - \log {P(C_0 | X)$$
- ○ Quadratic discriminant analysis can represent all possible decision boundaries that linear discriminant analysis can represent. 0.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Consider building a Bayesian predictive model: let X denote the features, t the label, θ the parameters, $p(\theta)$ the prior, and $p(t|X,\theta)$ the likelihood. Which of the following quantities does the Bayesian framework *ideally* use to make predictions on test data, denoted X^* and t^*?

1 pt · Multiple choice · 4 alternatives

- ○ the posterior mode (a.k.a. MAP estimator): $p(t^*|X^*,\hat{\theta}_{MAP})$ — 0.0
- ○ the maximum likelihood estimator (MLE): $p(t^*|X^*,\hat{\theta}_{MLE})$ — 0.0

  Feedback
  0
- ◉ **the posterior predictive distribution: $p(t^*|X^*,t,X) = \int_\theta p(t^*|X^*,\theta)\cdot p(\theta|t,X)d\theta$** — 1.0
- ○ the marginal likelihood: $p(t^*|X^*) = \int_\theta p(t^*|X^*,\theta)\cdot p(\theta)d\theta$ — 0.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Which of the following statements about Gaussian mixture models (GMMs) is **incorrect**?

1 pt · Multiple choice · 4 alternatives

- ◉ **Fitting a GMM with the EM algorithm usually returns the globally optimal parameter setting.** — 1.0
- ○ GMMs are a more flexible clustering model / algorithm than k-means, meaning that they can represent clustering configurations that k-means cannot. — 0.0
- ○ GMMs assume that each data point is drawn from one of K Gaussian distributions (where K is the total number of mixture components). — 0.0
- ○ A GMM represents a distribution that is at least as expressive as any single one of its component distributions. — 0.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

The kernel trick is a key concept used in support vector machines to solve non-linear problems. How does it achieve this?

1 pt · Multiple choice · 4 alternatives

○ By reducing the dimensionality of the feature space, making it easier to find a linear separator.	0.0
◉ **By transforming the original finite-dimensional space into a higher-dimensional space, potentially making the data separable by a hyperplane.**	1.0
○ By calculating the convex hulls for classes in the feature space and finding the linear separators between them.	0.0
○ By applying stochastic gradient descent in the primal space to ensure non-linearity in the decision boundary.	0.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Which of the following statements about kernel methods is **true**?

1 pt · Multiple choice · 4 alternatives

○ A kernel always operates in an infinite-dimensional space.	0.0
○ Kernel methods can only be applied to SVMs.	0.0
○ Using more complex kernels always results in better model performance.	0.0
◉ **Kernel methods allow SVMs to classify non-linearly separable data.**	1.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Which of the following statements about *the bootstrap* algorithm is **true**?

1 pt · Multiple choice · 4 alternatives

- ○ The bootstrap takes weak models and ensembles them to achieve strong aggregate performance. 0.0

  Feedback

  This is boosting.

- ◉ **The bootstrap aims to simulate the noise / variation introduced when the original data set was sampled from the underlying population.** 1.0

  Feedback

  In the bootstrap, we 'imagine' that the original dataset is actually the underlying population and sample from it to create new data sets, which hopefully mimics how the original dataset was sampled from the population.

- ○ The bootstrap creates new data sets that are smaller in size (i.e. number of data points) to improve computational efficiency. 0.0

  Feedback

  The synthetic data sets are of the same size as the original.

- ○ The bootstrap should never be used with high-dimensional data as it will likely result in overfitting. 0.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Which of the following models is **not** a generative model?

1 pt · Multiple choice · 4 alternatives

○ Probabilistic PCA	0.0
○ Quadratic Discriminant Analysis	0.0
○ Gaussian Mixture Model	0.0
◉ **Decision Tree**	1.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

You find that your decision tree is overfitting. Which of the following steps would best help reduce the overfitting? (choose just one answer)

1 pt · Multiple choice · 4 alternatives

◉ **Prune the decision tree so that its depth is decreased.**	1.0
○ Prune the decision tree so that its width is decreased.	0.0
○ Remove feature bagging.	0.0
○ Re-fit the tree and do not use information gain as the splitting heuristic.	0.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Modelling Categorical Data #28376407

10.5 pts · Last updated 26 Jan, 2024, 15:38 · Saved in ML1_2023_exams

Imagine a renowned casino that has just unveiled its latest slot machine. On each pull, the machine produces one of the K distinct outcomes. Although the casino claims that each of the K outcomes is equally likely, your goal in this exercise is to test this assertion. Having recently completed a Machine Learning 1 course, you begin by collecting a dataset of independent pulls, $\mathcal{D} = \{x_1, \ldots, x_N\}$, where each $x_n \in \{1, \ldots, K\}$. You then assume a categorical distribution to model the outcomes of the slot machine:

$$p(x_n = k|\boldsymbol{\theta}) = \theta_k$$

where we assume $1 \geq \theta_k \geq 0, \ \forall k$ (otherwise the likelihood is not defined). Note that since only one of the K events can be observed for each pull, the parameter vector $\boldsymbol{\theta}$ must sum to one:

$$\sum_{k=1}^{K} \theta_k = 1 \, .$$

Text

Derive the maximum likelihood estimate $\boldsymbol{\theta}_{MLE}$ based on the dataset $\mathcal{D}$. *Hint:* Do not forget to take the constraint on the model parameters into consideration.

4 pts · Open · 1 1/10 Page

The likelihood is given as

$$p(D \mid \{\theta_k\}) = \prod_{n=1}^{N} p(x_n \mid \{\theta_k\}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \theta_k^{t_{nk}}$$

with t_{nk} a one hot encoding of the categorical variable. 0.5 pts

The log-likelihood as (OK if directly given):

$$\sum_{n-1}^{N} \sum_{k=1}^{K} t_{nk} \log \theta_k$$

Constraint optimization, so define Lagrangian:

$$L := \sum_{n=1}^{N} \sum_{k=1}^{K} [t_{nk} \log \theta_k] + \lambda \left(\sum_{k=1}^{K} [\theta_k] - 1 \right)$$ 1 pt

Compute derivative w.r.t. θ_k and set to zero

$$\frac{\partial L}{\partial \theta_k} = \sum_{n=1}^{N} \frac{t_{nk}}{\theta_k} + \lambda = 0$$ 0.5 pts

Derive expression for θ_k:

$$\theta_k = -\frac{1}{\lambda} \sum_{n=1}^{N} t_{nk}$$ 0.5 pts

Compute derivative w.r.t. λ and set to zero:

$$\frac{\partial L}{\partial \lambda} = \sum_{k=1}^{K} \theta_k - 1 = 0$$ 0.5 pts

Fill in the derived expression $\theta_k = -\frac{1}{\lambda} \sum_{n=1}^{N} t_{nk}$ and move 1 and lambda to other hand side:

$$\lambda = -\sum_{k=1}^{K} \sum_{n=1}^{N} t_{nl}$$ 0.5 pts

Note that $\sum_{l} t_{nl} = 1$ and use it to derive expression for λ:

$$\lambda = -N$$

and the final answer thus is

$$\theta_k = \frac{1}{N} \sum_{n=1}^{N} t_{nk}$$ 0.5 pts

which equals the fraction of case $x_n = k$ as $\theta_k = \frac{N_k}{N}$ with $N_k = \sum_{n=1}^{N} t_{nk}$.

As an experienced modeler, you know that you cannot always rely on the MLE, especially in small-data settings. Therefore, you decide to assume a prior distribution over $\boldsymbol{\theta}$. The Dirichlet distribution

$$\mathrm{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})}\prod_{k=1}^{K}\theta_k^{\alpha_k-1}$$

is a natural choice as its support corresponds to K-dimensional vectors satisfying the constraint specified above. Here $\alpha_k > 0$ are the so-called concentration parameter. The normalization constant $B(\boldsymbol{\alpha})$ is a multivariate beta function.

Derive the log-posterior. You do not have to provide explicit expression for $B(\boldsymbol{\alpha})$ or any other terms that do not depend on $\boldsymbol{\theta}$. That is, you can treat them as a constant.

1.5 pts · Open · 2/5 Page

Provide the terms of the log-posterior as

$$\log\ likelihood + \log\ prior - \log\ evidence,$$

0.5 pts

in which the latter is treated as a constant, denoted with C.

Correctly compute the log of the prior and fill in the expressions

$$\underbrace{\sum_{n=1}^{N}\left[\sum_{k=1}^{K} t_{nk}\log\theta_k\right]}_{\log-\text{likelihood}} + \underbrace{\sum_{k=1}^{K}\left[(\alpha_k-1)\log\theta_k\right]}_{\log-\text{prior}} - \underbrace{C}_{-\log\text{evidence}-\log B(\boldsymbol{\alpha})}$$

1 pt

Find the maximum a posteriori estimate $\boldsymbol{\theta}_{\text{MAP}}$. *Hint:* Do not forget to take the constraint on the model parameters into consideration.

3 pts · Open · 1 Page

Objective: Compute derivative of Lagrangian w.r.t. θ_k and set to zero (do not forget about the constraint!). 0.5 pts

$$\frac{\partial}{\partial \theta_k} \sum_{n=1}^{N} \left[\sum_{k=1}^{K} t_{nk} \log \theta_k \right] + \frac{\partial}{\partial \theta_k} \sum_{k=1}^{K} \left[(\alpha_k - 1) \log \theta_k \right] + \frac{\partial}{\partial \theta_k} \lambda \left(\sum_{l=1}^{K} [\theta_l] - 1 \right) = 0$$

Compute derivatives

1. Derivative of first term: $\sum_{n=1}^{N} \left[\frac{t_{nk}}{\theta_k} \right]$.
2. Derivative of second term: $\frac{\alpha_k - 1}{\theta_k}$. 1 pt
3. Derivative of third term: λ.

Thus:

$$\frac{1}{\theta_k} \left(\sum_{n=1}^{N} [t_{nk}] + \alpha_k - 1 \right) + \lambda = 0$$

Expression for θ_k thus becomes

$$\theta_k = - \frac{\sum_{n-1}^{N} [t_{nk}] + \alpha_k - 1}{\lambda}$$ 0.5 pts

Same as before, compute derivative: w.r.t. λ and set to zero gives

$$\sum_{k=1} \theta_k = 1$$

Plug-in above derived expression for θ_k:

$$- \frac{1}{\lambda} \left(\sum_{n=1}^{N} [\sum_{k=1}^{K} [t_{nk}]] + \underbrace{\sum_{k=1}^{K} [\alpha_k]}_{:= N_\alpha} - \sum_{k=1}^{k} [1] \right) = 1$$ 0.5 pts

and derive the expression for λ:

$$\lambda = -(N + N_\alpha - K)$$

The final solution thus is given by

$$\theta_k = \frac{\sum_{n-1}^{N} [t_{nk}] + \alpha_k - 1}{N + N_\alpha - K}$$ 0.5 pts

For what value of concentration parameters $\boldsymbol{\alpha}$, will the resulting $\boldsymbol{\theta}_{\text{MAP}}$ estimator reduce back to the maximum-likelihood estimator $\boldsymbol{\theta}_{\text{MLE}}$?

1 pt · Open · 3/10 Page

When for each k we have $\alpha_k = 1$. This can be recognized by the Direchlet distribution just being a constant (uniform) prior. Or based on the previous answer by recognizing that than we get the same answer as before. 1 pt

A friend of yours, who works at the casino, discreetly told you that he heard a rumour about even outcomes ($k = 2, 4, \ldots$) being more likely than odd outcomes ($k = 1, 3, \ldots$). How would you incorporate this additional information into your model of the slot machine's behaviour?

1 pt · Open · 3/5 Page

By updating the constraint. 0.5 pts

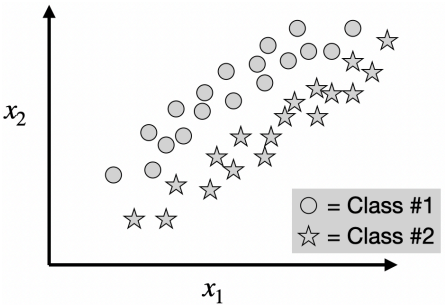
Explicitly, the constraint $\sum_{k=1}^{K} \theta_k = 1$ is replaced by two constraints:

1. $\sum_{k=even} [\theta_k] = \frac{2}{3}$ 0.5 pts
2. $\sum_{k=odd} [\theta_k] = \frac{1}{3}$

Principal Component Analysis (PCA) #28377033

8 pts · Last updated 9 Jan, 2024, 11:03 · Saved in Final Exam

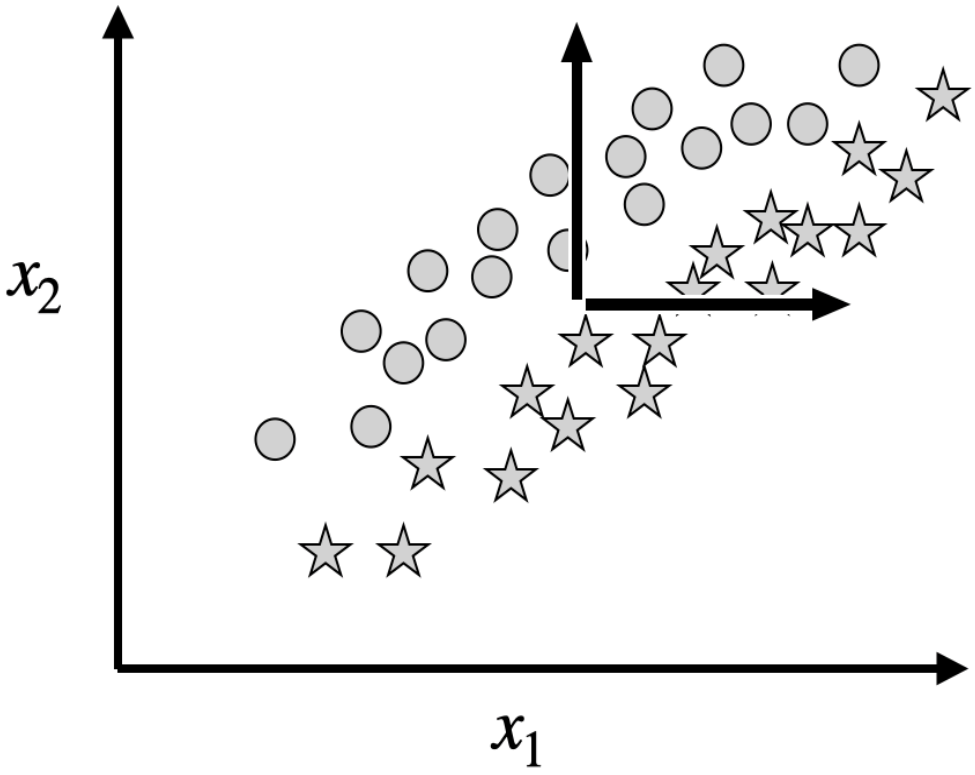
Consider applying ***principal component analysis*** **(PCA)** to the following two-dimensional data set, which has two classes.



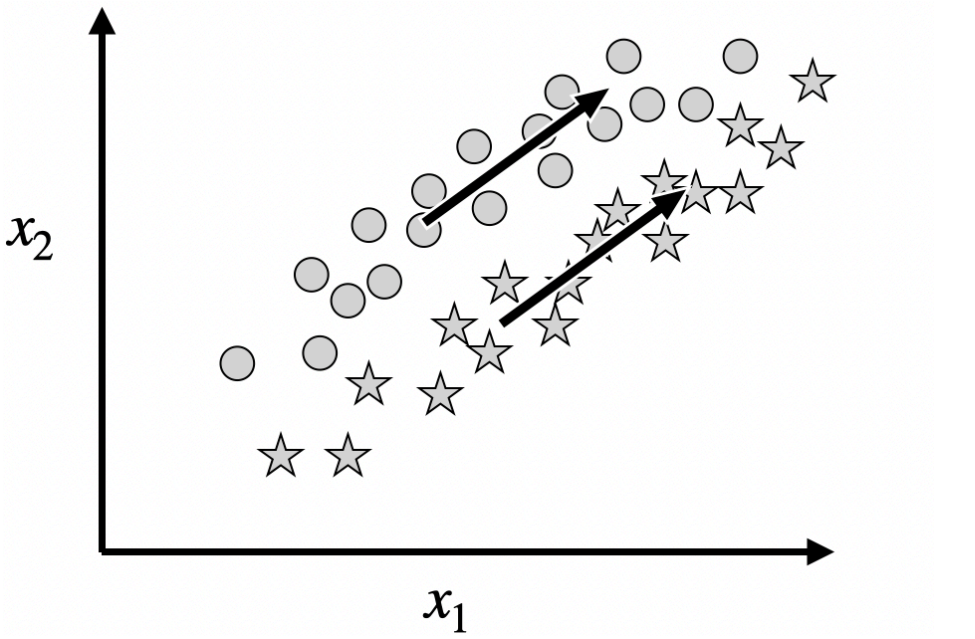
Text

Which of the following images shows the components (i.e. the eigenvectors) for this data set (from which we would select the principal one)?

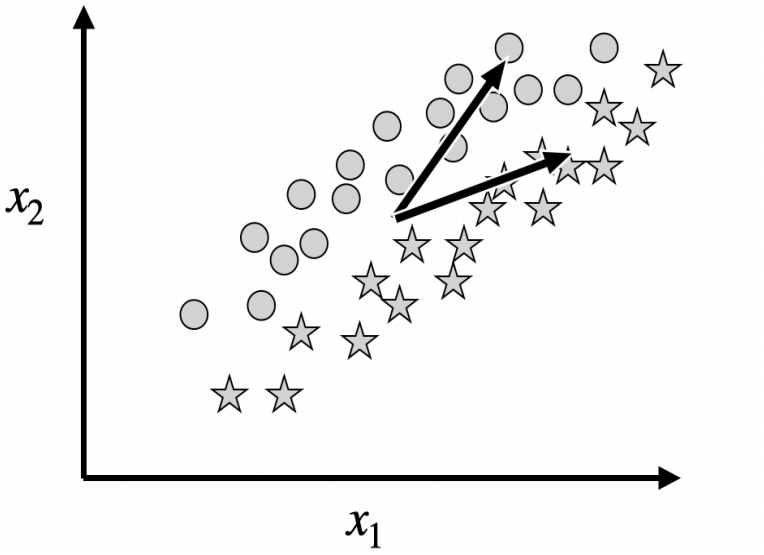
2 pts · Multiple choice · 4 alternatives



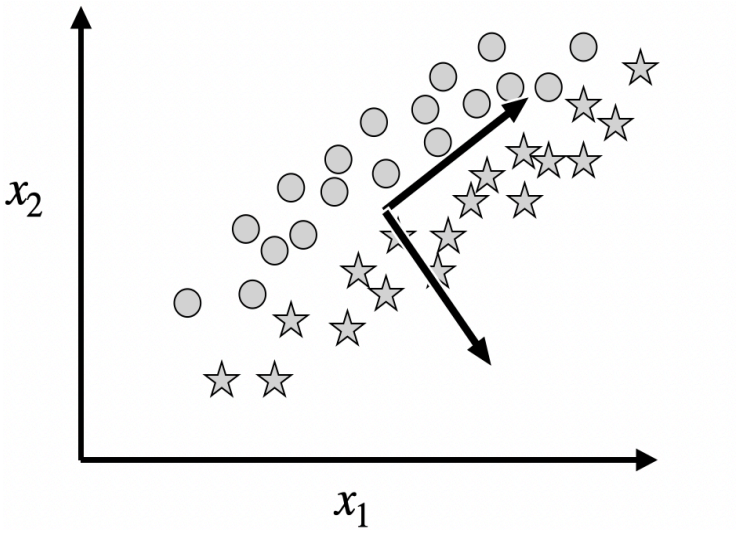
0.0



0.0



0.0



2.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Assume that the eigenvalues for the two components are $\lambda_1 = 7/10$ and $\lambda_2 = 1/10$. What *fraction* of the data's variance will be retained if we used PCA to reduce the data's dimensionality down to one dimension?

2 pts · Multiple choice · 6 alternatives

○ 7/10	0.0
◉ **7/8**	2.0
○ 1/10	0.0
○ 1/8	0.0
○ 8/10	0.0
○ 2/8	0.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Assume we are interested in training a classifier on this data set. Should we (**A)** apply PCA to the data first (to reduce it down to one feature dimension), or (**B)** leave the data as is (in two dimensions)?

1 pt · Multiple choice · 2 alternatives

○ **A**: Apply PCA, reducing the data to one feature dimension	0.0
◉ **B: Don't apply PCA, leave the data in two dimensions, as is shown above.**	1.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Give your reasoning behind your answer to the preceding question (for choosing A vs B).

1 pt · Open · 7/20 Page

States that performing PCA would collapse the data along the axis that does not preserve class information, essentially mixing together the classes and making the data useless from a classification perspective. — 1 pt

As visible in the figure, the point clouds are best separated along the direction of the smallest principal component. Discarding the second component gives features projected along to direction in which the clouds maximally overlap, and this is thus not a wise decision. When retaining both dimensions a hyperplane could still be found that separates the two classes. — 1 pt

Let X represent a $(N \times D)$-matrix of data with N observations and D features. Let W be a $(D \times K)$-semi-orthogonal matrix, with $D > K$, meaning that W's columns are orthonormal vectors.

Write down a loss function in terms of X and W such that it, when minimized with respect to W, is equivalent to fitting a PCA model with K components. Assume that your optimizer maintains the semi-orthogonality property of W. You may also assume the data is zero centered, i.e., $\sum_{n=1}^{N} \mathbf{x}_n = \mathbf{0}$ with $\mathbf{x}_n$ being the rows of X.

2 pts · Open · 1/2 Page

Formulate a reconstruction loss or maximum variance loss — 1 pt

Reconstruction loss: note that $Z = XW$ is the project data matrix of shape $(D \times K)$. The reconstruction is given by $\hat{X} = ZW^T$. A squared error (Frobenius norm, denoted with $\|\cdot\|_2$) on the elements of the reconstruction compared to the original data would be the suitable loss:

$$loss = \|X - \hat{X}\|_2^2$$

Maximum variance loss: Note that we define a loss so we want to minimize the negative total variance. Note that the projected data is given by $Z = XW$ and its covariance matrix $\mathrm{Cov}[\mathbf{z}, \mathbf{z}] = \frac{1}{N} Z^T Z$. The total variance could be defined as the Frobenius norm of the covariance matrix, and thus the loss could be given as — 1 pt

$$loss = -\|W^T X^T X W\|_2^2$$

Support Vector Regression #28376408

8.5 pts · Last updated 19 Jan, 2024, 15:11 · Saved in ML1_2023_exams

You recently learned about Support Vector Machines for classification and are curious about the regression setting. Assume a dataset $\mathcal{D} = \{(\mathbf{x}_1, t_1) \ldots, (\mathbf{x}_N, t_n)\}$, where $\mathbf{x}_n \in \mathbb{R}^D$ and $t_n \in \mathbb{R}$. The regression prediction is given by $y(\mathbf{x}_n) = \mathbf{w}^T\mathbf{x}_n + b$ with $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$.

Consider further, the ϵ-insensitive cost function $l_\epsilon(y(\mathbf{x}), t) = \max(0, |y(\mathbf{x}) - t| - \epsilon)$. This cost function is 0 if the absolute difference between prediction and target is smaller then ϵ. Using this cost function, we can now formulate the regression task as a constraint optimization problem. To soften the assumptions and allow for errors, we introduce slack variables $\{\xi_n\}$ and $\{\xi_n^*\}$.

We will now state the primal problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}(\xi_n + \xi_n^*) \quad \text{subject to} \quad \begin{cases} \forall n : (\mathbf{w}^T\mathbf{x}_n - b) - t_n \leq \epsilon + \xi_n \\ \forall n : t_n - (\mathbf{w}^T\mathbf{x}_n - b) \leq \epsilon + \xi_n^* \\ \forall n : \xi_n \geq 0 \\ \forall n : \xi_n^* \geq 0 \end{cases}$$

Text

Introduce Lagrange multipliers for the constraints and write down the primal Lagrangian. Use the following notation: $\{\alpha_i\}$ and $\{\alpha_i^*\}$ are the Lagrange multipliers for the first two constraints and $\{\mu_i\}$ and $\{\mu_i^*\}$ for the last two constraints.

1.5 pts · Open · 1/2 Page

$$\begin{aligned} L(\mathbf{w}, b, \{\xi_n\}, \{\xi_n^*\}, \{\alpha_n\}, \{\alpha_n^*\}, \{\mu_n\}, \{\mu_n^*\}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}[\xi_n + \xi_n^*] \\ + \sum_{n=1}^{N}[\alpha_n(y(\mathbf{x}_n) - t_n - \epsilon - \xi_n)] \\ + \sum_{n=1}^{N}[\alpha_n^*(t_n - y(\mathbf{x}_n) - \epsilon - \xi_n^*)] \\ - \sum_{n=1}^{N}[\mu_n\xi_n] - \sum_{n=1}^{N}[\mu_n^*\xi_n^*] \end{aligned}$$ 1.5 pts

For each incorrect sign or incorrect term -0.5 pts

For each incorrect sign or incorrect term -0.5 pts

Write down all KKT conditions.

1.5 pts · Open · 9/20 Page

Model answer

Primal feasibility:

$$\forall n : -y(\mathbf{x}_n) + t_n + \epsilon + \xi_n \geq 0$$
$$\forall n : -t_n + y(\mathbf{x}_n) + \epsilon + \xi_n^* \geq 0$$
$$\forall n : \xi_n \geq 0$$
$$\forall n : \xi_n^* \geq 0$$

Dual feasibiility:

$$\forall_n : \alpha_n \geq 0 \ , \ \alpha_n^* \geq 0 \ , \ \mu_n \geq 0 \ , \ \mu_n^* \geq 0$$

Complementary slackness

$$\forall n : (-y(\mathbf{x}_n) + t_n + \epsilon + \xi_n)\alpha_n = 0$$
$$\forall n : (-t_n + y(\mathbf{x}_n) + \epsilon + \xi_n^*)\alpha_n^* = 0$$
$$\forall n : \xi_n \mu_n = 0$$
$$\forall n : \xi_n^* \mu_n^* = 0$$

For all four (for each n) primal feasibility conditions	0.5 pts
For all four (for each n) dual feasibility conditions	0.5 pts
For all four (for each n) complimentary slackness conditions	0.5 pts

Derive the stationary conditions (by computing $\partial \ell / \partial \rho = 0$, where ℓ is the primal Lagrangian and ρ is a primal variable).

2 pts · Open · 1/2 Page

for correct derivation of $\frac{\partial L}{\partial \mathbf{w}} = 0$:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} + \sum_{n=1}^{N} \alpha_n \mathbf{x}_n - \sum_{n=1}^{N} \alpha_n^* \mathbf{x}_n$$

0.5 pts

for correct derivation of $\frac{\partial L}{\partial b} = 0$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} \alpha_n + \sum_{n=1}^{N} \alpha_n^*$$

0.5 pts

minus sign difference is OK because of inconsistency in defining y and how it is given in the constraints

for correct derivation of $\frac{\partial L}{\partial \xi_n} = 0$

$$\frac{\partial L}{\partial \xi_n} = C - \alpha_n - \mu_n$$

0.5 pts

for correct derivation of $\frac{\partial L}{\partial \xi_n^*} = 0$

$$\frac{\partial L}{\partial \xi_n} = C - \alpha_n^* - \mu_n^*$$

0.5 pts

Define the dual Lagrangian (no need to derive it) and explain how you could obtain it using your results from (c).

1.5 pts · Open · 1/2 Page

The dual Lagrangian L^* is obtained by minimizing the Lagrangian of 4a with respect to the primal variables. I.e.,

$$L^*(\{\alpha_n\}, \{\alpha_n^*\}, \{\mu_n\}, \{\mu_n^*\}) = \min_{\mathbf{w}, b, \{\xi_n\}, \{\xi_n^*\}} L(\mathbf{w}, b, \ldots)$$

1 pt

This can be done by eliminating the primal variables from L by using/substituting the conditions of 4c.

0.5 pts

Suppose we work with high dimensional features (large D) and notice that the regression model is very flexible / expressive, in the sense that it could easily overfit. We can control how well the model generalizes by tuning C. *Mark the correct answer*. In order to *prevent overfitting*, C should be

1 pt · Multiple choice · 2 alternatives

○ large	0.0
◉ small	1.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Give your reasoning behind your answer to the preceding question (for increasing or decreasing C).

1 pt · Open · 7/20 Page

C penalizes slack. If the data is complicated (possibly lot of noise and outliers) the
If C is very large you hardly give any slack and the function tries to follow the data as good as possible (even in the precense of noise and outliers) and this may lead to overfitting.
Conversely, with small C you cut more slack and the model will be less sensitive to outliers, and thus less chance of overfitting. 1 pt

Alternatively you could argue that low weights means smoother solutions (recall L2 regularization/ridge regression). With a small C the ||w|| component will be more important.

Neural Networks #28377399

12.5 pts · Last updated 9 Jan, 2024, 15:05 · Saved in Final Exam

Consider a neural network with two hidden layers and a skip connection (aka residual connection):

$$\begin{aligned} f(\mathbf{x}; \mathbf{w}_0, w_1, w_2) &= w_2 \cdot h_2 + h_1 \\ h_2 &= \sigma(w_1 \cdot h_1) \\ h_1 &= \sigma(\mathbf{w}_0^T \mathbf{x}) \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^D$ is a D-length (column) vector, $\mathbf{w}_0 \in \mathbb{R}^D$ are the first-layer weights, $w_1 \in \mathbb{R}$ is the second-layer weight, and $w_2 \in \mathbb{R}$ is the hidden-to-output weight.

Text

Write down how the chain rule is implemented to compute the derivative $\partial f / \partial \mathbf{w}_0$.. *Show all partial derivatives involved to the finest granularity allowed*.

For example, if $g(a; b, c) = (a \cdot b)/c$, then $\partial g / \partial a = (\partial g / \partial (a \cdot b))(\partial (a \cdot b)/\partial a)$. Feel free to define any intermediate computations, such as $a \cdot b$ in the example, with a variable.

2 pts · Open · 1/2 Page

Model answer

Define $a_1 = \mathbf{w}_0^T \mathbf{x}$ and $a_2 = w_1 \cdot h_1$.

Then the chain-rule gives us

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{w}_0} &= w_2 \frac{\partial h_2}{\partial \mathbf{w}_0} + \frac{\partial h_1}{\partial \mathbf{w}_0} \\ &= w_2 \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial \mathbf{w}_0} + \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial \mathbf{w}_0} \end{aligned}$$

or simplify one step further:

$$= \left(w_2 \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} + 1 \right) \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial \mathbf{w}_0}$$

For correct answer	2 pts
When not applying the chain-rule over the sigmoid activation ($\frac{\partial h}{\partial a}$)	-1 pts

Based on your prevoius answer, give the exact expression for the derivative $\partial f / \partial w_{0j}$, where w_{0j} denotes the jth component of $\mathbf{w}_0$, by evaluating all partial derivatives. Let $\sigma(\cdot)$ denote a logistic activation function

$$\sigma(z) = 1/(1 + \exp\{-z\})$$

and assume that the activations are logistic functions from this question forward.

To return to the above example, the derivative explicitly gives $\partial g / \partial a = (1/c)(b) = b/c$.

2 pts · Open · 9/20 Page

Model answer

$$= w_2 h_2 (1 - h_2) w_1 h_1 (1 - h_1) x_j + h_1 (1 - h_1) x_j$$

or

$$= (w_2 h_2 (1 - h_2) w_1 + 1) h_1 (1 - h_1) x_j$$

Same expression as above however replace all instances of $\mathbf{w}_0$ with w_{0j}.

Correct computation or expression for derivative of sigmoid

$$\sigma'(a) = \sigma(a)(1 - \sigma(a))$$

and thus 1 pt

$$\frac{\partial h_1}{\partial a_1} = h_1(1 - h_1)$$

and

$$\frac{\partial h_2}{\partial a_2} = h_2(1 - h_2)$$

$\frac{\partial a_2}{\partial h_1} = w_1$ 0.5 pts

$\frac{\partial a_1}{\partial w_{0j}} = x_j$ 0.5 pts

Write down the loss function for performing *ridge regression* using this neural network. Assume we observe a training dataset containing features $\{\mathbf{x}_n\}_{n=1}^N$, with $\mathbf{x}_n \in \mathbb{R}^D$, and corresponding responses $\{t_n\}_{n=1}^N$, with $t_n \in \mathbb{R}$, where N is the number of data points and D is the feature dimensionality.

2 pts · Open · 3/10 Page

Recognizing that ridge regression is least squares loss + squared penalty for each weight 1 pt

$loss = \sum_{n=1}^{N} \left[(f(\mathbf{x}_n; \mathbf{w}_0, w_1, w_2) - t_n)^2 \right] + \lambda(\mathbf{w}_0^T \mathbf{w}_0 + w_1^2 + w_2^2)$ 1 pt

When missing λ -0.5 pts

When missing a weight in the expression -0.5 pts

The ridge regression loss could also be derived from the Bayesian modeling principles. The Bayesian modeling viewpoint entails that we solve our problem using prior beliefs we might have in our problem. To obtain ridge regression from this point of view, 1) *what* kind of distribution should the neural network (NN) parametrize, and 2) *how* should the NN parametrize it? Additionally, 3) are there any other distributions that need to be modeled? 4) How then is the ridge regression loss obtained this probabilistic model?

3 pts · Open · 7/10 Page

1) The network parametrizes a Gaussian predictive distribution.	0.75 pts
2) The neural network models the mean of the Gaussian. The standard deviation is fixed.	0.75 pts
3) Also, **a Gaussian prior distributions** needs to be chosen for each weight, with zero mean and a standard deviation inversely proportional to λ.	0.75 pts
4) Then minimizing the **log-posterior** is equivalent to minimizing the ridge loss. Namely, the log-posterior results in a squared loss (from the log-likelihood) and the regularizing term (from the log-prior).	0.75 pts

Is the ridge regression loss for the described neural network convex with respect to its parameters?

1 pt · Multiple choice · 2 alternatives

○ Yes	0.0
◉ No	1.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Consider making w_2 a constant such that its value can *only* be zero: $w_2 = 0$. In other words, gradient descent would not change its value from zero. Under this assumption, to which of the following models is this neural network now equivalent?:

1 pt · Multiple choice · 4 alternatives

○ Linear regression	0.0
○ Kernel regression with the kernel determined by the neural network's hidden units.	0.0
◉ **Logistic regression**	1.0
○ Linear classifier with 3 or more classes	0.0

Feedback

Feedback when the question is answered correctly

Feedback when the question is answered partially correctly

Feedback when the question is answered incorrectly

Neural networks are typically optimized using stochastic gradient descent (SGD), as opposed to full-batch gradient descent (GD). Give two advantages of using SGD vs full-batch GD.

1.5 pts · Open · 1/2 Page

SGD is less likely to get stuck in local minima	0.75 pts
SGD is faster to compute than full-batch GD.	0.75 pts