SOLUTIONS

CS689: Machine Learning - Fall 2019

Final Exam

Dec 13, 2019 Name: _____

Instructions: Write your name on the exam sheet. The duration of this exam is **two hours**. No electronic devices may be used during the exam. You may consult either your notes or the Murphy text during the exam (you must choose one). Up to a three inch stack of notes is permitted. No other sources are permitted. Sharing of notes/texts during the exam is strictly prohibited. Show your work for all derivation questions. Provide answers that are as detailed as possible for explanation questions. Attempt all problems. Partial credit may be given for incorrect or incomplete answers. If you need extra space for answers, write on the back of the **preceding** page. If you have questions at any time, raise your hand.

Problem	Topic	Page	Points	Score
1	Loss Functions	1	10	
2	Maximum Likelihood	2	10	
3	Generalized Linear Regression	3	10	
4	Logistic Regression	4	10	
5	Lagrangian Duality	5	10	
6	Mixture Models	6	10	
7	Factor Analysis and Autoencoders	7	10	
8	Bayesian Inference	8	10	
9	Experiment Design	9	10	
10	Applications	10	10	
Total:			100	

1. (*10 points*) **Loss Functions** Suppose we have a classification problem where $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y} = \{1, ..., C\}$. Suppose that f is a prediction function satisfying $f : \mathcal{X} \to \mathcal{Y}$. Under what circumstances would it be sensible to assess the performance of f using the squared loss? When would it not be sensible? Explain your answers.

* For a classification problem, we generally care about the classification error or zero-one loss.

* In the case where C=2, the possible class labels are {1, 2} and the resulting values of squared loss and zero-one loss are given below.

y	$f(x)$	$y \neq f(x)$	$(y - f(x))^2$
1	1	0	0
1	2	1	1
2	1	1	1
2	2	0	0

* Thus, squared loss and zero-one loss exactly agree for C=2 and using squared loss is sensible.

* For C>2 the squared loss is larger when the difference in label values is larger, even though the label values are not meaningful. This is not sensible.

2. (*10 points*) **Maximum Likelihood:** Suppose we have a binary probabilistic classifier $P_\theta(Y = y|X = x)$. Given a data set of feature vectors $\mathcal{D} = \{x_n | 1 \leq n \leq N\}$, we are interested in modeling the distribution of the classifier output probabilities $\pi_n = P_\theta(Y = 1|X = x_n)$ using a Beta distribution $P(\Pi = \pi|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\pi^{a-1}(1-\pi)^{b-1}$.

(a) (5 points) What is the log likelihood of the model $P(\Pi = \pi|a, b)$ given the observations $\pi_1, ..., \pi_N$?

Let $\mathcal{D}' = \{\pi_1, ..., \pi_N\}$

$$\mathcal{L}(\mathcal{D}'|a,b) = \sum_{n=1}^{N} \log P(\pi = \pi_n|a,b)$$

$$= \sum_{n=1}^{N} \Big(\log\Gamma(a+b) - \log\Gamma(a) - \log\Gamma(b) + (a-1)\log\pi_n + (b-1)\log(1-\pi_n) \Big)$$

(b) (5 points) The *digamma function* $\psi(x)$ is a special function $\psi(x) = \frac{d\log\Gamma(x)}{dx}$ that provides the derivative of the log of the gamma function $\Gamma(x)$. Use this fact to derive the gradient vector for the model $P(\Pi = \pi|a, b)$. Show your work.

$$\nabla_a \mathcal{L}(\mathcal{D}'|a,b) = \sum_{n=1}^{N} \Big(\psi(a+b) - \psi(a) + \log\pi_n \Big)$$

$$\nabla_b \mathcal{L}(\mathcal{D}'|a,b) = \sum_{n=1}^{N} \Big(\psi(a+b) - \psi(b) + \log(1-\pi_n) \Big)$$

$$\nabla \mathcal{L}(\mathcal{D}'|a,b) = \Big[\nabla_a \mathcal{L}(\mathcal{D}'|a,b),\ \nabla_b \mathcal{L}(\mathcal{D}'|a,b) \Big]$$

3. (*10 points*) **Generalized Linear Regression:** Consider the probabilistic linear regression model $P(Y = y|\mathbf{X} = \mathbf{x}) = \mathcal{N}(y; \mathbf{w}^T\mathbf{x} + b, \sigma^2)$. This model asserts that the variance is constant for all $\mathbf{x}$. Suppose we are interested in generalizing the model so that the variance is a function of $\mathbf{x}$ as well.

(a) (5 points) Describe how you could modify the model to also make the variance data-dependent. Provide an updated form for $P(Y = y|\mathbf{X} = \mathbf{x})$.

The variance is constant because σ^2 does not depend on x. To fix this we need to set $\sigma^2 = f(x)$ for some function f. Since $\sigma^2 > 0$, we need $f(x) > 0$ for all x. We will let $f(x) = \exp(g(x))$ for $g: \mathbb{R}^D \to \mathbb{R}$. For example, an MLP with a linear output layer. We have: $P(y|x) = \mathcal{N}(y; w^Tx+b, f(x))$.

(b) (5 points) Suppose we only care about the mean squared error that the learned model you described in part (a) achieves when making predictions on test data. Do we need the learned data-dependent variance component of the model when making predictions? Explain your answer.

No. To make a prediction, we typically output the mean of the predictive distribution $\mathbb{E}_{P(y|x)}[y]$. For a normal model this is just μ. For the linear gaussian model it is w^Tx+b, which does not depend on the variance at all.

4. (*10 points*) **Logistic Regression Equivalence:** The binary logistic regression model with labels $y \in \{-1, 1\}$ and features $\mathbf{x} \in \mathbb{R}^D$ can be written as

$$P(Y = y|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + \exp(-y(\mathbf{w}^T\mathbf{x} + b))}$$

using one weight vector $\mathbf{w} \in \mathbb{R}^D$ and one bias b. On the other hand, the multi-class logistic regression model with $y \in \{1, 2, ..., C\}$ can be written as

$$P(Y = y|\mathbf{X} = \mathbf{x}) = \frac{\exp(\mathbf{w}_y^T\mathbf{x} + b_y)}{\sum_{y'=1}^{C} \exp(\mathbf{w}_{y'}^T\mathbf{x} + b_{y'})}$$

using one weight vector $\mathbf{w}_y$ and one bias b_y *per class*. Despite their apparent differences, show that the two models are exactly equivalent in the case where $C = 2$.

We will show that the multi-category case with $C=2$ reduces to the binary case.

$$P(y=2|x) = \frac{\exp(w_2^T x + b_2)}{\exp(w_1^T x + b_1) + \exp(w_2^T x + b_2)}$$

We divide the top and bottom by $\exp(w_2^T x + b_2)$ to get.

$$= \frac{1}{1 + \exp((w_1 - w_2)^T x + (b_1 - b_2))}$$

Now let $\hat{w} = w_2 - w_1$, $\hat{b} = b_2 - b_1$. We have:

$$= \frac{1}{1 + \exp(-(\hat{w}^T x + \hat{b}))}$$

If we map $y=1$ to $\hat{y}=-1$ and $y=2$ to $\hat{y}=1$, we have

$$P(Y=y|X=x) = P(\hat{Y}=\hat{y}|X=x) = \frac{1}{1+\exp(-\hat{y}(\hat{w}^T x + \hat{b}))}$$

5. (*10 points*) **Lagrangian Duality** Consider the alternate optimization problem for linear regression given below:

$$\arg\min_{\mathbf{w},\eta} \sum_{n=1}^{N} \eta_n^2$$
$$\text{s.t. } \forall n,\ \eta_n = y_n - (\mathbf{w}^T \mathbf{x}_n)$$
$$\|\mathbf{w}\|_2^2 \leq C$$

(a) (5 points) Write down the Lagrangian function for this optimization problem.

In standard form the constraints are $\eta_n - (y_n - w^T x_n) = 0$ and $C - \|w\|_2^2 \geq 0$. The Lagrangian is thus

$$\ell(w, \eta, \lambda, \gamma) = \sum_{n=1}^{N} \eta_n^2 - \sum_{n=1}^{N} \lambda_n \left(\eta_n - (y_n - w^T x_n)\right) - \gamma \left(C - \|w\|_2^2\right)$$

(b) (5 points) Derive the Lagrange dual for this optimization problem. Show your work.

$$q(\lambda, \gamma) = \min_{w, \eta} \ell(w, \eta, \lambda, \gamma)$$

$$\nabla_w \ell(w, \eta, \lambda, \gamma) = -\sum_{n=1}^{N} \lambda_n x_n + 2\gamma w = 0$$

$$\Rightarrow w = \sum_{n=1}^{N} \frac{\lambda_n x_n}{2\gamma}$$

$$\frac{\partial \ell(w, \eta, \lambda, \gamma)}{\partial \eta_n} = 2\eta_n - \lambda_n \Rightarrow \eta_n = \frac{\lambda_n}{2}$$

$$q(\lambda, \gamma) = \sum_{n=1}^{N} \frac{\lambda_n^2}{4} - \sum_{n=1}^{N} \lambda_n \left(\frac{\lambda_n}{2} - \left(y_n - \left(\frac{1}{2\gamma} \sum_{n'=1}^{N} \lambda_{n'} x_{n'}\right)^T x_n\right)\right)$$

$$- \gamma \left(C - \sum_{n=1}^{N} \sum_{n'=1}^{N} \lambda_n \lambda_{n'} x_{n'}^T x_n\right)$$

6. (*10 points*) **Mixture Models** Suppose we have a data set $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ consisting of data vectors $\mathbf{x}_n = [x_{1n}, ..., x_{Dn}]$ where each data dimension represents an integer count.

(a) (5 points) Describe how we could model the distribution $P(\mathbf{X} = \mathbf{x})$ as a mixture model. Be specific and provide equations to support your description.

We can use a mixture of products of count distributions such as geometric or poisson.

We have:

$$P(z=k) = \theta_k$$

$$P(x=x \mid z=k) = \prod_{d=1}^{D} P(x_d = x_d \mid z=k) = \prod_{d=1}^{D} (1-\pi_{dk})^{x_d} \pi_{dk} \quad \text{(geometric)}$$

$$P(x=x) = \sum_{k=1}^{K} P(x=x \mid z=k) P(z=k) = \sum_{k=1}^{K} \theta_k \cdot \prod_{d=1}^{D} (1-\pi_{dk})^{x_d} \pi_{dk}$$

(b) (5 points) Suppose we want to learn the model using direct numerical optimization. Provide the specific objective function you would use and explain your answer.

Since the z variables are not observed, we must maximize the log marginal likelihood:

$$\mathcal{L}(D \mid \theta, \pi) = \sum_{n=1}^{N} \log P(x_n = x_n)$$

$$= \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \theta_k \prod_{d=1}^{D} (1-\pi_{dk})^{x_{dn}} \pi_{dk} \right)$$

7. (*10 points*) **Factor Analysis and Autoencoders**

(a) (5 points) Consider the factor analysis model is given by $P(\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$ and $P(\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{w}^T\mathbf{z}, \Psi)$ where Ψ is a positive definite matrix. What additional restrictions are needed on Ψ for this model to learn a useful latent representation for the data? Explain your answer.

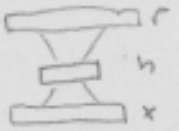
We need to restrict Ψ to be a diagonal matrix.

The reason is that this will force the structure in the data to be explained by the latent factors z. Otherwise, Ψ itself can model the covariance in the data. Setting $\Psi = \sigma^2 I$ yields PCA and will also result in learning useful structure.

(b) (5 points) Explain why there is no modeling benefit to using a linear autoencoder with more than one hidden layer.

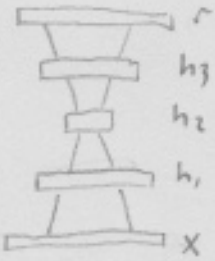
* In a 1 layer linear autoencoder we have:

$h = W^T x$; $r = V^T h$



* If we add an extra set of layers we have:

$h_1 = W_1^T x$; $h_2 = W_2^T h_1$; $h_3 = V_2^T h_2$; $r = V_1^T h_3$



* Since the layers are linear, we have:

$h_2 = W_2^T W_1^T x$ and $r = V_1^T V_2^T h_2$

$= \hat{W} x$ $= \hat{V} h_2$

* Thus, adding extra layers does not increase representational capacity.

8. (*10 points*) **Bayesian Inference:** Suppose that $x \in \mathbb{N}$ is a count random variable that follows the negative Binomial distribution:

$$P(X = x|\theta, r) = \binom{x+r-1}{x} \cdot (1-\theta)^r \theta^x$$

We use a Beta prior distribution as defined below. Use this information to answer the following questions.

$$P(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

(a) (5 points) Given data $\mathcal{D} = \{x_n|n = 1, ..., N\}$, what is the likelihood of $\mathcal{D}$ given θ? Explain your answer and simplify as much as possible.

* $P(D|\theta) = \prod_{n=1}^{N} P(X=x_n|\theta, r) = \prod_{n=1}^{N} \binom{x_n+r-1}{x_n} (1-\theta)^r \theta^{x_n}$

$$= \prod_{n=1}^{N} \binom{x_n+r-1}{x_n} \cdot (1-\theta)^{Nr} \cdot \theta^{\sum_{n=1}^{N} x_n}$$

* The likelihood is the product over the data cases of the probability of each case given θ.

(b) (5 points) Given data $\mathcal{D} = \{x_n|n = 1, ..., N\}$, what is the posterior distribution of θ given $\mathcal{D}$? Show your work and explain your answer.

* $P(\theta|D) \propto P(D|\theta)P(\theta)$

$$\propto (1-\theta)^{Nr} \cdot \theta^{\sum_{n=1}^{N} x_n} \cdot \theta^{\alpha-1} \cdot (1-\theta)^{\beta-1}$$

$$= (1-\theta)^{\beta+Nr-1} \cdot \theta^{\alpha+\sum_{n=1}^{N} x_n - 1}$$

$$\Rightarrow \text{Beta}\left(\alpha + \sum_{n=1}^{N} x_n, \beta + Nr\right)$$

* The posterior distribution is proportional to the likelihood times the prior. We expand, drop constants and identify the form of posterior.

9. (*10 points*) **Experiment Design:** Suppose we have a data set containing labeled instances $(\mathbf{x}_{tn}, y_{tn})$ for a collection of individuals $n = 1, ..., N$ and time points $t = 1, ..., T$.

(a) (5 points) If we want to assess how well a supervised model can perform when applied to a future set of time points for the same individuals, what experimental design should we use? Explain your answer.

We first need to train the model. We then need to evaluate it. The most basic design is train-test. Since we want to evaluate generalization to future times, we need to simulate this using the data we have. We can pick a value $T_0 < T$ and as the training set use $D_{tr} = \{(x_{tn}, y_{tn}) \mid t \leq T_0, 1 \leq n \leq N\}$. As the test set we use $D_{te} = \{(x_{tn}, y_{tn}) \mid t > T_0, 1 \leq n \leq N\}$

(b) (5 points) Suppose the data set contains many features and we start by using all available data vectors $\mathbf{x}_{tn}$ to learn and apply a dimensionality reduction model, obtaining lower-dimensional feature vectors $\mathbf{x}'_{tn}$. We then apply the procedure you described in part (a) to a data set of $(\mathbf{x}'_{tn}, y_{tn})$ instances. Is the end-to-end experiment design still correctly assessing how well this approach can perform when applied to data from new time points? Explain your answer.

No. The supervised model depends on the values x'_{tn}, and x'_{tn} are inferred using the complete data set. It is thus using information from the test set during training. If the distribution of x changes through time, this procedure would underestimate the error.

10. (*10 points*) **Applications:** Suppose we have a data set of completely observed real-valued vectors $\mathcal{D} = \{\mathbf{x}_n | n = 1, ..., N\}$ where $\mathbf{x}_n \in \mathbb{R}^D$. Our goal is to learn a **dimensionality reduction** model for these data. However, at deployment time, we expect to encounter incomplete data vectors (vectors with missing entries) and we must still be able to produce reduced dimensional representations for them. Describe a solution for modeling these data that **does not** require an auxiliary imputation model. Your answer should cover (1) a description of your model, (2) your approach to learning the model, and (3) how the model is applied to incomplete data cases. Provide supporting equations and justify your choices.

① We can use a factor analysis model for this task. The model is $P(z) = N(z; 0, I)$, $P(x|z) = N(x; W^T z, \Psi)$ where Ψ is a diagonal matrix and W is the factor loading matrix. We make this choice because the model can directly support missing data.

② We can learn the model as usual from complete data using direct marginal likelihood maximization (marginalizing over z).

③ Given an incomplete data instance x, we need to compute $P(z = z | x^o = x^o)$. We can get this using:

$$P(z = z | x^o = x^o) = \frac{P(x^o = x^o | z = z) P(z = z)}{\int P(x^o = x^o | z = z) P(z = z) \, dz}$$

$$= \frac{\int P(x = x | z = z) \, dx^m \cdot P(z = z)}{\int \int P(x = x | z = z) \, dx^m \cdot P(z = z) \, dz}$$