Name: _____ NetID: _____

S&DS 355 / 555

Introductory Machine Learning

Midterm Exam #2, Sample Solution

Tuesday, November 19, 2019

Complete all of the problems. You are allowed one double-sided (8.5×11) sheet of paper with notes. No electronic devices, including calculators. You have 75 minutes to complete the exam.

For your reference, recall the following distributions:

$$\text{Normal}(\mu, \sigma^2): \quad p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}, \quad x \in \mathbb{R}$$

$$\text{Beta}(\alpha, \beta): \quad p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}, \quad \theta \in [0,1]$$

$$\text{Dirichlet}(\alpha_1, \ldots, \alpha_K): \quad p(\theta) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \theta_k^{\alpha_k - 1}, \quad \sum_{k=1}^K \theta_k = 1, \;\; \theta_k \geq 0$$

$$\text{Bernoulli}(\theta): \quad \mathbb{P}(z) = \theta^z (1-\theta)^{(1-z)}, \quad z \in \{0, 1\}$$

And the commonly used activation functions for neural networks:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad \text{sigmoid}(x) = \frac{e^x}{1+e^x}, \quad \text{ReLU}(x) = \max(x, 0), \quad \text{identity}(x) = x$$

1	2	3	4	5	total
20	10	10	10	10	60

1. *True or False, Yes or No?* (20 points)

   Indicate the *best* answer to each of the following statements.

   In a class-based bigram language model, the probability of the next word is

$$p(w_n \mid w_{n-1}) = p(w_n \mid c(w_n))\, p(c(w_n) \mid c(w_{n-1}))$$

   where $c(w)$ is the class of word w.

   In an embedding-based bigram language model, the probability of the next word is

$$p(w_n \mid w_{n-1}) = \frac{\exp(\phi(w_n)^T \phi(w_{n-1}))}{\sum_{v \in \text{Vocabulary}} \exp(\phi(v)^T \phi(w_{n-1}))}$$

   where $\phi(w) \in \mathbb{R}^{100}$ is embedding vector for word w.

YES [NO] (a) As the number of classes decreases, the language model's perplexity on test data will generally decrease.

YES [NO] (b) For words to have similar embedding vectors, they must co-occur often.

YES [NO] (c) The embedding vectors can be obtained by bottom up clustering.

YES [NO] (d) Each word can belong to multiple word classes.

YES [NO] (e) The embedding vectors are chosen to solve common word analogies.

The following questions concern latent Dirichlet allocation topic model, using the usual terminology where θ_d are the per-document topic proportions (with Dirichlet prior), $Z_{d,n}$ are the per-word topic assignments, and β_k are the topics (with Dirichlet prior).

[YES] NO (a) A goal of topic modeling is to automatically find the major semantic themes in a corpus of documents

YES [NO] (b) This topic model is commonly used as a language model to predict the next word.

YES [NO] (c) The documents in a corpus are generated independently under this model.

YES [NO] (d) In practice, reordering the words in each document in the corpus gives a different model.

[YES] NO (e) Conditioned on all of the topic assignments $Z_{d,n}$, the posterior distribution over each topic β_k is a single Dirichlet distribution.

The following questions concern general Bayesian inference.

(a) Bayesian inference involves three things: a model for assigning a likelihood to data, a prior distribution over the parameters of the model, and a method for computing the conditional probability of those parameters after observing data, using Bayes rule.

YES [NO]

(b) Bayesian inference cannot be used for logistic regression, because it is a discriminative model.

[YES] NO

(c) Bayesian inference is based on the entire posterior distribution of the parameters, rather than estimating a specific value for those parameters.

[YES] NO

(d) The posterior mean is less affected by the choice of prior as the sample size increases.

YES [NO]

(e) Under a Bayesian treatment of models, the parameters θ are fixed numbers.

The following questions concern basic feedforward neural networks.

[YES] NO (a) If the activation function $\sigma(x) = x$ is the identity, a neural network is equivalent to a linear model.

YES [NO] (b) The number of parameters in a neural network is proportional to the total number of neurons.

[YES] NO (c) Training a neural network for classification uses the same objective function on the output as is used for classical logistic regression.

YES [NO] (d) The hidden neurons can be seen as latent variables in a probabilistic model.

YES [NO] (e) Training the network automatically determines the optimal network architecture.

2. *Bayesian inference* (10 points)

Suppose X is a random variable corresponding to a roll of a 6-sided die, with probability $\theta = (\theta_1, \theta_2, \ldots, \theta_6)$ that each of the six faces comes up on any toss. We observe data $D_n = \{x_1, \ldots, x_n\}$ where $x_i \in \{1, 2, 3, 4, 5, 6\}$, and the rolls are independent. Suppose the prior distribution on θ is Dirichlet$(\alpha, \alpha, \alpha, \alpha, \alpha, \alpha)$ where $\alpha > 0$. Let $s_k = \sum_{i=1}^{n} \mathbb{1}(x_i = k)$ be the number or rolls that land on k.

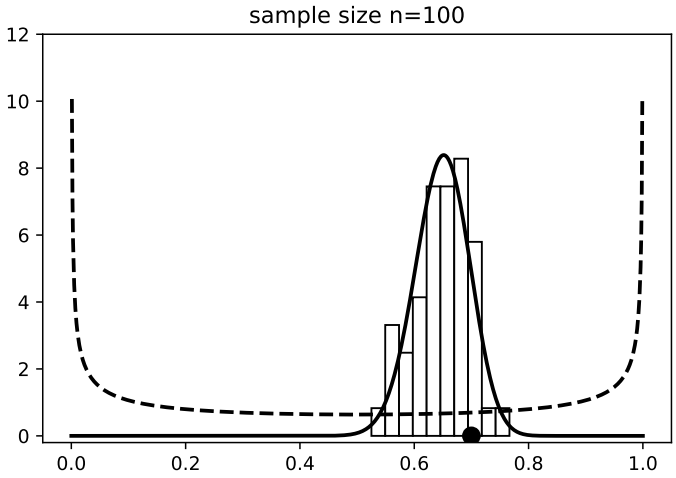
(a) What is the probability (likelihood) $\mathbb{P}((x_1, x_2, x_3, x_4) = (5, 3, 6, 3) \,|\, \theta)$?

$\theta_3^2 \, \theta_5 \, \theta_6$

(b) What is the posterior distribution of θ given D_n?

Dirichlet$(\alpha + s_1, \alpha + s_2, \alpha + s_3, \alpha + s_4, \alpha + s_5, \alpha + s_6)$

(c) In the demo of Bayesian inference in class, we showed "movies" where one frame looked like this:



Briefly describe what each of the following elements of this plot is showing:

(1) The horizontal axis

The parameter θ

(2) The black dot at 0.7

The parameter for the true model

(3) The dashed curve that rises up to around 10 at the endpoints 0.0 and 1.0

The prior distribution, Beta$(1/2, 1/2)$

(4) The histogram

A random sample from the posterior

(5) The solid bell-shaped curve

The posterior distribution

3. *Language models and embeddings* (10 points)

(a) Define the pointwise mutual information.

$$\text{PMI}(w_1, w_2) = \log\left(\frac{p_{\text{near}}(w_1, w_2)}{p(w_1)\,p(w_2)}\right)$$

where $p_{\text{near}}(w_1, w_2)$ is the probability that words w_1 and w_2 appear near each other in the corpus.

(b) Define the perplexity of a language model, and describe in words what it means.

Perplexity $= p(w_1, w_2, \ldots, w_N)^{-1/N}$

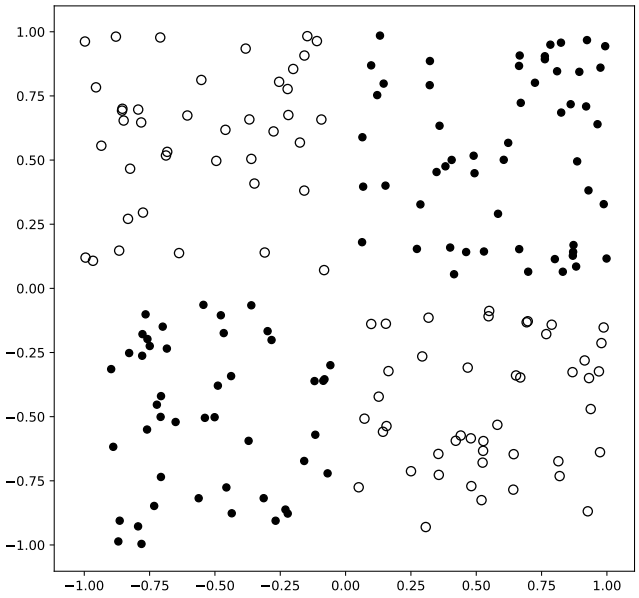
This can be interpreted as the average number of equally likely words that the model predicts. For example, if the perplexity is 100, it means that effectively there are about 100 words that have probability of order $1/100$.

(c) Use $\phi(\text{w})$ to denote the embeddings of words. Suppose we wish to find the three words w that are most similar to "yale" in an embedding model. Write down the solution as an optimization problem. Note that part of the problem is for you to define what is meant by "most similar."

Sort the numbers $v(w) = \|\phi(w) - \phi(\text{yale})\|$ in increasing order, for words w in the vocabulary. The word yale will have value $v(\text{yale}) = 0$. Take the next three words.

4. *Neural nets* (10 points)

Consider a function $f(x) = w_2\sigma(w_{11}x_1 + w_{12}x_2 + b_1) + b_2$ where σ is an activation function and $x = (x_1, x_2)$ is a 2-dimensional input. So, $w_{11}, w_{12}, w_2, b_1, b_2$ are numbers. This can be thought of as a neural network with a single hidden layer, consisting of one neuron. It gives a binary classification model by $\mathbb{P}(Y = 1 \mid x) = \dfrac{e^{f(x)}}{1 + e^{f(x)}}$. Suppose that we have a data set that looks like this:



The open white points are labeled $Y = 0$ and the solid black points are labeled $Y = 1$. For this problem, you are asked to consider whether or not this simple neural network will give an accurate classifier. *For the purpose of this problem, we will call a classifier "accurate" if the training error rate is less that* 1%.

(a) Can this data set be accurately classified with a linear decision rule? Give a brief justification for your answer.

No. Any way that you draw a line through the square, there will be a mix of white and black points on both sides of the line.

(b) Suppose the neural network is trained four different ways, using the activation functions ReLU, tanh, identity, and sigmoid. Which of these four neural networks will give accurate classifiers? Briefly explain your answer.

None of the four neural nets will give an accurate classifier. The decision boundary will still be linear. The activation function just controls how the probability falls off from the line $p(Y = 1 \,|\, x) = \frac{1}{2}$. You can try this out on `playground.tensorflow.org`.

(c) Suppose the neural network is now changed to have input $x_1 x_2$; so the function is now $\widetilde{f}(x) = w_2 \sigma(w_1 x_1 x_2 + b_1) + b_2$ with parameters w_1, b_1, w_2, b_2. Four new networks are trained, using each of the activation functions ReLU, tanh, identity, and sigmoid. Which of the resulting four neural networks will give accurate classifiers? Briefly explain your answer

All four neural nets will give an accurate classifier. The optimal classification rule can be seen from the scatter plot of the data: $\widehat{Y}(x) = 1$ if $\text{sign}(x_1 x_2) = 1$ and $\widehat{Y} = 0$ otherwise. A single neuron with input $x_1 x_2$ can model this accurately. You can try this out on `playground.tensorflow.org`.

5. *Code* (10 points)

(a) What is the value of the following Python expression?

```
np.tanh([x for x in np.linspace(-10, 10, 3)])
```

$[-1, 0, 1]$ or $[\tanh(-10), 0, \tanh(10)]$ $(\tanh(\pm 10) \approx 1 \mp 10^{-8})$

(b) Explain in a couple of sentences what the function `mystery` defined below does:

```
def mystery(X, Y):
  n, d = X.shape
  W = np.random.randn(d,1)
  b = np.random.randn(1,1)

  for i in range(10000):
      f = np.dot(X, W) + b
      phat = np.exp(f)/(1+np.exp(f))
      dloss = phat - Y
      dloss /= num_examples
      dW = np.dot(X.T, dloss)
      db = np.sum(dloss)
      W += -.1 * dW
      b += -.1 * db
  return W, b
```

The function returns the maximum likelihood estimates of a linear logistic regression model, trained using gradient descent.