CPSC 340 Final (Fall 2015)

Name:

Student Number:

Please enter your information above, turn off cellphones, space yourselves out throughout the room, and wait until the official start of the exam to begin. You can raise your hand to ask a question, and please look up occasionally in case there are clarifications written on the projector (the time remaining will also be written on the projector). You are welcome to (quietly) leave early if you finish early.

The final consists of 9 questions, and they will all be equally weighted in the marking scheme. Note that some question have multiple parts, written as **(a)**, **(b)**, **(c)** and **(d)**. *Clearly mark* where you are answering each part, *show your work/reasoning for each part*, and make sure to check that you answered all parts before handing in your final.

Good luck!

1 Training/Validation/Testing

You are asked by a client to build a system that solves a binary classification problem. They give you 3000 training examples and a set of 100 features for each example. You have been assured that the examples have been generated in way that makes them IID, and the examples have been given to you *sorted* based on the values of the first feature. The client not only wants an accurate model, but they want an estimate of the accuracy of the final model.

Assume that the data is stored in 3000 by 100 matrix X, and the labels are stored in a 3000 by 1 vector y. As in the assignments, assume that you have a 'model' function that depends on a parameter 'k' with the following interface:

- model = train(X,y,k); % Train model on $\{X, y\}$ with parameter k
- yhat = predict(model,Xhat); % Predicts using the model on $Xhat$.

Assume that k can take any integer value from 1 to 10.

Give pseudo-code for a training/validation/testing procedure that:

(a) Chooses a good value of 'k'.

(b) Reports an unbiased estimate of the accuracy of the final model.

2 KNN, Naive Bayes, and Softmax

Consider the dataset below, which has 10 training examples and 2 features:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}.$$

Suppose that you want to classify the following test example:

$$\hat{x} = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

(a) What class label would we assign to the test example if we used a k-nearest neighours classifier, with $k = 3$ and the Euclidean distance measure?

(b) What class label would we assign to the test example under a naive Bayes model? (Show your work.)

(c) Suppose we fit a multi-class linear classifier using the softmax loss, and we obtain the following weight matrix:

$$W = \begin{bmatrix} +2 & +2 & +3 \\ -1 & +2 & -1 \end{bmatrix}$$

Under this model, what class label would we assign to the test example? (Show your work.)

3 Parametric vs. Non-Parametric

Make a table with a column labeled ‘parametric’ and a column labeled ‘non-parametric’. Place each of the following methods into one of the columns:

- Mean of a list of numbers.
- Scatterplot.
- Depth-10 decision trees.
- 3-nearest neighbours.
- Naive Bayes with 1000 variables.
- Random forests with 10 random trees of depth 10.
- K-means with 5 means.
- Density-based clustering.
- Linear regression with linear basis.
- Linear regression with RBF basis.
- Principal component analysis.
- Non-negative matrix factorization.
- Sammon mapping.
- Neural networks.

4 L1-Regularized Latent-Factor Model

We have a matrix X, where we have observed a subset of its individual elements. Let $\mathcal{R}$ be the set of indices (i,j) where we have observed the element x_{ij}. We want to build a model that predicts the missing entries, so we use a latent-factor model with an L1-regularizer on the coefficients W and a separate L2-regularizer on the coefficients Z,

$$\underset{W\in\mathbb{R}^{k\times d}, Z\in\mathbb{R}^{n\times k}}{\operatorname{argmin}} \sum_{(i,j)\in\mathcal{R}} \left[\frac{1}{2}(x_{ij} - w_j^T z_i)^2\right] + \lambda_W \sum_{j=1}^{d} \left[\|w_j\|_1\right] + \lambda_Z \sum_{i=1}^{n} \left[\|z_i\|^2\right],$$

where the regularization parameters satisfy $\lambda_W > 0$ and $\lambda_Z > 0$.

(a) What is the affect of λ_W on the sparsity of the parameters W and Z? What is the effect of λ_Z on the sparsity of W and Z?

(b) What is the affect of λ_Z on the two parts of the fundamental trade-off in machine learning? What is the effect of k on the two parts?

(c) Would the answers to (b) change if $\lambda_W = 0$?

(d) Suppose each element of the matrix X is either $+1$ or -1 and our goal is to build a model that makes the sign of $w_j^T z_i$ match the sign of x_{ij}. Write down a (continuous) objective function that would be more suitable.

5 Label Propagation

Consider a transductive learning setting where we have a set of labelled examples $y_i \in \{-1, +1\}$ for i ranging from 1 to n. We also have a set of t unlabeled examples that we would like to label. While we do not have features for any examples, we are given weights w_{ij} indicating how strongly we prefer unlabeled example i to have the same label as labeled example j, and another set of weights v_{ij} indicating how strongly we prefer unlabeled example i to have the same label as unlabeled example j. We'll assume that $v_{ij} = v_{ji}$ and $v_{ii} = 0$.

To find the labels of the unlabeled examples, a standard label propagation objective is

$$\underset{\hat{y}_1 \in \mathbb{R}, \hat{y}_2 \in \mathbb{R}, \dots, \hat{y}_t \in R}{\operatorname{argmin}} \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{t} \left[w_{ij}(y_j - \hat{y}_i)^2 \right] + \frac{1}{2} \sum_{i=1}^{t} \sum_{j=i+1}^{t} \left[v_{ij}(\hat{y}_j - \hat{y}_i)^2 \right] + \frac{\lambda}{2} \sum_{i=1}^{t} \hat{y}_i^2.$$

The regularization term encourages the predictions to be close to 0, so that the model becomes less confident in the labels of examples where we have to propagate labels quite far to reach them. Although we can fit this model with gradient descent, a standard approach to fitting it is by cycling through each of the $\hat{y}_i$ and updating them to their optimal value given the values of the remaining $\hat{y}_j$ for $j \neq i$.

(a) Derive the partial derivative of this objective function with respect to a particular $\hat{y}_i$.

(b) Derive the optimal value of a particular $\hat{y}_i$, given the values of the remaining $\hat{y}_j$ for $j \neq i$.

(c) Describe a procedure for selecting a good value of λ.

6 Outlierness Ratio

In class we defined an 'outlierness' ratio of an example $x_i \in \mathbb{R}^d$ for $i = 1$ to n. This ratio depends on the k-nearest neighbours, $N_k(x_i)$, and the average distance to these k-nearest neighbours

$$D_k(x_i) = \frac{1}{k} \sum_{j \in N_k(x_i)} \|x_i - x_j\|.$$

Given these definitions, the 'outlierness' ratio is defined by the quantity

$$O(x_i) = \frac{D_k(x_i)}{\frac{1}{k}\sum_{j \in N_k(x_i)} D_k(x_j)},$$

which roughly measures whether x_i is further away from its neighbours than its neighbours are from their neighbours (we'll assume that no points have the exact same distance from each other).

(a) If we want to compute this measure for a single example x_i, what is the cost of computing this measure in $O()$ notation in terms of n, d, and k (give your reasoning)?

(b) Consider the case where you don't have explicit x_i values, but you instead have an *undirected graph* defined on the examples, and each edge in the graph has a *similarity score* in the range $(0, 1]$ between examples. Describe how you could define something like the outlierness ratio in this setting. (You can assume that the graph is connected, but you should not assume that each point has at least k neighbours in the graph.)

7 Principal Component Analysis

Consider the following dataset, containing 5 examples with 2 features each:

x_1	x_2
-2	-1
-1	0
0	1
1	2
2	3

(a) What is the first principal component?

(b) What is the (L2-norm) reconstruction error of the point (3,3)? (Show your work.)

(c) What is the (L2-norm) reconstruction error of the point (3,4)? (Show your work.)

8 Poisson Regression

Suppose we have a set of training examples (x_i, y_i) and we want to fit a linear model of the form $y_i \approx w^T x_i$. However, we do not want to use the squared error since the values in y_i represents *counts* (like 'number of Facebook likes'). So instead we assume that y_i follows a Poisson distribution with a mean of $\exp(w^T x_i)$,

$$p(y_i|w^T x_i) = \frac{\exp(y_i w^T x_i)\exp(-\exp(w^T x_i))}{y_i!}.$$

We want to find the w that maximizes these probabilities assuming that our examples are IID,

$$\underset{w \in \mathbb{R}^d}{\text{argmax}} \prod_{i=1}^{n} p(y_i|w^T x_i).$$

(a) Show how finding w corresponds to minimizing an additive loss function,

$$\underset{w \in \mathbb{R}^d}{\textbf{argmin}} \sum_{i=1}^{n} f(y_i, w^T x_i),$$

and derive the form of this loss function (simplifying as much as possible).

(b) If the largest value of y_i in the training set is k, what is the cost of evaluating this objective function in terms of n, d, and k?

(c) Given the parameters w and the features $\hat{x}$ for a new example, derive an efficient algorithm for finding a value of c that maximizes $p(\hat{y} = c|w^T\hat{x})$.

(Hint: try to discover the relationship between $p(\hat{y} = c|w^T\hat{x})$ and $p(\hat{y} = c - 1|w^T\hat{x})$.)

9 Stochastic Gradient

Using the L2-regularized logistic loss to fit a binary classifier corresponds to solving the optimization problem

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^{n} [\log(1 + \exp(-y_i w^T x_i))] + \frac{\lambda}{2} w^T w.$$

Using $f(w)$ to denote the objective function, the gradient of the objective function can be written in the form

$$\nabla f(w) = \sum_{i=1}^{n} [g(x_i, w) x_i + \frac{\lambda}{n} w].$$

for a function g that returns a scalar given the training example x_i and parameter vector w. The cost of computing g is $O(m)$ if x_i has m non-zero values, since it requires multiplying each non-zero element of x_i by the corresponding element of w, so in the worst case computing g costs $O(d)$.

(a) Write pseudo-code doing an iteration of stochastic gradient on this model with a constant step-size α. What is the cost of performing an iteration of stochastic gradient in terms of n and d? (You can assume that generating a random number between 1 and n costs $O(1)$.)

(b) How does the cost per iteration in part (a) change if each x_i has at most m non-zeroes?

(c) Show how we can reduce the cost in part (b) by representing w as the product of a scalar β and a vector v, so that $w = \beta v$.