CSE 158 – Lecture 10
Web Mining and Recommender Systems

Midterm recap
Midterm on Wednesday!

- 5:10 pm – 6:10 pm
- Closed book – but I’ll provide a similar level of basic info as in the last page of previous midterms

- Assignment 2 will also be out this week (but we can talk about that next week)
Week 1 recap
Learning approaches attempt to model data in order to solve a problem.

Unsupervised learning approaches find patterns/relationships/structure in data, but are not optimized to solve a particular predictive task.
• E.g. PCA, community detection

Supervised learning aims to directly model the relationship between input and output variables, so that the output variables can be predicted accurately given the input.
• E.g. linear regression, logistic regression
Linear regression assumes a predictor of the form

\[ X\theta = y \]

(or \( Ax = b \) if you prefer)
Regression diagnostics

Mean-squared error (MSE)

\[
\frac{1}{N} \| y - X \theta \|_2^2
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} (y_i - X_i \cdot \theta)^2
\]
Representing the month as a feature

How would you build a feature to represent the **month**?
Representing the month as a feature
Occam’s razor

“Among competing hypotheses, the one with the fewest assumptions should be selected”
Regularization is the process of penalizing model complexity during training.

\[ \arg\min_{\theta} = \frac{1}{N} \| y - X\theta \|_2^2 + \lambda \| \theta \|_2^2 \]

How much should we trade-off accuracy versus complexity?
A validation set is constructed to "tune" the model's parameters

- Training set: used to optimize the model's parameters
- Test set: used to report how well we expect the model to perform on unseen data
- Validation set: used to tune any model parameters that are not directly optimized
Regularization
A few “theorems” about training, validation, and test sets

• The training error increases as lambda increases
• The validation and test error are at least as large as the training error (assuming infinitely large random partitions)
• The validation/test error will usually have a “sweet spot” between under- and over-fitting
CSE 158 – Lecture 10
Web Mining and Recommender Systems

Week 2
Classification

Will I **purchase** this product?  
(yes)

Will I **click on** this ad?  
(no)
Classification

What animal appears in this image?

(mandarin duck)
What are the **categories** of the item being described?

(book, fiction, philosophical fiction)

From [Booklist](#)

Houellebecq's deeply philosophical novel is about an alienated young man searching for happiness in the computer age. Bored with the world and too weary to try to adapt to the foibles of friends and coworkers, he retreats into himself, descending into depression while attempting to analyze the passions of the people around him. Houellebecq uses his nameless narrator as a vehicle for extended exploration into the meanings and manifestations of love and desire in human interactions. Ironically, as the narrator attempts to define love in increasingly abstract terms, he becomes less and less capable of experiencing that which he is so desperate to understand. Intelligent and well written, the short novel is a thought-provoking inspection of a generation's confusion about all things sexual. Houellebecq captures precisely the cynical disillusionment of disaffected youth. *Bonnie Johnston* --This text refers to an out of print or unavailable edition of this title.
Linear regression assumes a predictor of the form

$$X\theta = y$$

- matrix of features (data)
- unknowns (which features are relevant)
- vector of outputs (labels)
Regression vs. classification

But how can we predict **binary** or **categorical** variables?

\[ f(\text{data}) \rightarrow \text{labels} \]

\[ \{0, 1\}, \{\text{True, False}\} \]

\[ \{1, \ldots, N\} \]
We’ll attempt to build classifiers that make decisions according to rules of the form

\[ y_i = \begin{cases} 
1 & \text{if } X_i \cdot \theta > 0 \\
0 & \text{otherwise} 
\end{cases} \]
In week 2

1. Naïve Bayes
   Assumes an independence relationship between the features and the class label and “learns” a simple model by counting

2. Logistic regression
   Adapts the regression approaches we saw last week to binary problems

3. Support Vector Machines
   Learns to classify items by finding a hyperplane that separates them
Naïve Bayes (2 slide summary)

\[
(f_{i} \perp f_{j} | \text{label})
\]

\[
p(f_{i}, f_{j} | \text{label}) = p(f_{i} | \text{label}) p(f_{j} | \text{label})
\]
Naïve Bayes (2 slide summary)
Double-counting: naïve Bayes vs Logistic Regression

Q: What would happen if we trained two regressors, and attempted to “naively” combine their parameters?

\[
\text{no. of pages} = \alpha + \beta_1 \cdot \delta(\text{mentions wizards})
\]

\[
\text{no. of pages} = \alpha + \beta_2 \cdot \delta(\text{mentions witches})
\]

\[
\text{no. of pages} = \alpha + \beta_1 \cdot \delta(\text{mentions wizards}) + \beta_2 \cdot \delta(\text{mentions witches})
\]
Logistic regression

**sigmoid function:** \( \sigma(t) = \frac{1}{1+e^{-t}} \)
Logistic regression

**Training:**

$X_i \cdot \theta$ should be maximized when $y_i$ is positive and minimized when $y_i$ is negative

$$\arg \max_\theta \prod_i \delta(y_i = 1)p_\theta(y_i|X_i) + \delta(y_i = 0)(1 - p_\theta(y_i|X_i))$$

$\delta(\text{arg}) = 1$ if the argument is true, $= 0$ otherwise
Logistic regression

$$\text{arg max}_{\theta} \prod_i \delta(y_i = 1)p_{\theta}(y_i | X_i) + \delta(y_i = 0)(1 - p_{\theta}(y_i | X_i))$$
Q: Where would a logistic regressor place the decision boundary for these features?
Logistic regression

- Logistic regressors don’t optimize the number of “mistakes”
- No special attention is paid to the “difficult” instances – every instance influences the model
- But “easy” instances can affect the model (and in a bad way!)
- How can we develop a classifier that optimizes the number of mislabeled examples?
Support Vector Machines

\[
\begin{align*}
\theta x - \alpha &= 1 \\
\theta x - \alpha &= 0 \\
\theta x - \alpha &= -1
\end{align*}
\]

\[
\arg \min_{\theta, \alpha} \frac{1}{2} \|\theta\|_2^2 \\
\text{such that} \\
\forall i y_i (\theta \cdot X_i - \alpha) \geq 1
\]
The classifiers we’ve seen in Week 2 all attempt to make decisions by associating weights (theta) with features (x) and classifying according to

\[ y_i = \begin{cases} 
1 & \text{if } X_i \cdot \theta > 0 \\
0 & \text{otherwise}
\end{cases} \]
Summary

• **Naïve Bayes**
  • Probabilistic model (fits $p(label|data)$)
  • Makes a conditional independence assumption of the form $(feature_i \perp \!\!\!\!\!\!\perp feature_j | label)$ allowing us to define the model by computing $p(feature_i | label)$ for each feature
  • Simple to compute just by counting

• **Logistic Regression**
  • Fixes the “double counting” problem present in naïve Bayes

• **SVMs**
  • Non-probabilistic: optimizes the classification error rather than the likelihood
Which classifier is best?

1. When data are highly imbalanced
   If there are far fewer positive examples than negative examples we may want to assign additional weight to negative instances (or vice versa)

   e.g. will I purchase a product? If I purchase 0.00001% of products, then a classifier which just predicts “no” everywhere is 99.99999% accurate, but not very useful
2. When mistakes are more costly in one direction

False positives are nuisances but false negatives are disastrous (or vice versa)

e.g. which of these bags contains a weapon?
Which classifier is best?

3. When we only care about the “most confident” predictions

E.g. does a relevant result appear among the first page of results?
Evaluating classifiers

decision boundary

negative  positive
Evaluating classifiers

![Confusion Matrix Diagram]

- **Label**
  - true
  - false

- **Prediction**
  - true
  - false

<table>
<thead>
<tr>
<th></th>
<th>true</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true positive</td>
<td>false positive</td>
</tr>
<tr>
<td>false</td>
<td>false negative</td>
<td>true negative</td>
</tr>
</tbody>
</table>

**Classification accuracy**

\[
\text{Classification accuracy} = \frac{\text{correct predictions}}{\#\text{predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}
\]

**Error rate**

\[
\text{Error rate} = \frac{\text{incorrect predictions}}{\#\text{predictions}} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}
\]
• Linear classification – know what the different classifiers are and when you should use each of them. What are the advantages/disadvantages of each
• Know how to evaluate classifiers – what should you do when you care more about false positives than false negatives etc.
CSE 158 – Lecture 10
Web Mining and Recommender Systems

Week 3
Why dimensionality reduction?

Goal: take **high-dimensional** data, and describe it compactly using a small number of dimensions.

Assumption: Data lies (approximately) on some **low-dimensional** manifold. (a few dimensions of opinions, a small number of topics, or a small number of communities)
Principal Component Analysis

\[ X \]

- rotate

[\varphi]

- discard lowest-variance dimensions

\[ Y \]

- un-rotate

\[ \varphi^T \]
Principal Component Analysis

Construct such vectors from 100,000 patches from real images and run PCA:

Color:
We want to find a low-dimensional representation that best compresses or "summarizes" our data.

To do this we’d like to keep the dimensions with the highest variance (we proved this), and discard dimensions with lower variance. Essentially we’d like to capture the aspects of the data that are “hardest” to predict, while discard the parts that are “easy” to predict.

This can be done by taking the eigenvectors of the covariance matrix (we didn’t prove this, but it’s right there in the slides).
Q: What would PCA do with this data?
A: Not much, variance is about equal in all dimensions.
Clustering

**But:** The data are highly **clustered**

Idea: can we compactly describe the data in terms of **cluster memberships**?
K-means Clustering

1. Input is still a matrix of features:

   \[
   X = \begin{pmatrix}
   5 & 3 & \cdots & 1 \\
   4 & 2 & 1 \\
   3 & 1 & 3 \\
   2 & 2 & 4 \\
   1 & 5 & 2 \\
   \vdots & \vdots & \vdots \\
   1 & 2 & \cdots & 1
   \end{pmatrix}
   \]

2. Output is a list of cluster “centroids”:

   \[
   \text{centroids} = \begin{pmatrix}
   1.1 & 2.1 \\
   3.5 & 1.8 \\
   0.2 & 0.1 \\
   3.0 & -0.3
   \end{pmatrix}
   \]

3. From this we can describe each point in \( X \) by its cluster membership:

   \[
   Y = (1, 2, 4, 3, 4, 2, 4, 2, 2, 3, 3, 2, 1, 1, 3, \ldots, 2)
   \]

   \[
   f = [0, 0, 1, 0] \\
   f = [0, 0, 0, 1]
   \]
K-means Clustering

Greedy algorithm:

1. Initialize C (e.g. at random)
2. Do
3. Assign each $X_i$ to its nearest centroid
4. Update each centroid to be the mean of points assigned to it
5. While (assignments change between iterations)

$$y_i = \arg \min_k \| X_i - C_k \|_2^2$$

$$C_k = \frac{\sum_i \delta(y_i = k) X_i}{\sum_i \delta(y_i = k)}$$

(also: reinitialize clusters at random should they become empty)
Q: What if our clusters are hierarchical?

A: We’d like a representation that encodes that points have some features in common but not others.
Hierarchical (agglomerative) clustering works by gradually fusing clusters whose points are closest together.

Assign every point to its own cluster:
Clusters = [[[1],[2],[3],[4],[5],[6],...,[N]]
While len(Clusters) > 1:
   Compute the center of each cluster
   Combine the two clusters with the nearest centers
1. Connected components

Define communities in terms of sets of nodes which are reachable from each other

- If \( a \) and \( b \) belong to a **strongly connected component** then there must be a path from \( a \rightarrow b \) and a path from \( b \rightarrow a \)
- A **weakly connected component** is a set of nodes that would be strongly connected, if the graph were undirected
What is the **Ratio Cut** cost of the following two cuts?

\[
\text{Ratio Cut}(\cdots) = \frac{1}{2} \left( \frac{3}{33} + \frac{3}{1} \right) = 1.54545
\]

\[
\text{Ratio Cut}(\cdots) = \frac{1}{2} \left( \frac{9}{16} + \frac{9}{18} \right) = 0.53125
\]
3. Clique percolation

- Clique percolation searches for “cliques” in the network of a certain size (K). Initially each of these cliques is considered to be its own community.
- If two communities share a (K-1) clique in common, they are merged into a single community.
- This process repeats until no more communities can be merged.

1. Given a clique size K
2. Initialize every K-clique as its own community
3. While (two communities I and J have a (K-1)-clique in common):
4. Merge I and J into a single community
Week 3

• Clustering & Community detection – understand the basics of the different algorithms
  • Given some features, know when to apply PCA vs. K-means vs. hierarchical clustering
  • Given some networks, know when to apply clique percolation vs. graph cuts vs. connected components
Definitions

Or equivalently...

\[
R = \begin{pmatrix}
1 & 0 & \cdots & 1 \\
0 & 0 & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 0 & \cdots & 1
\end{pmatrix}
\]

\(R_u\) = binary representation of items purchased by \(u\)

\(R_{.,i}\) = binary representation of users who purchased \(i\)

\(I_u = \)

\(U_i = \)
Recommender Systems Concepts

• How to represent rating / purchase data as sets/matrices
• Similarity measures (Jaccard, cosine, Pearson correlation)
• Very basic ideas behind latent factor models
Jaccard similarity

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

$$Jaccard(U_i, U_j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|}$$

- Maximum of 1 if the two users purchased exactly the same set of items (or if two items were purchased by the same set of users)
- Minimum of 0 if the two users purchased completely disjoint sets of items (or if the two items were purchased by completely disjoint sets of users)
Cosine similarity

\[ \text{Cosine}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \]

\[ \theta = \cos^{-1} \left( \frac{A \cdot B}{\|A\| \|B\|} \right) \]

- \( \cos(\theta) = 1 \)  
  (\( \theta = 0 \)) \( \rightarrow \) A and B point in exactly the same direction

- \( \cos(\theta) = -1 \)  
  (\( \theta = 180 \)) \( \rightarrow \) A and B point in opposite directions (won’t actually happen for 0/1 vectors)

- \( \cos(\theta) = 0 \)  
  (\( \theta = 90 \)) \( \rightarrow \) A and B are orthogonal

\( U_{\text{harry potter}} \)  
(vector representation of users who purchased harry potter)
Pearson correlation

Compare to the cosine similarity:

Pearson similarity (between users):

\[
\text{Sim}(u, v) = \frac{\sum_{i \in I_u \cap I_v} (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (R_{u,i} - \bar{R}_u)^2 \sum_{i \in I_u \cap I_v} (R_{v,i} - \bar{R}_v)^2}}
\]

Cosine similarity (between users):

\[
\text{Sim}(u, v) = \frac{\sum_{i \in I_u \cap I_v} R_{u,i} R_{v,i}}{\sqrt{\sum_{i \in I_u \cap I_v} R_{u,i}^2 \sum_{i \in I_u \cap I_v} R_{v,i}^2}}
\]
Rating prediction

\[ f(u, i) = \alpha + \beta_u + \beta_i \]

- How much does this user tend to rate things above the mean?
- Does this item tend to receive higher ratings than others?

\[
\begin{align*}
\alpha &= 4.2 \\
\beta_{\text{pitch black}} &= -0.1 \\
\beta_{\text{julian}} &= -0.2
\end{align*}
\]
Latent-factor models

\[ f(u, i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i \]
CSE 158 – Lecture 10
Web Mining and Recommender Systems

Last year’s midterm
Section 1: Regression

Feature design

Suppose we collected the following data about businesses from Google Local:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Av. Rating</th>
<th>Price</th>
<th>Address</th>
<th>Latitude/Longitude</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>T C’s Referee Sports Bar</td>
<td>5.0</td>
<td>$$$</td>
<td>Sioux Falls, SD 57106</td>
<td>43.529, -96.792</td>
<td>m-f/11am-10pm, s-s/11am-1am</td>
</tr>
<tr>
<td>1</td>
<td>Old Chicago</td>
<td>3.0</td>
<td>$$$</td>
<td>Beaverton, OR 97006</td>
<td>45.535, -122.862</td>
<td>m-f/11am-1am</td>
</tr>
<tr>
<td>2</td>
<td>Sabatino’s Italian Kitchen</td>
<td>4.0</td>
<td>$$$</td>
<td>Arlington, MA 02474</td>
<td>42.406, -71.143</td>
<td>m-f/10am-10pm, s-s/10am-9pm</td>
</tr>
<tr>
<td>3</td>
<td>Oakville Grocery</td>
<td>4.5</td>
<td>$</td>
<td>Healdsburg, CA 95448</td>
<td>35.063, 121.524</td>
<td>mon-sun/9am-5pm</td>
</tr>
<tr>
<td>4</td>
<td>Hog Wild Pit BBQ</td>
<td>3.5</td>
<td>$$$</td>
<td>Wichita, KS 67213</td>
<td>37.681, -97.389</td>
<td>mon-sun/11am-8pm</td>
</tr>
</tbody>
</table>

1. What would be the mean squared error (MSE) of a trivial model that simply predicted the average rating for all items (1 mark)? \[\text{A:} \]

2. Suppose we wanted to train a personalized model that predicted the rating I would give to a business based on the population-level average and price, i.e.,

   \[
   \text{my rating} = \theta_0 + \theta_1[\text{average rating}] + \theta_2[\text{price}].
   \]

   Write down the complete feature matrix (in the space below) that you would use to solve the above equation (1 mark):

   \[
   \begin{bmatrix}
   \theta_0 \\
   \theta_1 \\
   \theta_2
   \end{bmatrix}
   =
   \begin{bmatrix}
   y_1 \\
   y_2 \\
   y_3 \\
   \vdots \\
   y_n
   \end{bmatrix}
   \]
3. Write down the predictions that would be obtained for the five businesses if using the features above if the parameters were $\theta = [0.1, 1.0, -0.2]^T$ (1 mark)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Predicted Rating</th>
<th>Q4 answer</th>
<th>Q5 answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>T C's Referee Sports Bar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Old Chicago</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sabatinos Italian Kitchen</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Oakville Grocery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Hog Wild Pit BBQ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. The opening hours might also influence my preferences. How would you construct useful features for the above businesses, if I have a preference toward businesses that are open late? Using your representation write down (in the table above) the features corresponding to each business (1 mark):

A:
4. The opening hours might also influence my preferences. How would you construct useful features for the above businesses, if I have a preference toward businesses that are open late? Using your representation write down (in the table above) the features corresponding to each business (1 mark):

A:

5. How would you incorporate opening-hour features if my preferences are toward businesses that are open outside of work hours (i.e., mon-fri/9am-5pm)? Write down the features corresponding to each business (1 mark):

A:

6. Finally, suppose I want to model how people's preferences change as a function of geography (based on 1,000 U.S. businesses including the five above), i.e.,

\[
\text{average rating} = x(\text{geographical features}) \cdot \theta
\]
How might you use the features available above (e.g. address or latitude/longitude) to model such geographical trends (1 mark)? (describe your solution, rather than writing down the actual features)

A:

Diagnostics

7. Suppose we trained our model above by minimizing the regularized mean squared error, i.e.,

$$\arg\min_{\theta} \|y - X\theta\|^2_2 + \lambda \|\theta\|^2_2$$

Suppose that we split our data into training, validation, and test sets (and that we do so randomly, given plenty of data). Which of the plots below could correspond to the performance (i.e., MSE) on the training and validation sets? For each that could not, briefly explain why below (2 marks).
Section 2: Classification

The following is a list of Vin Diesel’s films:

<table>
<thead>
<tr>
<th>No.</th>
<th>Title</th>
<th>Year</th>
<th>IMDB rating</th>
<th>MPAA rating</th>
<th>length (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Last Witch Hunter</td>
<td>2015</td>
<td>6.3</td>
<td>PG-13</td>
<td>106</td>
</tr>
<tr>
<td>2</td>
<td>Furious 7</td>
<td>2015</td>
<td>7.4</td>
<td>PG-13</td>
<td>137</td>
</tr>
<tr>
<td>3</td>
<td>Guardians of the Galaxy</td>
<td>2014</td>
<td>8.1</td>
<td>PG-13</td>
<td>121</td>
</tr>
<tr>
<td>4</td>
<td>Riddick</td>
<td>2013</td>
<td>6.4</td>
<td>R</td>
<td>119</td>
</tr>
<tr>
<td>5</td>
<td>Fast &amp; Furious 6</td>
<td>2013</td>
<td>7.2</td>
<td>PG-13</td>
<td>130</td>
</tr>
<tr>
<td>6</td>
<td>Fast Five</td>
<td>2011</td>
<td>7.3</td>
<td>PG-13</td>
<td>131</td>
</tr>
<tr>
<td>7</td>
<td>Fast &amp; Furious</td>
<td>2009</td>
<td>6.6</td>
<td>PG-13</td>
<td>107</td>
</tr>
<tr>
<td>8</td>
<td>The Fast and the Furious: Tokyo Drift</td>
<td>2006</td>
<td>6.0</td>
<td>PG-13</td>
<td>104</td>
</tr>
<tr>
<td>9</td>
<td>The Pacifier</td>
<td>2005</td>
<td>5.5</td>
<td>PG</td>
<td>95</td>
</tr>
<tr>
<td>10</td>
<td>The Chronicles of Riddick</td>
<td>2004</td>
<td>6.7</td>
<td>PG-13</td>
<td>119</td>
</tr>
<tr>
<td>11</td>
<td>xXx</td>
<td>2002</td>
<td>5.8</td>
<td>PG-13</td>
<td>124</td>
</tr>
<tr>
<td>12</td>
<td>The Fast and the Furious</td>
<td>2001</td>
<td>6.7</td>
<td>PG-13</td>
<td>106</td>
</tr>
<tr>
<td>13</td>
<td>Pitch Black</td>
<td>2000</td>
<td>7.1</td>
<td>R</td>
<td>109</td>
</tr>
<tr>
<td>14</td>
<td>The Iron Giant</td>
<td>1999</td>
<td>8.0</td>
<td>PG</td>
<td>86</td>
</tr>
<tr>
<td>15</td>
<td>Saving Private Ryan</td>
<td>1998</td>
<td>8.6</td>
<td>R</td>
<td>169</td>
</tr>
</tbody>
</table>

You hear a rumor that Vin Diesel has a new film coming out that is (A) Over two hours long (B) Rated PG-13 (C) Has the word “Furious” in the title. Let’s try to estimate the probability that it will (D) have an IMDB rating of 7.0 or above.

8. Based on the data above (and not making any other assumptions) write down the probability

\[ p(D|A \land B \land C) \]

(1 mark) \[ A: \]
9. The above probability may be unreliable as it is based on very few observations that exhibit the required features. So, we'll try to decide whether D is likely to be true or not following the Naïve Bayes assumption. Write down all of the terms involved and finally the probability ratio, and the conclusion you draw as a result (3 marks).

A:
Evaluation measures

Suppose we are performing a ranking task to try and identify pages that are relevant to some particular search query, and that we achieve this by building a logistic regressor that outputs a score indicating the probability that a page is relevant. Suppose the scores we obtain are the following:

<table>
<thead>
<tr>
<th>page id</th>
<th>score</th>
<th>actually relevant?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.78</td>
<td>yes</td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>0.36</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>0.18</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>0.95</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>0.92</td>
<td>yes</td>
</tr>
<tr>
<td>7</td>
<td>0.11</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>0.20</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>0.56</td>
<td>no</td>
</tr>
</tbody>
</table>

10. Write down the number of true positives, true negatives, false positives, and false negatives of our logistic classifier (2 marks).

<table>
<thead>
<tr>
<th>TP:</th>
<th>TN:</th>
<th>FP:</th>
<th>FN:</th>
</tr>
</thead>
</table>

11. Complete the table below by ranking pages in decreasing order of confidence (3 marks).

<table>
<thead>
<tr>
<th>page id</th>
<th>confidence</th>
<th>actually relevant?</th>
<th>precision@k</th>
<th>recall@k</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.95</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Section 3: Communities & clustering

12. Suppose a social network is divided into the two communities shown below (filled vs. unfilled nodes). If we wanted an algorithm to find these communities automatically, which of ratio cuts versus normalized cuts would be more appropriate and why (1 mark)?

A:
14. Suppose you ran *hierarchical clustering* on the points below, resulting in the dendrogram shown in the center. How would you use the output of this algorithm (i.e., the clusters/dendrogram) to generate useful feature representations for the original points? Write your features for the 7 points below (1 mark).

A:
Last year’s midterm
Algorithm design

15. Suppose you wanted to design a system to estimate what tip a prospective fare would give for a taxi ride in San Diego. Describe below what data and features you would collect to estimate this value, and what techniques you would use to solve the task (3 marks).
CSE 158 – Lecture 10
Web Mining and Recommender Systems

Spring 2015 midterm
Section 1: Regression

Q1: Restaurants & ratings (10 marks)

Suppose we collected the following data about restaurants from *Yelp*:

<table>
<thead>
<tr>
<th>Name</th>
<th>Average Rating</th>
<th>Takes reservations?</th>
<th>Take-out?</th>
<th>Price</th>
<th>Good for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oceana Coastal Kitchen</td>
<td>4.5</td>
<td>Yes</td>
<td>No</td>
<td>$$$</td>
<td>Breakfast</td>
</tr>
<tr>
<td>Beyer Deli</td>
<td>5.0</td>
<td>No</td>
<td>Yes</td>
<td>$</td>
<td>Lunch</td>
</tr>
<tr>
<td>Werewolf</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Brunch</td>
</tr>
<tr>
<td>C Level</td>
<td>4.0</td>
<td>No</td>
<td>Yes</td>
<td>$$</td>
<td>Lunch, Dinner</td>
</tr>
<tr>
<td>Cucina Urban</td>
<td>4.5</td>
<td>Yes</td>
<td>Yes</td>
<td>$$</td>
<td>Dinner</td>
</tr>
</tbody>
</table>

and that from this data we want to estimate

\[
\text{av. rating} \approx \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}]
\]
and that from this data we want to estimate

\[
\text{av. rating} \simeq \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}]
\]

1. What is the average rating across all restaurants (1 mark)? \[ \text{A: } \]

2. What is the Mean Squared Error of the a predictor that just predicts the average rating for all items (1 mark)? \[ \text{A: } \]

3. Suppose we’d like to write down the above expression for the rating in the form \( y \simeq X\theta \). Complete the following equation to do so:

\[
\begin{bmatrix}
4.5 \\
\end{bmatrix}
\simeq
\begin{bmatrix}
1 & 1 & 0 & 3
\end{bmatrix}
\theta
\]

(1 mark)

4. In the expression \( y \simeq X\theta \), which term encodes the labels, which term encodes the features, and which term encodes the parameters (1 mark)? \[ \text{labels: features: parameters: } \]
and that from this data we want to estimate

$$\text{av. rating} \approx \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}]$$

5. Suppose that after fitting our model for the rating we obtain $\theta = [7, 0.5, -1, -1]^T$. What is the interpretation of $\theta_0 = 7$ in this expression (1 mark)?

A: 

6. What is the interpretation of $\theta_3 = -1$ (1 mark)?

A:
and that from this data we want to estimate

\[
\text{av. rating } \simeq \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}]
\]

9. Suppose you wanted to incorporate the ‘Good for’ field (the last column of the above table) into your model. How would you represent the features in order to do so? Answer this by writing down the model you would use:

\[
\text{av. rating } \simeq \theta_0 + \theta_1[\text{takes reservations}] + \theta_2[\text{has take-out}] + \theta_3[\text{price}]
\]

\[
\theta_3[\text{price}]+ \boxed{A:}
\]

and by completing the feature matrix using your representation:

\[
X = \begin{bmatrix}
1 & 1 & 0 & 3
\end{bmatrix}
\]

(2 marks)
Q2: Training, testing, & model selection (6 marks)

Suppose we are training regressors to minimize the regularized Mean Squared Error

\[
\sum_{(x,y) \in \text{train}} \frac{1}{|\text{train}|} (y - x \cdot \theta)^2 + \lambda \|\theta\|_2^2.
\]

10. Suppose that we fit some model for \(\lambda \in \{0.01, 0.1, 1, 10, 100, 1000\}\) and obtain the following performance on the training and validation sets:

Which value of \(\lambda\) would you select based on the results above (1 mark)? \[\lambda = \]

11. Answer the following questions about training, validation, and test sets:

(a) What is the role of a validation set (1 mark)?

A: 

(b) How does the training error typically vary with \(\lambda\) (1 mark)?

A: 

(c) What is meant by under/over fitting? Which values of \(\lambda\) in the above figure correspond to maximum over/under fitting (1 mark)?

A: 
12. Further suppose that we consider two different feature representations (model X and model Y), and two different regularization parameters ($\lambda = 1$ and $\lambda = 10$) and obtain the following results on the training and validation sets:

<table>
<thead>
<tr>
<th>model</th>
<th>training error</th>
<th>validation error</th>
</tr>
</thead>
<tbody>
<tr>
<td>model X, $\lambda = 1$</td>
<td>23.34</td>
<td>?</td>
</tr>
<tr>
<td>model X, $\lambda = 10$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>model Y, $\lambda = 1$</td>
<td>?</td>
<td>18.32</td>
</tr>
<tr>
<td>model Y, $\lambda = 10$</td>
<td>25.98</td>
<td>?</td>
</tr>
</tbody>
</table>

(‘?’ indicates an unknown value).

Assuming that our training/validation/test sets are large, independent samples, is the above information enough to determine which model and which value of $\lambda$ we would expect to yield the best performance on the test set? If so, which model and which value of $\lambda$ would you expect to perform best and why? Explain your answer (2 marks).

A:
Q3: Fantasy novels (6 marks)

Suppose we have a database consisting of the following books:

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Circle of Sorcerers</td>
<td>Fantasy</td>
<td>True</td>
</tr>
<tr>
<td>Knights: The Eye of Divinity</td>
<td>Fantasy</td>
<td></td>
</tr>
<tr>
<td>Superman/Batman: Sorcerer Kings</td>
<td>Graphic Novel</td>
<td></td>
</tr>
<tr>
<td>In the Blood</td>
<td>Mystery</td>
<td></td>
</tr>
<tr>
<td>Remains of the Day</td>
<td>Literature &amp; Fiction</td>
<td></td>
</tr>
<tr>
<td>Blood Song</td>
<td>Fantasy</td>
<td></td>
</tr>
<tr>
<td>Flame Moon</td>
<td>Fantasy</td>
<td></td>
</tr>
<tr>
<td>The Book of The Sword: A History of Daggers</td>
<td>Fantasy</td>
<td></td>
</tr>
<tr>
<td>A Storm of Swords</td>
<td>History</td>
<td></td>
</tr>
<tr>
<td>The Storm Book</td>
<td>Fantasy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children’s</td>
<td></td>
</tr>
</tbody>
</table>

Further, suppose we are given the following classifier to classify Fantasy vs. non-Fantasy books:

```plaintext
if (Title contains 'Sorcerer' or 'Blood' or 'Knights' or 'Moon' or 'Storm'):
    return True
else:
    return False
```

13. What are the predictions made by this classifier? Write your answers in the last column of the table above (1 mark).

14. Of these predictions, what is the number of true positives, true negatives, false positives, and false negatives (1 mark)?

<table>
<thead>
<tr>
<th></th>
<th>true positive</th>
<th>true negative</th>
<th>false positive</th>
<th>false negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>A:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15. What are the true positive rate (hint: TP / (TP + FN)), true negative rate, and balanced error rate (1 mark)?

<table>
<thead>
<tr>
<th></th>
<th>true positive rate</th>
<th>true negative rate</th>
<th>balanced error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>A:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

16. In class we saw three approaches to classification: naïve Bayes, logistic regression, and support-vector machines. Describe one benefit of each approach compared to the other two (3 marks).

 naïve Bayes:

 logistic regression:

 SVM:
Section 3: Communities & clustering

Q4: Algorithms for community detection, dimensionality reduction, and clustering

Recall three algorithms we saw in class to detect communities: connected components, ratio cut, and clique percolation (pseudocode is given as Algorithms 1, 2, and 3 at the end of the test).

17. Identify the communities that would be produced on the graphs below using these three algorithms. Circle the communities directly in the space below (some more graphs are provided on the final page in case you need to re-write your answer):

<table>
<thead>
<tr>
<th>Connected components</th>
<th>Ratio cut (2 communities)</th>
<th>Clique percolation ($k = 3$)</th>
<th>Clique percolation ($k = 2$)</th>
</tr>
</thead>
</table>

(1 mark) (1 mark) (1 mark) (1 mark)
18. Suppose we are given the following 2-dimensional data $X$, and wish to cluster it so as to minimize the reconstruction error ($\sum_{x \in X} \|\bar{x} - x\|^2$). Separate the points into three clusters such that the reconstruction error (when replacing each point by its cluster centroid) would be minimized. Draw the clusters directly in the space below (1 mark):

![Clusters Diagram]

19. By replacing each point with one of the three centroids above, we have effectively ‘compressed’ the data, since each (2-d) point is replaced by a (1-d) integer. Another way to compress the data would be to perform Principal Component Analysis, and discard the lowest variance dimension, which would also result in a 1-d representation of the data. Out of these two possible compressed representations, which one would result in the lower reconstruction error on the above data, and why (1 mark)?

A: 

20. In class we saw hierarchical clustering, an algorithm that works by successively joining clusters whose centroids are nearest. Psuedocode is given in Algorithm 4 over the page.

Suppose you are given the following set of points:

<table>
<thead>
<tr>
<th>Step</th>
<th>Clusters merged</th>
<th>List of clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(initialization)</td>
<td>{a}, {b}, {c}, {d}, {e}, {f}, {g}</td>
</tr>
<tr>
<td>1</td>
<td>{a} merges with {b}</td>
<td>{a, b}, {c}, {d}, {e}, {f}, {g}</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>{a, b, c, d, e, f, g}</td>
</tr>
</tbody>
</table>

If we were to perform hierarchical clustering on this data, in what order would the clusters be joined? Answer this question by completing the table above (2 marks).
CSE 158 – Lecture 10
Web Mining and Recommender Systems

HW Questions
No reduction after degree 1 (HW1/wk1)
Train vs. lambda (Classification, HW1/wk2)
Misc. questions
Representing the **day** as a feature

How would you build a feature to represent the time of **day**?
Representing the *day* as a feature

How would you build a feature to represent the time of *day*?
• Suppose we have a linear regression model to predict college GPA
• One of the features of this model encodes whether a student owns a car
• The fitted model looks like:

\[ y = \ldots - 0.4[\text{owns a car}] + \ldots \]

Conclusion: “The GPA of the average student who owns a car is 0.4 lower than that of the average student”

Q: is this conclusion reasonable?