CSE 190 – Lecture 3
Data Mining and Predictive Analytics

Supervised learning – Classification
• The examples are in Python but I only know Java/ArnoldC/Malbolge!
• What are training/test/MSE/These funny symbols → ∥ · ∥?
Last week we started looking at supervised learning problems.

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

We studied **linear regression**, in order to learn linear relationships between features and parameters to predict **real-valued** outputs.

\[ X \theta = y \]

- **matrix of features** (data)
- **unknowns** (which features are relevant)
- **vector of outputs** (labels)
Last week...

\[ f(\text{user features}, \text{movie features}) \rightarrow \text{star rating} \]
How can we predict **binary** or **categorical** variables?

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

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

\{1, \ldots, N\}
Today...

Will I purchase this product? (yes)

Will I click on this ad? (no)
Today...

What animal appears in this image?

(mandarin duck)
Today...

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.
Today...

We’ll attempt to build **classifiers** that make decisions according to rules of the form

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

**Ranking** results in order of how likely they are to be relevant
This week...

Evaluating classifiers

• False positives are nuisances but false negatives are disastrous (or vice versa)
  • Some classes are very rare
• When we only care about the “most confident” predictions

e.g. which of these bags contains a weapon?
We want to associate a probability with a label and its negation:

\[
p(label|data) \]
\[
p(\neg label|data) \]

(classify according to whichever probability is greater than 0.5)

**Q:** How far can we get just by counting?
Naïve Bayes

e.g. \( p(\text{movie is "action" | schwarzenneger in cast}) \)

Just count!

\#fims with Arnold = 45

\#action films with Arnold = 32

\( p(\text{movie is "action" | schwarzenneger in cast}) = \frac{32}{45} \)
What about:

\[ p(\text{movie is “action”} \mid \text{schwarzenege in cast and release year = 2015 and mpaa rating = PG and budget < $1000000}) \]

#(training) fims with Arnold, released in 2015, rated PG, with a budget below $1M = 0

#(training) action fims with Arnold, released in 2015, rated PG, with a budget below $1M = 0
Q: If we’ve never seen this combination of features before, what can we conclude about their probability?

A: We need some **simplifying assumption** in order to associate a probability with this feature combination.
Naïve Bayes assumes that features are conditionally independent given the label

\[(\text{feature}_i \perp \text{feature}_j | \text{label})\]
Conditional independence?

\[(a \perp b \mid c)\]

(a is conditionally independent of b, given c)

“if you know c, then knowing a provides no additional information about b”

(I remembered my umbrella \(\perp\) the streets are wet \(\mid\) it’s raining)
Naïve Bayes

\[(\text{feature}_i \perp \text{feature}_j | \text{label})\]

\[p(\text{feature}_i, \text{feature}_j | \text{label}) = p(\text{feature}_i | \text{label})p(\text{feature}_j | \text{label})\]
Naïve Bayes

\[ p(label|features) = \frac{p(label)p(features|label)}{p(features)} \]

due to our conditional independence assumption:

\[ p(label|features) = \frac{p(label) \prod_i p(feature_i|label)}{p(features)} \]
The denominator doesn’t matter, because we really just care about

\[ p(label|features) \] vs. \[ p(\neg label|features) \]

both of which have the same denominator
Amazon editorial descriptions:

Amazon.com Review

For most children, summer vacation is something to look forward to. But not for our 13-year-old uncle, and cousin who detest him. The third book in J.K. Rowling's Harry Potter series catapults Dursleys' dreadful visitor Aunt Marge to inflate like a monstrous balloon and drift up to the ceiling (and from officials at Hogwarts School of Witchcraft and Wizardry who strictly forbid students to out into the darkness with his heavy trunk and his owl Hedwig.

As it turns out, Harry isn't punished at all for his errant wizardry. Instead he is mysteriously rescued, violently purple bus to spend the remaining weeks of summer in a friendly inn. His third year at Hogwarts explains why the officials let him off easily. It seems that Sirius Black is loose. Not only that, but he's after Harry Potter. But why? And why do the Dementors, the guar are unaffected? Once again, Rowling has created a mystery that will have children and adults cl

50k descriptions:
http://jmcauley.ucsd.edu/cse190/data/amazon/book_descriptions_50000.json
Example 1

\[ P(\text{book is a children’s book} \mid \text{“wizard” is mentioned in the description} \text{ and } \text{“witch” is mentioned in the description}) \]

Code available on:

http://jmcauley.ucsd.edu/cse190/code/week2.py
Example 1

Conditional independence assumption:

“If you know a book is for children, then knowing that wizards are mentioned provides no additional information about whether witches are mentioned”

obviously ridiculous
Double-counting

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

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

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

\[
\text{no. of pages} = \alpha + \beta_1 \cdot \delta(\text{mentions wizards}) + \beta_2 \cdot \delta(\text{mentions witches})
\]
A: Since both features encode essentially the same information, we’ll end up double-counting their effect.
Logistic Regression also aims to model

\[ p(label|data) \]

By training a classifier of the form

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

Last week: regression

\[ y_i = X_i \cdot \theta \]

This week: **logistic** regression

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

Q: How to convert a real-valued expression \((X_i \cdot \theta \in \mathbb{R})\) Into a probability \((p_\theta(y_i|X_i) \in [0, 1])\)
Logistic regression

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

How to optimize?

\[ L_\theta(y|X) = \prod_{y_i=1} p_\theta(y_i|X_i) \prod_{y_i=0} (1 - p_\theta(y_i|X_i)) \]

- Take logarithm
- Subtract regularizer
- Compute gradient
- Solve using gradient \textbf{ascent}
  (solve on blackboard)
Logistic regression

Log-likelihood:

\[ l_\theta(y|X) = \sum_i - \log(1 + e^{-X_i \cdot \theta}) + \sum_{y_i=0} -X_i \cdot \theta - \lambda \|\theta\|_2^2 \]

Derivative:

\[ \frac{\partial l}{\partial \theta_k} = \sum_i X_{ik} (1 - \sigma(X_i \cdot \theta)) + \sum_{y_i=0} -X_{ik} - 2\lambda \theta_k \]
The most common way to generalize binary classification (output in \{0,1\}) to multiclass classification (output in \{1 \ldots N\}) is simply to train a binary predictor for each class.

e.g. based on the description of this book:
• Is it a Children’s book? \{yes, no\}
• Is it a Romance? \{yes, no\}
• Is it Science Fiction? \{yes, no\}
• ...

In the event that predictions are inconsistent, choose the one with the highest confidence.
Further reading:
- On Discriminative vs. Generative classifiers: A comparison of logistic regression and naïve Bayes (Ng & Jordan ‘01)
- Boyd-Fletcher-Goldfarb-Shanno algorithm (BFGS)
CSE 190 – Lecture 3
Data Mining and Predictive Analytics

Supervised learning – SVMs
**Q:** Where would a logistic regressor place the decision boundary for these features?
Q: Where would a logistic regressor place the decision boundary for these features?

- Positive examples: easy to classify
- Negative examples: hard to classify
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?
This is essentially the intuition behind Support Vector Machines (SVMs) – train a classifier that focuses on the “difficult” examples by minimizing the misclassification error.

We still want a classifier of the form

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

But we want to minimize the number of misclassifications:

\[ \arg \min_\theta \sum_i \delta(y_i(X_i \cdot \theta - \alpha) \leq 0) \]
Simple (seperable) case: there exists a perfect classifier
The classifier is defined by the hyperplane $\theta x - \alpha = 0$
Q: Is one of these classifiers preferable over the others?
Support Vector Machines

A: Choose the classifier that maximizes the distance to the nearest point
Support Vector Machines

\[ \theta x - \alpha = 1 \]
\[ \theta x - \alpha = 0 \]
\[ \theta x - \alpha = -1 \]

\[
\text{arg min}_{\theta, \alpha} \frac{1}{2} \| \theta \|^2_2
\]

such that
\[
\forall i \ y_i (\theta \cdot X_i - \alpha) \geq 1
\]
This is known as a “quadratic program” (QP) and can be solved using “standard” techniques.

\[
\begin{align*}
\arg\min_{\theta, \alpha} & \quad \frac{1}{2} \|\theta\|^2 \\
\text{such that} & \quad \forall_i y_i (\theta \cdot X_i - \alpha) \geq 1
\end{align*}
\]

See e.g. Nocedal & Wright (“Numerical Optimization”), 2006
Support Vector Machines

**But:** is finding such a separating hyperplane even possible?
Support Vector Machines

**Or:** is it actually a good idea?
Support Vector Machines

Want the margin to be as wide as possible

While penalizing points on the wrong side of it
Support Vector Machines

Soft-margin formulation:

$$\arg \min_{\theta, \alpha, \xi > 0} \frac{1}{2} \| \theta \|_2^2 + C \sum_i \xi_i$$

such that

$$\forall_i y_i (\theta \cdot X_i - \alpha) \geq 1 - \xi_i$$
Judging a book by its cover

Images features are available for each book on
http://jmcauley.ucsd.edu/cse190/data/amazon/book_images_5000.json

http://caffe.berkeleyvision.org/
Judging a book by its cover

Example: train an SVM to predict whether a book is a children’s book from its cover art

(code available on)
http://jmcauley.ucsd.edu/cse190/code/week2.py
The number of errors we made was extremely low, yet our classifier doesn’t seem to be very good – why?
(stay tuned next lecture!)
The classifiers we’ve seen today 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 $(feature_i \perp feature_j|label)$ 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
Questions?