CSE 255 – Lecture 1
Data Mining and Predictive Analytics

Supervised learning – Regression
What is supervised learning?

**Supervised learning** is the process of trying to infer from **labeled data** the underlying function that produced the labels associated with the data.
What is supervised learning?

Given **labeled training data** of the form

\[ \{(\text{data}_1, \text{label}_1), \ldots, (\text{data}_n, \text{label}_n)\} \]

Infer the function

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

Suppose we want to build a movie recommender

e.g. which of these films will I rate highest?
Q: What are the labels?

A: ratings that others have given to each movie, and that I have given to other movies
Example

Q: What is the **data**?

A: **features** about the movie and the users who evaluated it

Movie features: genre, actors, rating, length, etc.

User features: age, gender, location, etc.

<table>
<thead>
<tr>
<th>Product Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Genres</strong></td>
</tr>
<tr>
<td><strong>Director</strong></td>
</tr>
<tr>
<td><strong>Starring</strong></td>
</tr>
<tr>
<td><strong>Supporting actors</strong></td>
</tr>
<tr>
<td><strong>Studio</strong></td>
</tr>
<tr>
<td><strong>MPAA rating</strong></td>
</tr>
<tr>
<td><strong>Captions and subtitles</strong></td>
</tr>
<tr>
<td><strong>Rental rights</strong></td>
</tr>
<tr>
<td><strong>Purchase rights</strong></td>
</tr>
<tr>
<td><strong>Format</strong></td>
</tr>
</tbody>
</table>
Movie recommendation:

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

\[ = \]

\[ f(\text{user features, movie features}) \rightarrow \text{star rating} \]
Design a system based on **prior knowledge**, e.g.

```python
def prediction(user, movie):
    if (user['age'] <= 14):
        if (movie['mpaa_rating']) == "G"):
            return 5.0
        else:
            return 1.0
    else if (user['age'] <= 18):
        if (movie['mpaa_rating']) == "PG"):
            return 5.0
    ..... Etc.
```

Is this **supervised learning**?
Solution 2

Identify words that I frequently mention in my social media posts, and recommend movies whose plot synopses use similar types of language.

Plot synopsis

Is this supervised learning?

argmax similarity(synopsis, post)
Solution 3

Identify which attributes (e.g. actors, genres) are associated with positive ratings. Recommend movies that exhibit those attributes.

Is this **supervised learning?**
Solution 1

(design a system based on prior knowledge)

Disadvantages:
• Depends on possibly false **assumptions** about how users relate to items
• Cannot adapt to new data/information

Advantages:
• Requires no data!
Solution 2

(identify similarity between wall posts and synopses)

Disadvantages:
• Depends on possibly false assumptions about how users relate to items
• May not be adaptable to new settings

Advantages:
• Requires data, but does not require labeled data
Solution 3

(identify attributes that are associated with positive ratings)

Disadvantages:
• Requires a (possibly large) dataset of movies with labeled ratings

Advantages:
• Directly optimizes a measure we care about (predicting ratings)
• Easy to adapt to new settings and data
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.

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.
Regression is one of the simplest supervised learning approaches to learn relationships between input variables (features) and output variables (predictions)
Linear regression assumes a predictor of the form

\[ X\theta = y \]

(matrix of features (data)

unknowns (which features are relevant)

vector of outputs (labels)

(or \( Ax = b \) if you prefer)
Linear regression assumes a predictor of the form

\[ X \theta = y \]

**Q:** Solve for theta

**A:** \[ \theta = (X^T X)^{-1} X^T y \]
Example 1

How do preferences toward certain beers vary with age?
Example 1

**Beers:**

![Bourbon County Bottle](image)

**Ratings/reviews:**

*4.35/5* | *RoW-5.2%
---|---
Serving: 355 mL bottle poured into a 9 oz Libbey Embassy snifter (*bottled on: 08ABS14 1109*).
Appearance: Deep, dark near-black brown, Hazy, light brown fringes of foam and limited lacing; no head.
Smell: Roasted malt, vanilla, and some warming alcohol.
Taste: Roasted malts, cocoa, burnt caramel, molasses, vanilla and dark fruit. Bourbon barrel is hinted at but never takes over.
Mouthfeel: Medium to full body and light carbonation with a very lush, silky smooth feel.
Overall: Not as complex or intense as some newer barrel-aged stouts, but so smooth and balanced with all the elements tightly integrated.

*HipCzech, Yesterday at 05:38 AM*

**User profiles:**

*HipCzech*

Affiliado
Male, from Texas

Profile Page

Member Since: Jul 12, 2014
Points: 175
Beers: 108
Places: 6
Posts: 0
Smoother than all of it: 0
Likes Received: 0
Trading: 0% | 0

*Today at 02:19 AM*

*HipCzech was last seen: Today at 02:19 AM*
Example 1

50,000 reviews are available on
http://jmcauley.ucsd.edu/cse255/data/beer/beer_50000.json
(see course webpage)
How do preferences toward certain beers vary with age? How about \textbf{ABV}? 

Real-valued features

(code for all examples is on \url{http://jmcauley.ucsd.edu/cse255/code/lecture1.py})
Example 1

Real-valued features

What is the interpretation of:

\[ \theta = (3.4, 10e^{-7}) \]

(code for all examples is on http://jmcauley.ucsd.edu/cse255/code/lecture1.py)
Example 2

Categorical features

How do beer preferences vary as a function of gender?

(code for all examples is on http://jmcauley.ucsd.edu/cse255/code/lecture1.py)
Random features

What happens as we add more and more **random** features?

(code for all examples is on [http://jmcauley.ucsd.edu/cse255/code/lecture1.py](http://jmcauley.ucsd.edu/cse255/code/lecture1.py))
Regression diagnostics

**Mean-squared error (MSE)**

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

\[
= \frac{1}{N} \sum_{i=1}^{N} (y_i - X_i \cdot \theta)^2
\]
Q: Why MSE (and not mean-absolute-error or something else)
Regression diagnostics

Quantile-Quantile (QQ)-plot

Probability Plot

Ordered Values

Quantiles

$r^2 = 0.9714$
Coefficient of determination

Q: How low does the MSE have to be before it’s “low enough”?
A: It depends! The MSE is proportional to the variance of the data
Regression diagnostics

Coefficient of determination
(R^2 statistic)

Mean: $\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$

Variance: $Var(y) = \frac{1}{N} \sum_{i=1}^{N} (\bar{y} - y_i)^2$

MSE: $\frac{1}{N} \sum_{i=1}^{N} (X_i \cdot \theta - y_i)^2$
Regression diagnostics

Coefficient of determination
(R² statistic)

\[ FVU(f) = \frac{MSE(f)}{Var(y)} \]

(FVU = fraction of variance unexplained)

\[ FVU(f) = 1 \quad \rightarrow \quad \text{Trivial predictor} \]

\[ FVU(f) = 0 \quad \rightarrow \quad \text{Perfect predictor} \]
Coefficient of determination
(R^2 statistic)

\[ R^2 = 1 - \frac{MSE(f)}{Var(y)} \]

- R^2 = 0 → Trivial predictor
- R^2 = 1 → Perfect predictor
Q: But can’t we get an $R^2$ of 1 (MSE of 0) just by throwing in enough random features?

A: Yes! This is why MSE and $R^2$ should always be evaluated on data that wasn’t used to train the model.

A good model is one that generalizes to new data.
Overfitting

When a model performs well on \textit{training} data but doesn’t generalize, we are said to be \textit{overfitting}.

\textbf{Q:} What can be done to avoid overfitting?
Occam’s razor

“Among competing hypotheses, the one with the fewest assumptions should be selected”

(image from personalspirituality.net)
Q: What is a “complex” versus a “simple” hypothesis?
Q: What is a “complex” versus a “simple” hypothesis?
Occam’s razor

**A1:** A “simple” model is one where theta has few non-zero parameters (only a few features are relevant)

**A2:** A “simple” model is one where theta is almost uniform (few features are significantly more relevant than others)
Occam’s razor

**A1:** A “simple” model is one where theta has few non-zero parameters

\[ \|\theta\|_1 \text{ is small} \]

**A2:** A “simple” model is one where theta is almost uniform

\[ \|\theta\|_2 \text{ is small} \]

(“proof” on whiteboard)
Regularization is the process of penalizing model complexity during training.

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

- MSE
- (l2) model complexity
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?
Optimizing the (regularized) model

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

- We no longer have a convenient closed-form solution for theta
- Need to resort to some form of approximation algorithm
Optimizing the (regularized) model

Gradient descent:

1. Initialize $\theta$ at random
2. While (not converged) do
   \[ \theta := \theta - \alpha f'(\theta) \]

All sorts of annoying issues:
- How to initialize theta?
- How to determine when the process has converged?
- How to set the step size alpha

These aren’t really the point of this class though
Optimizing the (regularized) model

Gradient descent in scipy

(code for all examples is on http://jmcauley.ucsd.edu/cse255/code/lecture1.py)
Model selection

$$\arg \min_{\theta} = \frac{1}{N} \left\| y - X\theta \right\|^2_2 + \lambda \left\| \theta \right\|^2_2$$

How much should we trade-off accuracy versus complexity?

Each value of lambda generates a different model. \textbf{Q:} How do we select which one is the best?
Model selection

How to select which model is best?

**A1:** The one with the lowest training error?

**A2:** The one with the lowest test error?

We need a **third** sample of the data that is not used for training or testing.
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
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
Homework

Homework is **available** on the course webpage
http://cseweb.ucsd.edu/~jmcauley/cse255/homework1.pdf

Please submit it at the beginning of the next lecture (January 12)
Questions?