value means that the user has not read the joke, but doesn’t mean that the rating should be zero. A more reasonable choice is to minimize the MSE only on rated joke. Let’s define a loss function:

\[ L\left(\{u_i\}, \{v_j\}\right) := \sum_{(i,j) \in T} (\langle u_i, v_j \rangle - R_{i,j})^2 + \lambda \sum_{i=1}^{n} \|u_i\|_2^2 + \lambda \sum_{j=1}^{m} \|v_j\|_2^2, \]

where \( T \) and \( R_{i,j} \) here are from the training set and \( \lambda > 0 \) is the regularization coefficient. Implement an algorithm to learn vector representations by minimizing the loss function \( L(\{u_i\}, \{v_j\}) \). Note that you may need to tune the hyper-parameter \( \lambda \) to optimize the performance.

- HW3 problem 4c

Compute \( \nabla_{u_k} L = \sum_{(i,j) \in T} \nabla_{u_k} (\langle u_i, v_j \rangle - R_{i,j})^2 + \ldots \)

\[ = \sum_{(i,j) \in T} \sum_{i=k}^{m} v_j (u_i^T v_j - R_{i,j}) + 2 \lambda u_k \]

\[ u_k = \left( \sum_{(i,j) \in T : i=k} v_j v_j^T + \lambda I \right)^{-1} \left( \sum_{(i,j) \in T : i=k} R_{i,j} v_j \right) \]

\[ v_k = \left( \sum_{(i,j) \in T : j=k} u_i u_i^T + \lambda I \right)^{-1} \left( \sum_{(i,j) \in T : j=k} R_{i,j} u_i \right) \]
Announcements

• HW3 problem 4c

Given \( \{(x_i, y_i)\}_{i=1}^{n} \),

\[ l(w) = \left( \sum_{i=1}^{n} (y_i - x_i^T w)^2 \right) + \lambda \|w\|_2^2 \]

What is \( \nabla l(w) \)?

What is \( \arg\min_w l(w) \)?

\[ \nabla l(w) = \left( \sum_{i=1}^{n} 2(y_i - x_i^T w)(-x_i) \right) + 2\lambda w \]

\[ = \sum_{i=1}^{n} 2x_i(x_i^T w - y_i) + 2\lambda w \]

\[ \nabla w l(w) = 0 \quad \left( \sum x_i x_i^T + \lambda I \right)w = \sum x_i y_i \]

\[ \hat{w} = \left( \sum x_i x_i^T + \lambda I \right)^{-1} \left( \sum x_i y_i \right) \]
Announcements

• HW3 problem 4c
Sequences and Recurrent Neural Networks

Machine Learning – CSE4546
Kevin Jamieson
University of Washington

November 30, 2017
Variable length sequences

Images are usually standardized to be the same size (e.g., 256x256x3)

Neural Network
Variable length sequences

Images are usually standardized to be the same size (e.g., 256x256x3)

But what if we wanted to do classification on country-of-origin for names?

Hinton

Scottish
English
Irish

Recurrent Neural Network
Variable length sequences

Recurrent Neural Network

Standard RNN

LSTM

Slide: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
TF*IDF

n documents/articles with lots of text

How to get a feature representation of each article?

1. For each document $d$ compute the proportion of times word $t$ occurs out of all words in $d$, i.e. term frequency

   \[ TF_{d,t} \]

2. For each word $t$ in your corpus, compute the proportion of documents out of $n$ that the word $t$ occurs, i.e., document frequency

   \[ DF_t \]

3. Compute score for word $t$ in document $d$ as $TF_{d,t} \log\left( \frac{1}{DF_t} \right)$
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

Two Hearted Ale - Input ~2500 natural language reviews

http://www.ratebeer.com/beer/two-hearted-ale/1502/2/1/

Reviews for each beer

Bag of Words weighted by TF*IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in d dimensions
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

**Two Hearted Ale - Weighted Bag of Words:**

Reviews for each beer

Bag of Words weighted by TF*IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in \( d \) dimensions
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

Weighted count vector for the \( i \)th beer:

\[
z_i \in \mathbb{R}^{400,000}
\]

Cosine distance:

\[
d(z_i, z_j) = 1 - \frac{z_i^T z_j}{||z_i|| ||z_j||}
\]

Reviews for each beer

Bag of Words weighted by TF*IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in \( d \) dimensions

Two Hearted Ale - Nearest Neighbors:
- Bear Republic Racer 5
- Avery IPA
- Stone India Pale Ale &\#40;IPA&\#41;
- Founders Centennial IPA
- Smuttynose IPA
- Anderson Valley Hop Ottin IPA
- AleSmith IPA
- BridgePort IPA
- Boulder Beer Mojo IPA
- Goose Island India Pale Ale
- Great Divide Titan IPA
- New Holland Mad Hatter Ale
- Lagunitas India Pale Ale
- Heavy Seas Loose Cannon Hop3
- Sweetwater IPA
Find an embedding \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \) such that
\[
\|x_k - x_i\| < \|x_k - x_j\| \quad \text{whenever} \quad d(z_k, z_i) < d(z_k, z_j)
\]
for all 100-nearest neighbors.

(10^7 constraints, 10^5 variables)

Solve with hinge loss and stochastic gradient descent. (20 minutes on my laptop) \( d=2, \text{err}=6\% \) \( d=3, \text{err}=4\% \)

Could have also used local-linear-embedding, max-volume-unfolding, kernel-PCA, etc.

**BeerMapper - Under the Hood**

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

---

Reviews for each beer

Bag of Words weighted by TF*IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in \( d \) dimensions
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

| Reviews for each beer | Bag of Words weighted by TF*IDF | Get 100 nearest neighbors using cosine distance | Non-metric multidimensional scaling | Embedding in d dimensions |
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

- Reviews for each beer
- Bag of Words weighted by TF*IDF
- Get 100 nearest neighbors using cosine distance
- Non-metric multidimensional scaling

Sanity check: styles should cluster together and similar styles should be close.
BeerMapper - Under the Hood

Algorithm requires feature representations of the beers \( \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \)

---

Reviews for each beer

Bag of Words weighted by TF*IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in \( d \) dimensions

Sanity check: styles should cluster together and similar styles should be close.
Other document modeling

Matrix factorization:

1. Construct word x document matrix of counts
2. Compute non-negative matrix factorization
3. Use factorization to represent documents
4. Cluster documents into topics

Also see latent Dirichlet factorization (LDA)
Previous section presented methods to embed documents into a latent space.

Alternatively, we can embed words into a latent space.

This embedding came from directly querying for relationships.

**word2vec** is a popular unsupervised learning approach that just uses a text corpus (e.g. nytimes.com)
Word embeddings, word2vec

Source Text

The quick brown fox jumps over the lazy dog.

Training Samples

(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

The quick brown fox jumps over the lazy dog.

(quick, the)
(quick, brown)
(quick, fox)

The quick brown fox jumps over the lazy dog.

(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)

The quick brown fox jumps over the lazy dog.

(the, quick)
(the, brown)
Training neural network to predict co-occurring words. Use first layer weights as embedding, throw out output layer

slide: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
Word embeddings, word2vec

Training neural network to predict co-occurring words. Use first layer weights as embedding, throw out output layer

slide: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
word2vec outputs

king - man + woman = queen

country - capital

slide: https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/
Active Learning, classification

Machine Learning – CSE4546
Kevin Jamieson
University of Washington

November 30, 2017
Impressive recent advances in image recognition and translation…
Impressive recent advances in image recognition and translation...

Challenges for large models:

1) An enormous amount of *labeled data* is necessary for training
Impressive recent advances in image recognition and translation…

Challenges for large models:

1) An enormous amount of labeled data is necessary for training

2) An enormous amount of wall-clock time is necessary for training
Example: Image recognition

airplane
automobile
bird
Example: Image recognition

- airplane
- automobile
- bird
Example: Image recognition

Nonadaptive label assignment

- airplane
- automobile
- bird
Example: Image recognition

Nonadaptive label assignment
Example: Image recognition

Nonadaptive label assignment

Adaptive label assignment

airplane
automobile
bird
Example: Image recognition

Nonadaptive label assignment

Adaptive label assignment

- airplane
- automobile
- bird
$x_1, x_2, \ldots$ i.i.d.

$x_j$ may depend on $\{x_i\}_{i<j}$

complexity (reliability/robustness, scalability/computation, etc)
Being convinced that data-collection *should be adaptive* is not the same thing as knowing *how to be adaptive*.
Third  “Maybe his second week will go better”
Second  “I’d like to see other people”
First  “The corrupt media will blow this way out of proportion”
- $n \approx 5000$ captions submitted each week
- crowdsourcing contest to volunteers who rate captions
- goal: identify funniest caption

newyorker.com/cartoons/vote
“It's amazing to think he started out in the lobby.”
“I thought all our plants moved to Mexico.”
"Be patient. He'll grow on you."

Which caption do we show next?

1) **Non-adaptive** uniform distribution over captions
2) **Adaptive**: stop showing captions that will not win
4-5 times fewer ratings needed

Which caption do we show next?

1) **Non-adaptive** uniform distribution over captions
2) **Adaptive**: stop showing captions that will not win
Best-action identification problem

Objective: with probability .99, identify \( \arg \max_{i=1,\ldots,n} \mu_i \) using as few total samples as possible.

While the algorithm does not exit:
- The algorithm shows caption \( i \in \{1,\ldots,n\} \)
- Observe iid Bernoulli with \( \mathbb{P}(\text{"funny"}) = \mu_i \)

Stopping rule

Sampling rule
Best-arm Identification $n=2$

Consider $n = 2$ and flip coins $i = 1, 2$ to get $X_{i,1}, X_{i,2}, \ldots, X_{i,m}$

By a Chernoff Bound, if $\Delta = \mu_1 - \mu_2$ then

$$m = 2 \log(1/\delta) \Delta^{-2} \implies \hat{\mu}_{1,m} > \hat{\mu}_{2,m} + 2 \sqrt{\frac{\log(1/\delta)}{2m}} \implies \mu_1 > \mu_2$$

with probability $\geq 1 - 2\delta$
average # votes for “Funny”

confidence interval $\propto \sqrt{\frac{\log(n)}{\#\text{votes}}}$

Funny

$0 \ 1$

1 2 3

captions

n-1  n
confidence interval $\propto \sqrt{\frac{\log(n)}{\#\text{votes}}}$

more votes $\Rightarrow$ smaller intervals
confidence interval \( \propto \sqrt{\frac{\log(n)}{\#\text{votes}}} \)

keep sampling until intervals do not overlap
### Non-adaptive Algorithm

**# votes** Non-adaptive: 

\[
n \max_{i=1, \ldots, n} \Delta_i^{-2} \log(n)\]

### Successive Elimination

**[Even-dar…’06]**: 

\[
\sum_{i=1}^{n} \Delta_i^{-2} \log(n)\]

Stop sampling caption \( i \) as soon as no overlap
Learn an accurate classifier using a small number of labels

Find the winner of a competition using a small number of judgements

Very related to adaptive A/B testing

Find the ad that results in highest click-through-rate and keep showing it

Balance of exploration versus exploitation