FML Learning

Michael Walker
Student Research Forum
The goal is to try to predict whether an unseen example would be a “top” or “flop”
Research Question

- How well do these machine-learning algorithms classify FML examples?
  - **K-nearest neighbors**
    - Majority vote of nearest neighbor examples
  - **Bayesian Learning**
    - Sum of the probabilities of each of the features (individual words) of an example
  - **SVM**
    - Maximum margin separator, kernel trick
Data Used

- Automatically downloaded 883 examples from FMyLife.com
- Stop words and punctuation were removed as they were expected to be unimportant
- Each example consists of the text as a string, votes for “your life sucks”, and votes for “you deserved it”
Supervised Learning

- Measures the accuracy of the classification
- Split data into a test set and training set with distinct examples
- Accuracy is simply measured as:
  
  \[
  \text{Accuracy} = \frac{\text{(\# of examples from test set that are correctly classified)}}{\text{(total \# of examples in test set)}}
  \]
K-nearest neighbors

- Find the k nearest neighbors to the examples we want to classify
- Each neighbor then votes (either “top” or “flop”) and Q is classified according the majority vote
K-nearest neighbors

Example

- To find the k=3 nearest neighbors in the examples below, find the 3 examples that have the most words in common with the query:

kNearestNeighbors(“dog ran away”) = [ A, B, C ]

A: (“cat ran away”, top) – 2 words in common
B: (“dog ate my shoe”, top) – 1 word in common
C: (“dog bit girlfriend”, flop) – 1 word in common
D: (“got bad haircut”, top) – 0 words in common

- Take the majority vote: 2 “tops” and 1 “flop” so the query is classified as a “top”
K-nearest neighbors

Results after 15 fold cross-validation:

- Accuracy: 60.1%
- Outperforms baseline accuracy of 52%
  (classifying every examples as “top”)
Baysian Learning

- For all words found in all of the examples, find the probability that each word will be found in a “top” example or a “flop”
- For each word in the query example Q, lookup it’s associated probability
- Sum of the probabilities of the individual words in Q to find the overall probability
Baysian Learning

To classify string "dog ran away":

\[
P(\text{"dog"}) + P(\text{"ran"}) + P(\text{"away"}) = \frac{48}{65} + \frac{63}{97} + \frac{13}{33} = \frac{124}{195} = 63.6\%\]

So this example would be classified as a “top”
Baysian Learning

Results after 15 fold cross-validation:

- Accuracy: 59.3%
- Outperforms baseline accuracy of 52% (classifying every examples as “top”)
Support Vector Machine

- Key advance: **maximum margin separator**
Support Vector Machine

- Key advance: **kernel trick**

Data before and after being mapped into a three-dimensional input space \((x_1^2, x_2^2, \sqrt{2} x_1 x_2)\)
Data Reformatting for Weka

- Weka cannot analyze strings
- String need to be converted to a numeric format with StringToWordVector
- StringToWordVector records which words appear in the string and in what order as a series of numbers
Support Vector Machine

Results after 15 fold cross-validation:

- Accuracy: 59.9%
- Outperforms baseline accuracy of 52% (classifying every examples as “top”)