LING 520: Computational Analysis of English

Semester: FALL '16

Instructor: Sowmya Vajjala

Iowa State University, USA

13 October 2016

Class Outline

- Text Classification: Review of tuesday
- Naive Bayes classifier
- K-Nearest neighbour classifier
- Text classification and NLTK

What is text classification?

- Assuming we have some example texts which have some pre-defined class/category labels,
- text classification has this goal: developing a "model" of categorization based these example texts (training data)
- ... and using this model to assign categories to new texts.

Text Classification - Process

- Step 1: You have some collection of texts labeled with some categories, which you will use as training data for your model

Text Classification - Process

- Step 1: You have some collection of texts labeled with some categories, which you will use as training data for your model
- Step 2: You design some features that you think can distinguish between these categories (kitchen sink strategy or hand-crafted)

Text Classification - Process

- Step 1: You have some collection of texts labeled with some categories, which you will use as training data for your model
- Step 2: You design some features that you think can distinguish between these categories (kitchen sink strategy or hand-crafted)
- Step 3: Convert those texts into feature vectors.

Text Classification - Process

- Step 1: You have some collection of texts labeled with some categories, which you will use as training data for your model
- Step 2: You design some features that you think can distinguish between these categories (kitchen sink strategy or hand-crafted)
- Step 3: Convert those texts into feature vectors.
- Step 4: Develop or use an existing learning algorithm that can learn a classification function based on the values of all these features for the texts

Text Classification - Process

- Step 1: You have some collection of texts labeled with some categories, which you will use as training data for your model
- Step 2: You design some features that you think can distinguish between these categories (kitchen sink strategy or hand-crafted)
- Step 3: Convert those texts into feature vectors.
- Step 4: Develop or use an existing learning algorithm that can learn a classification function based on the values of all these features for the texts
- Step 5: Evaluate the classification based on some measure. Tune your classifier with better features, better learning algorithm or with more data (or all of those)

Text Classification - Process

- Step 1: You have some collection of texts labeled with some categories, which you will use as training data for your model
- Step 2: You design some features that you think can distinguish between these categories (kitchen sink strategy or hand-crafted)
- Step 3: Convert those texts into feature vectors.
- Step 4: Develop or use an existing learning algorithm that can learn a classification function based on the values of all these features for the texts
- Step 5: Evaluate the classification based on some measure. Tune your classifier with better features, better learning algorithm or with more data (or all of those)
- Step 6: Stop when you are satisfied, and deploy your classifier in some real world application

Text Classification - Process

- Step 1: You have some collection of texts labeled with some categories, which you will use as training data for your model
- Step 2: You design some features that you think can distinguish between these categories (kitchen sink strategy or hand-crafted)
- Step 3: Convert those texts into feature vectors.
- Step 4: Develop or use an existing learning algorithm that can learn a classification function based on the values of all these features for the texts
- Step 5: Evaluate the classification based on some measure. Tune your classifier with better features, better learning algorithm or with more data (or all of those)
- Step 6: Stop when you are satisfied, and deploy your classifier in some real world application
  (and discover all that research still does not result in material benefit!)

Measuring Success in Learning

Multiple ways. Depends on the nature of your dataset, and your application.

- Prediction accuracy on test set: typically used in most ML evaluation for text, images, videos, all sorts of things
- False positive rate (Type 1 Error), False negatives (Type 2 error) - typically in medical applications
- Precision (TP/(TP+FP)), Recall (TP/(TP+FN)), F-score (2PR/(P+R)) - typically in information retrieval, text classification
- Revenue increase - in e-commerce applications

Some commonly used features in text classification

- ngrams (word, character, POS, mixed representations)
- specific hand-crafted features: e.g., number of spelling errors, number of dependent clauses per clause, number of preposition phrases per sentence etc.
- feature representation: binary (presence or absence), count (number of occurrences), ratios etc.

Some commonly used learning algorithms

- Naive bayes classifier
- K-nearest neighbors classifier
- Logistic regression
- Decision trees
- Random forests
- Support vector machines
- neural network classifiers

.. etc.

Note: I will only give an overview of how these work. Details are found in machine learning classes.

Naive Bayes classifier

Naive Bayes Classifier

- Let us say I have a collection of emails (E1, E2 ... En). My problem is to classify them as spam or non-spam.
- Let us assume I already have some training data of 1000 emails labeled as Spam, 1000 labeled non-spam.
- Bayes classifier solves the text classification problem using bayes rule. For some email E1
  P(spam|E1) = P(spam)*P(E1|spam)/P(E1)
  P(non-spam|E1) = P(non-spam)*P(E1|non-spam)/P(E1)
- if first probability is higher than second, the email is spam. Else, it is non-spam.
- Since this is a comparison, we can ignore the denominator.

Naive Bayes - continued

Let us take individual terms:

- P(spam), P(non-spam): prior probability of seeing a spam or non-spam message. If your training data has 400 spam and 100 non-spam messages, what are P(spam) and P(non-spam)?

Naive Bayes - continued

Let us take individual terms:

- P(spam), P(non-spam): prior probability of seeing a spam or non-spam message. If your training data has 400 spam and 100 non-spam messages, what are P(spam) and P(non-spam)?
- P(E1|spam),P(E1|non-spam): likelihood that the email is actually spam or non-spam based on our training data. How do we get this?
- If we take a "bag of words" approach, and consider each word as a feature, each unique word in the email becomes a feature.
- If an email has only two words: "my mail", P(E1|spam) = P(my|spam)*P(mail|spam). P(E1|non-spam) = P(my|non-spam)*P(mail|non-spam).

Naive Bayes - continued

Let us take individual terms:

- P(spam), P(non-spam): prior probability of seeing a spam or non-spam message. If your training data has 400 spam and 100 non-spam messages, what are P(spam) and P(non-spam)?
- P(E1|spam),P(E1|non-spam): likelihood that the email is actually spam or non-spam based on our training data. How do we get this?
- If we take a "bag of words" approach, and consider each word as a feature, each unique word in the email becomes a feature.
- If an email has only two words: "my mail", P(E1|spam) = P(my|spam)*P(mail|spam). P(E1|non-spam) = P(my|non-spam)*P(mail|non-spam).
- If an email has 100 words, P(E1|spam) and P(E1|non-spam) are products of 100 conditional probabilities. You assign E1 to spam if P(E1|spam) is higher than P(E1|non-spam) and vice-versa.

Naive Bayes - conclusion

- Assumption: Each feature is independent of the other.
- There is no in-built way to account for inter-correlation between features
- So, this assumption does not really tell the whole story about what is happening. But it works for predictive modeling!

K-nearest neighbors classifier

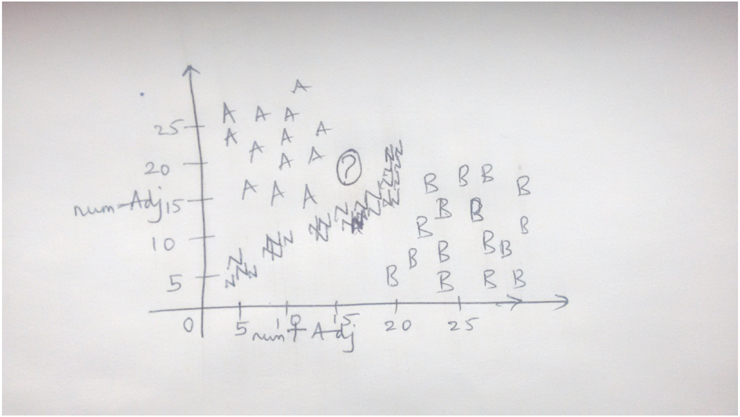
k-NN classifier

- Idea: A document belongs to the majority category among its k-neighbors.

k-NN classifier

- Idea: A document belongs to the majority category among its k-neighbors.
- Let us say my classification problem is: classifying movie reviews into three groups - positive, negative, neutral.
- My training data: say 500 examples for each of these categories.
- Let us say I am using only two features: Use of positive adjectives, Use of negative adjectives
- If I say my k is 5, when I have to classify a new review, and 3 of its neighbors on this feature space have category "positive", 1 has "negative", 1 has "neutral", I will choose "positive" as the category for this new review, because majority of my k neighbors have "positive".
- What is neighborhood? - any measure of distance.

k-NN classifier - 2D example



kNN - conclusion

- Also called "instance based classifier" or "lazy learner"
- Does not really have a "model" or "function". All computation of near-ness or far-ness happens during actual classification
- If you have large amounts of training data, and large feature set, this will become extremely slow.
- selecting k is heuristic.
- relationship between features is till not considered. Features are considered independent of each other.

NLTK and Text Classification

- Follow 1.1 and 1.2 in Chapter 6 and try to understand:
    1. How to develop a classifier in Python using Naive Bayes algorithm
    2. What exactly are the features in the example there?
    3. What are the most informative features, and are they consistent between your and your neighbor's computer?
    4. What is the classification accuracy?
    5. Let us say you want to add one more feature - starting letter. How do you do that?

Next Week

- Brief overview of some more classification algorithms: Logistic Regression, Random Forests, Support Vector Machines
- LightSide text mining toolkit, and Assignment 4 Description
- Conclusion of text classification
- Recap of concepts so far + Tutorial on Friday evening (21st October)
- Submit Assignment 3 on time!