LING 120: Language and Computers

Semester: FALL '17

Instructor: Sowmya Vajjala

Iowa State University, USA

18 October 2017

Outline

1. Text classification: Introduction
2. Exercise on text classification
3. Next class: Spam classification/Sentiment analysis - examples of how to do text classification
4. Later: Evaluating text classification

What is text classification

- "Classification" in general refers to grouping entities into pre-defined set of classes, based on some properties.

What is text classification

- "Classification" in general refers to grouping entities into pre-defined set of classes, based on some properties.
- Text classification is one of the methods of processing textual data, where the purpose is to categorize the text into one of the pre-defined set of categories, based on the language used.

What is text classification

- "Classification" in general refers to grouping entities into pre-defined set of classes, based on some properties.
- Text classification is one of the methods of processing textual data, where the purpose is to categorize the text into one of the pre-defined set of categories, based on the language used.
- Let us say I have four categories of textual data: book reviews, movie reviews, electronics reviews and other reviews on amazon.com. The process of taking a review, and assigning it to one of these four categories - is text classification.
- Note: "Text" can be documents, sentences or even words.

Where do we get those ”pre-assigned” categories?

- sometimes, historical data (e.g., if we want to predict weather based on past trends)

Where do we get those "pre-assigned" categories?

- sometimes, historical data (e.g., if we want to predict weather based on past trends)
- sometimes, ask human annotators to categorize small collection of text (e.g., if I keep clicking spam for all spam I see in my inbox, after a while, those messages will be directly classified into spam folder)

Where do we get those "pre-assigned" categories?

- sometimes, historical data (e.g., if we want to predict weather based on past trends)
- sometimes, ask human annotators to categorize small collection of text (e.g., if I keep clicking spam for all spam I see in my inbox, after a while, those messages will be directly classified into spam folder)
- For most of the known classification tasks, there are some standard datasets one can use to develop classification models
- Eventual evaluation: when you actually use these classifiers somewhere, and you learn something.. or in ecommerce, if the user is satisfied, and revenue is increased.

Where is classification useful?

- detecting whether the new email you got is spam or not spam. (spam classification)
- automatically detecting whether a movie review is positive or negative (opinion mining, sentiment analysis etc.)
- identifying if a news article is about "sports" or "politics" or "cinema" or "science" (google)
- identifying whether a given word in the sentence refers to a person name or not.
- identifying if a group of words form a multi-word expression or not.
- identifying whether a insurance claim is valid or not
- identifying if a post needs urgent attention or can wait (reachout.com)

What is difficult about text classification?

- Let us take this problem of classifying English learners as: beginner, intermediate and advanced.
- Let us say we even have 1000 example texts for each category, classified by expert English teachers.
- Now, how do we go about developing a classifier for doing this automatically?

What is difficult about text classification?

- Let us take this problem of classifying English learners as: beginner, intermediate and advanced.
- Let us say we even have 1000 example texts for each category, classified by expert English teachers.
- Now, how do we go about developing a classifier for doing this automatically?
- Should we look at vocabulary? use n-grams? errors? somehow capture syntax?

What is difficult about text classification?

- Let us take this problem of classifying English learners as: beginner, intermediate and advanced.
- Let us say we even have 1000 example texts for each category, classified by expert English teachers.
- Now, how do we go about developing a classifier for doing this automatically?
- Should we look at vocabulary? use n-grams? errors? somehow capture syntax?
- Should we combine everything? How? How do we even work with 3000 documents and come up with patterns??

Text Classification - Applications

- Think for 5 minutes (may be in groups), and try to list with some applications of text classification you can think of

Text Classification - Applications

- Think for 5 minutes (may be in groups), and try to list with some applications of text classification you can think of
- Some examples: classifying learner errors into different types (spelling, non-spelling, for example); classifying the learners into proficiency levels; General applications: sentiment analysis of product reviews on amazon, grouping search results into categories, recommending news articles related to what you are reading etc.

some examples of text classification

- Classifying the text of a tweet into one of the 5 languages: English, French, German, Chinese, Arabic. (language identification)
- Predicting whether a review about a product on amazon.com is positive or negative (or neutral) about the product (sentiment)
- Whether a webpage's text is suitable for children or not.

... and so on.

What does it mean to ”learn” to classify?

What is Machine learning?

- If you show a computing machine several examples of something, it can "learn" the patterns in these examples and try to identify identical occurrences for new data.

What is Machine learning?

- If you show a computing machine several examples of something, it can "learn" the patterns in these examples and try to identify identical occurrences for new data.
- One example: If I show the machine several articles about Donald Trump, and several articles about Hillary, the machine should be able to learn some patterns seen in both these categories. Patterns can be use of certain words and phrases, use of statistics, syntactic structures in speeches etc.

What is Machine learning?

- If you show a computing machine several examples of something, it can "learn" the patterns in these examples and try to identify identical occurrences for new data.
- One example: If I show the machine several articles about Donald Trump, and several articles about Hillary, the machine should be able to learn some patterns seen in both these categories. Patterns can be use of certain words and phrases, use of statistics, syntactic structures in speeches etc.
- Basic setup for machine learning: we have access to some set of examples, called "training set". Our goal is to make the machine learn what we want it to learn from those examples.

What is Machine learning?

- If you show a computing machine several examples of something, it can "learn" the patterns in these examples and try to identify identical occurrences for new data.
- One example: If I show the machine several articles about Donald Trump, and several articles about Hillary, the machine should be able to learn some patterns seen in both these categories. Patterns can be use of certain words and phrases, use of statistics, syntactic structures in speeches etc.
- Basic setup for machine learning: we have access to some set of examples, called "training set". Our goal is to make the machine learn what we want it to learn from those examples.
- As an approximation of what it learnt, we test how it does on a "test set". If we are satisfied, we start using this learnt model in real life.

Types of Machine learning?

Broadly, there are two types of machine learning:

- Supervised learning: when we know our categories
- Unsupervised learning: when we want to find out hidden/unknown groupings.

Note: This is oversimplification. If you really want to know more, enroll in a machine learning course. Coursera has a great introductory course by Andrew Ng (great does not mean easy).

Types of Machine learning?

Can you think of one ”supervised” learning and one ”unsupervised” learning scenario for corpus data?

Types of Machine learning?

Can you think of one ”supervised” learning and one ”unsupervised” learning scenario for corpus data? Supervised learning: one example is classifying all news articles into either ”sports” or ”non-sports”
Unsupervised learning: one example is identifying what are the dominant topics discussed on Twitter in the past 10 days.

How does "learning" happen?

Two aspects:

- Designing features for the machine to learn
- Developing or using an existing learning algorithm that can learn a classification function based on the values of all these features.
- An example "function" is learning weights for individual variables in linear regression.

Feature Design

- What in our opinion can be useful properties to check patterns for a given classification problem?
- Let us take the problem of classifying an email into spam or non-spam. What can be useful properties to design such a system?

Feature Design

- What in our opinion can be useful properties to check patterns for a given classification problem?
- Let us take the problem of classifying an email into spam or non-spam. What can be useful properties to design such a system?
- One more: let us say we want to classify English writing into "beginner", "intermediate" and "advanced". What can be the possible things to look at?

Feature Design

- What in our opinion can be useful properties to check patterns for a given classification problem?
- Let us take the problem of classifying an email into spam or non-spam. What can be useful properties to design such a system?
- One more: let us say we want to classify English writing into "beginner", "intermediate" and "advanced". What can be the possible things to look at?
- All these properties that can be relevant to perform machine based classification are called "features".
- The process automating feature extraction from your data (text or any other form) is called feature engineering.

Feature Design continued

There are two ways of doing feature engineering.

- 1. Kitchen sink strategy: In Spam classification example, consider all words or bi/tri grams as features, and leave it to the learning algorithm to choose what works.
- Advantage: Easy to do feature engineering, because we do not have to worry about what among those features is relevant.
- 2. Hand-crafted: Choosing specific features such as: "Use of all caps", "use of words from list X" for the same problem.
- Advantage: It is easy to understand which features are useful for the classifier and which are not. Disadvantage: Coming up with such specific features can be time consuming.

In sum, What kind of "features" or "patterns" will the model learn?

- Word occurrences are the most commonly used patterns.
- We can also look at word sequences (Ngrams)
- Part of speech tag patterns
- All of them put together
- Or some other stuff, such as some specialized linguistic patterns (e.g., number followed by some preposition, three adjectives preceding a noun etc.)

Learning Algorithm

- Goal of a learning algorithm is to take a feature representation of the training data (texts) and come up with a function that can assign weights to individual features, and use this function to predict the category for any new text it sees.
- Let us say I have 3 features: num. Nouns, num. Verbs, num. Adjectives. I have two categories: A and B. I have 1000 example texts (500 labeled A, 500 labeled B).
- A learning algorithm can learn something like this:
  1. Prediction = 0.3*numNN - 0.9*numVB + 1.1*numADJ
  2. If Prediction <=1, category is A. else, category is B.

Note: This is just one example function I created from air. There are 100s of learning algorithms, and machine learning researchers come up with new ways to learn everyday.

How does the machine "learn" these patterns?

- Lot of machine learning algorithms are already in place to "learn" from several forms of data.
- Our job is to pick a couple of them and compare them with our data, and choose the best one.

How does the machine "learn" these patterns?

- Lot of machine learning algorithms are already in place to "learn" from several forms of data.
- Our job is to pick a couple of them and compare them with our data, and choose the best one.
- Good thing about this is: it is like driving a car. you do not have to know all the internal working details to drive it.
- Bad thing: you end up working with a black box.

What is left?

- How does this actually work, step by step? (Friday)
- What is a good measure of evaluating classification? (Monday)

Today's attendance exercise

On Canvas, go to Today's date. Understand the problem in the pdf attachment in the forum posting, and work on that. You can work in teams if you want.