Text Analytics

Bob Stine
Department of Statistics
University of Pennsylvania

http://www-stat.wharton.upenn.edu/~stine/

Statistical methods for the analysis of textual data have come of age. Techniques that allow you to mine textual data for underlying sentiments, scan for hateful or discriminatory language, or create predictive are now commonly available in modern software. These methods do more than simply count, though counting through millions of words is impressive in itself. Along with counting, algorithms developed in NLP, the field of natural language processing, provide richer grammatical and syntactic analysis, such as identifying parts of speech, sentence parsing (*e.g.*, subject and predicate), and named entity recognition (people and places). Modern software tools allow the routine use of these methods by non-specialists – at least those who have taken this course!

Textual data typically comes with other information. Modern data streams routinely combine text with the familiar numerical data that might be used, for instance, in a regression model. For example, real estate listings routinely combine the selling price of the property with a verbal description. Some descriptions include numerical data, such as the number of rooms, but many others only verbally describe the property, often using an idiosyncratic vernacular. Advances in text analytics allow us to convert this text into numerical features suitable for other statistical models. Unsupervised techniques are available to create features directly from text, requiring minimal user input. Because these constructions are unsupervised, the resulting features perform like typical regressors. Techniques range from naïve to subtle. One can simply use raw counts of words, form principal components from these counts, or build regressors from counts of adjacent words. We will consider several examples to illustrate the surprising success of these methods. To partially explain the success, we will explore proposed hierarchical generating models often associated with nonparametric Bayesian analysis. Because regressors derived from text may be difficult to interpret, we also show how to develop interpretive hooks from quantitative features.

This course is self-contained with no explicit prerequisite beyond familiarity with statistical methods at the level of regression analysis. That said, some familiarity with multivariate methods (particularly principal components) and exposure to probability models would be helpful. The course will predominantly use packages from R as the main software with references to other tools (such as the Python-based NLTK) that may be helpful for automating certain chores.

Monday Introduction and Natural Language Processing (NLP)

The first day starts with an overview of the course and then introduces essential methods for getting, handling, and manipulating text. We'll look at simple tasks such as preparing text data for analysis and more elaborate processes such as acquiring tweets. Regular expressions can be very helpful for these tasks. Then we will delve into various types of data that you might consider, ranging from data with a natural response (permitting use of supervised training) to more descriptive analyses that can lead to hypotheses for subsequent analysis (*e.g.*, cluster analysis). Data with a response include product ratings or advertisements (such as movie reviews or real estate listings) in contrast to text from, say, a Twitter feed for which there's no built-in response.

We will spend much of the first day getting familiar with the `tm` package in R, covering important methods in NLP that include

- Tokenizing (handling rare words, smoothing, Zipf distributions)
- Regular expressions
- Stemming
- Part-of-speech tagging, parsing
- Document-term matrix
- Word clouds
- Named entity recognition (NER)

Time permitting, we will use clustering methods (such as *k*-means) to locate similar documents derived from these characteristics in the context of a running analysis based on tasting notes for thousands of wines.

Related references

Bird, S, E Klein, E Loper (2009) *Natural Language Processing with Python*. O'Reilly.

Feinerer, I, Kurt Hornik, and David Meyer (2008) Text mining infrastructure in R, *Journal of Statistical Software*, **25**.

Grimmer, J, and BM Stewart (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*.

Hopkins, D and G King (2010). Extracting systematic social science meaning from text. *American Journal of Political Science*, **54**, 229–247.

Janssens, J (2015). *Data Science at the Command Line*. O'Reilly.

Jurasfsky, D and JH Martin (2008). *Speech and Language*, Prentice-Hall.

Lindberg, N (2015). Egrep for linguists. http://stts.se/egrep_for_linguists/egrep_for_linguists.html

Manning, CD and H Schütze (1999), *Foundations of Statistical Natural Language Processing*, MIT Press.

Morrot, G, F Brochet and D Dubourdieu (2001). The color of odors. *Brain and language*, **79**, 309-320.

Tuesday Sentiment analysis and regression models

We will use basic properties of text to build models for properties of items that produce or are associated with the text documents. The easiest of these models look for the certain words (or their synonyms) or the type of language and vocabulary. For example, how well do words in a document distinguish positive from negative product reviews on a web site, Republican from Democrat speeches, or red from white wines.

We will use basic regression models, such as least squares and logistic regression. With least squares, the objective is usually clear: minimize the squared prediction error. When classification is the objective, the objective is often less recognizable and includes

- Sensitivity, specificity, and the ROC curve
- Accuracy, precision, recall, and f1

as well as more sensitive metrics (perplexity and log likelihood) that are available when using a probability model. Many tools expect you to have thousands of documents, but what do you do if you only have a few, as in the classical analysis of the Federalist Papers? For that, we'll explore a variation on naïve Bayes.

Related references

Eshbaugh-Soha, M (2010). The tone of local presidential news coverage. *Political Communication,* **27**, 121–40.

Hu, M and B Liu (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168-177.

Ng, V, S Dasgupta, and SM Arifin (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of COLING/ACL*, 611–618.

Monroe, BL, MP Colaresi, and KM Quinn (2008). Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, **16**, 372–403.

Mosteller, F and DL Wallace (1963). Inference in an authorship problem. *J of the American Statistical Association*, **58**, 275–309.

Tang, H, S Tan, and X Cheng (2009). A survey on sentiment detection of reviews. *Expert Systems with Applications*, **36**, 10760–10773.

Tausczik, YR and JW Pennebaker (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *J of Language and Social Psychology*, **29**, 24–54.

Thomas, M, B Pang, and L Lee (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. *Proceedings of EMNLP 2006*, 327–335.

Yano, T, NA Smith and JD Wilkerson (2012). Textual predictors of bill survival in Congressional committees. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* 793–802.

Wednesday More extensive regression models

Once you realize the value of words in regression models, there's no reason to stick to linear equations. Many other techniques are available to exploit the large collection of features that can be constructed from text, such as

- Boosted trees, and K nearest neighbors (kNN)

To judge the performance of the methods, classification of a held-out test sample is the standard. For such prediction tasks, we will have to consider issues such as how to deal with words that have not been seen (out-of-vocabulary).

Related references

James, G, T Hastie, R Tibshirani (2013). *Introduction to Statistical Learning*. Springer (available on-line for free from author).

Monroe, BL, MP Colaresi, and KM Quinn (2008). Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, **16**, 372–403.

Thomas, M, B Pang, and L Lee (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. *Proceedings of EMNLP 2006*, 327–335.

Yano, T, NA Smith and JD Wilkerson (2012). Textual predictors of bill survival in Congressional committees. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* 793–802.

Thursday Word embedding

Word embedding treats text as a "bag of words" and represents each word as numerical data. Words are converted from a sequence of letters to a point in a high dimensional space. These are collectively known as 'word embedding' or 'spectral methods' (not to be confused with the analysis of time series). Surprisingly, models derived from embeddings typically produce better classifiers than models built from hand-built word lists, allowing modelers to avoid the heuristic selection of key words.

The conversion of text to numbers makes available familiar tools from multivariate statistics. Indeed, a popular approach known as 'latent semantic analysis' is essentially principal components analysis (PCA). The literature has a variety of ways for constructing these embeddings or so-called "eigenwords", and we will focus on an approach that characterizes these statistics as covariances. As in PCA, interpreting the resulting features can be a challenge. "Lighthouse variables" can help.

A theme for the lecture is the relationship between n-gram statistics and covariances and correlations. This connection originates from characterizing text as a stochastic process ("token space"). Key topics include

- Latent semantic analysis (LSA)
- Spectral methods, eigenwords
- Random projection
- Singular value decomposition
- Principal components analysis and regression (PCA, PCR)
- Canonical correlation analysis

Related references

Bellegarda, JR (2005). Latent semantic mapping. *Signal Processing Magazine, IEEE*, **22**, 70-80.

Deerwester, SC , ST Dumais, TK Landauer, GW Furnas and RA Harshman (1990). Indexing by latent semantic analysis. *JAsIs*, **41**, 391-407.

Halko, N, PG Martinsson, and JA Tropp (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, **53**, 217–288.

Landauer, TK, PW Foltz, and D Laham (1998). An introduction to latent semantic analysis. Discourse Processes, **25**, 259-284.

Maas, AL, RE Daly, P T Pham, D Huang, A Y Ng and C Potts (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 142-150.

Rubinstein, D, E Levi, R Schwartz and A Rappoport (2015). How Well Do Distributional Models Capture Different Types of Semantic Knowledge? *Proceedings of ACL.*

Schwarz, Turney, PD and P Pantel (2010). From frequency to meaning: vector space models of semantics. *J. of Artificial Intelligence Research*, 37, 141-188.

Friday Hierarchical and neural models

Word embedding ignores many of the features of text that we consider important to language. Although they work well, they are often not very satisfying. Probabilistic models – models that define a probabilistic process that generates language – can be more satisfying, albeit at the price of more calculation and complexity.

This class begins with a class of hierarchical probability models known as 'topic models' that partially explain the success of word embedding. Topic models describe text using a mixture of underlying latent variables known as topics. We'll see that text latent semantic analysis essentially recovers these topics – which we can also model directly. The situation resembles the way in which factor analysis explains principal components analysis.

Going further, neural networks have recent set the gold standard for certain problems in text modeling, such as speaker identification and speech processing. We'll consider a special example that offers an alternative way to find an embedding, namely word2vec. Probability models provide likelihoods that permit a different analysis from that afforded by spectral methods. In particular, neural networks can not only be used to develop classifiers, but also used to synthesize new examples of various textual styles.

Related references

Blei, D, A Y Ng and M I Jordan (2003). Latent Dirichlet allocation. *J of Machine Learning Research*, **3**, 993-1022.

Blei, D (2012). Probabilistic topic models. *Communications of the ACM*, **55**, 77-84.

B Grün and K Hornik (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, **40**, 1-30.

Karpathty, A (2015). The unreasonable effectiveness of recurrent neural networks, http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

McAuliffe, J D and D M Blei (2008). Supervised topic models. In *Advances in neural information processing systems*, 121-128.

Mikolov, T, I Sutskever, K Chen, G Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111-3119.

Paul, M J, and M Dredze (2011). You are what you Tweet: Analyzing Twitter for public health. In *ICWSM*, 265-272.

Zaremba, W and I Sutskever (2014). Learning to execute. ArXiv.