Analyzing CS 229 Projects

Michael Chang (mchang91)
Ethan Saeta (esaeta)

CS 229

1 Introduction

The goal of this project is to study the characteristics of CS 229 projects. We explore whether past projects have certain traits that distinguish them from other machine learning papers and whether projects can generally be clustered by topic. In doing so, we hope to determine whether it would be possible to predict the kinds of projects we will see this year.

In this paper, we first describe the data we use and how we process it. Then, we explore whether these questions can be answerwed using techniques for classification. Finally, we look at using unsupervised learning techniques to get more information about project topics.

1.1 Data Set

Our main data set is the CS 229 project reports from 2010 and 2011. We convert each PDF to plain text using the standard UNIX "pdftotext" utility and tokenize the text files, ignoring all nonalphabetic characters. (In particular, “229” is not a valid token in our model.) We run the Porter2 stemming algorithm[1] on each token, which improves results in all experiments; for conciseness, we have omitted the results of our experiments without stemming.

In our first classification experiment, we compare 229 projects to other works in machine learning. For this, we use papers published at NIPS in 2010 and 2011[2]. These papers are processed in the same way as the projects.

	# Documents
CS 229 2010	135
CS 229 2011	139
NIPS 2010	292
NIPS 2011	307

	CS 229	CS 229 + NIPS
# Unique Tokens	20,521	45,883
# Stemmed Tokens	14,374	35,010

Table 1: Statistics about our data set

Table 1 summarizes some statistics about the data set we used. The right table shows that stemming reduces the size of document vectors by 25-30%.

2 Classification Experiments

For these experiments, we use a multinomial event model to construct vectors for each document. That is, the j-th entry of $x^{(i)}$ represents the number of times word j in our vocabulary occurs in document i. We train an SVM with a linear kernel using liblinear[3]. We use 2010 projects and papers as our training set and 2011 projects and papers as our test set.

[1] http://snowball.tartarus.org/algorithms/english/stemmer.html

[2] http://nips.cc/Conferences/

[3] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

2.1 CS 229 Project vs. NIPS paper

For this experiment, we consider all 229 projects positive examples and all NIPS papers negative examples. The results of this experiment are shown in Table 2.

		Actual	
		CS 229	NIPS
Predicted	CS 229	133	12
	NIPS	6	294

Precision:	91.7%
Recall:	95.7%
Accuracy:	96.0%

Table 2: Results of CS 229 vs. NIPS classification

These results suggest that CS 229 papers are in fact very distinctive, even among work in machine learning. Looking at the data that was misclassified, we found that a lot of the 229 projects that were classified as NIPS papers either introduced new learning algorithms or used more advanced techniques (such as neural networks), whereas most 229 projects directly apply the techniques we learned in class to various fields.

2.1.1 Limiting the Number of Pages

Next, we explore how much the accuracy of our classifier depends on the fraction of the document we consider. Our goal is to see whether predicting if a document is a 229 project or a NIPS paper can be done effectively using only the first couple of pages, or whether the classifier needs to consider the entire document. Since 229 projects and NIPS papers are generally different lengths, we run the clasifier using a percentage of each document (starting from the beginning of the document). The results of these experiments are shown in Table 3, and a graph of precision, recall, and accuracy is shown in Figure 1.

	Positive		Negative				
% of Doc	True	False	True	False	Precision	Recall	Accuracy
20%	131	19	287	8	87.3%	94.2%	93.9%
40%	129	19	287	10	87.2%	92.8%	93.5%
60%	133	17	289	6	88.7%	95.7%	94.8%
80%	134	23	283	5	85.4%	96.4%	93.7%
100%	133	12	294	6	91.7%	95.7%	96.0%

Table 3: Results of classifier on limited number of pages

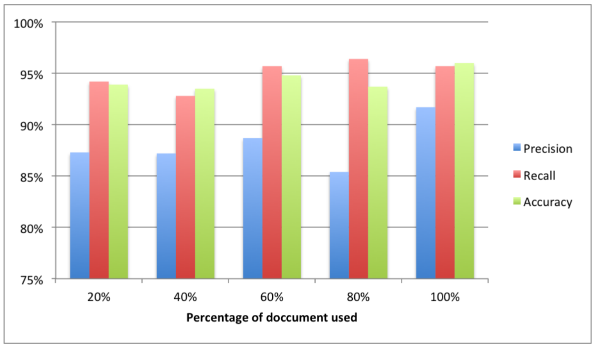


Figure 1: Graph of classifier accuracy

Since using just 20% of each document already yields precision, recall, and accuracy results above 85%, the first page or two of each document seems to be a very strong indicator of whether the document is a 229 project or NIPS paper. The lack of significant improvement at 40%, 60%, and 80% suggests that the middle pages of the document do not add substantial information that helps classify the document, while the spike in accuracy at 100% suggests that the last page is also a good indicator of document class. Thus, we conclude that the first and last page of each document are the best indicators of whether it is a 229 project or NIPS paper.

2.2 CS 229 Project Topic Classification

Next, we turn our attention to clustering 229 projects by topic. We pose this problem as a classification problem by manually labeling projects (based on their titles) according to their general field, such as “vision” or “robotics.” For this experiment, we choose to focus on “vision,” as we found a significant number of projects related to vision.

Using the same approach as above, we now label vision projects as positive examples, and all other projects as negative examples. Our training set (2010 projects) contains 23 positive examples (out of 135), and our test set (2011) contains 18 (out of 139). Results are shown in Table .

		Actual	
		Vision	Other
Predicted	Vision	10	3
	Other	8	118

Precision:	76.9%
Recall:	55.6%
Accuracy:	93.5%

Table 4: Results of classifier on vision projects

These results suggests that it is generally easy to identify a non-vision project, but some vision projects are difficult to identify. Much of this difficulty likely comes from the fact that the field of "vision" is ss large that vision projects from one year may not adequately represent the field. For example, there is a project in 2011 about digital image forensics. However, since there are no projects about image forensics in 2010, this project was misclassified.

3 Automated Clustering

The results above show a significant limitation in using classification to identify project topics: doing so assumes that the training set contains enough examples to accurately represent the topic. Another problem is that though many general topics, such as vision and robotics, show up each year, there are some topics that are only receive attention in one particular year. For example, in 2011, there are many projects about predicting the stock market using Twitter, whereas there is only one project in 2010 that mentions stocks. Using the method above would not work here, since there are no training examples to use to identify the topic.

Thus, we turn our attention to unsupervised techniques for identifying clusters within a single year, without depending on data from previous years. Running the standard K means algorithm[4] often results in all projects being placed in a single cluster. The biggest problem we identified came from in the minimization of $\left\|x^{(j)} - \mu_i\right\|$.

To solve this problem, we first let $x_j^{(i)} = 1$ if word j appears in project i, regardless of frequency. Then, we make the following modification to K means: set the cluster of project i to

$$c^{(i)} = \arg\max_j \left\langle x^{(i)}, \mu_j \right\rangle$$

Where $\langle x, y \rangle$ is the inner product of x and y.

This formula rewards projects for having words in common with the cluster centroid, without penalizing them for words they do not have or words that are irrelevant to the centroid. This makes sense intuitively, since projects will generally contain many words that do not indicate their topic, and not all words that indicate topic will appear in every project on that topic.

Running this algorithm with $K = 3$ often produces a small (< 20 project) cluster. However, whether such a cluster is produced and what topic it potentially represents are highly sensitive to the starting position of the cluster centroids. Therefore, we use the following approach:

[4] http://cs229.stanford.edu/notes/cs229-notes7a.pdf

- Initialize $M_{ij} = 0$ for all pairs of projects (i, j).
- Repeat N times (We used $N = 250$):
  - Run the above K means-like algorithm, with $K = 3$ and cluster centroids initialized to random projects.
  - For each cluster whose size is $< C$ (we used $C = 50$), for each pair of projects (i, j) in the cluster, increment M_{ij}.

This gives us a matrix containing the number of times each pair of projects appears in the same, relatively small cluster. For each project i, we can rank the projects it is most likely related to by sorting the projects j according to the values of M_{ij}.

To visualize the clusters, we represent each project as a node in a graph. We draw an edge between node i and node j if j appears in the top 5 ranking of i, and vice versa. (Requiring a mutually high ranking accounts for projects which are unrelated to many other projects and thus rank projects almost arbitrarily.) The result is shown in Figure 2. We have omitted nodes with no connected edges and highlighted interesting parts of the graph. A list of some of the projects in the graph is presented in Table 5.

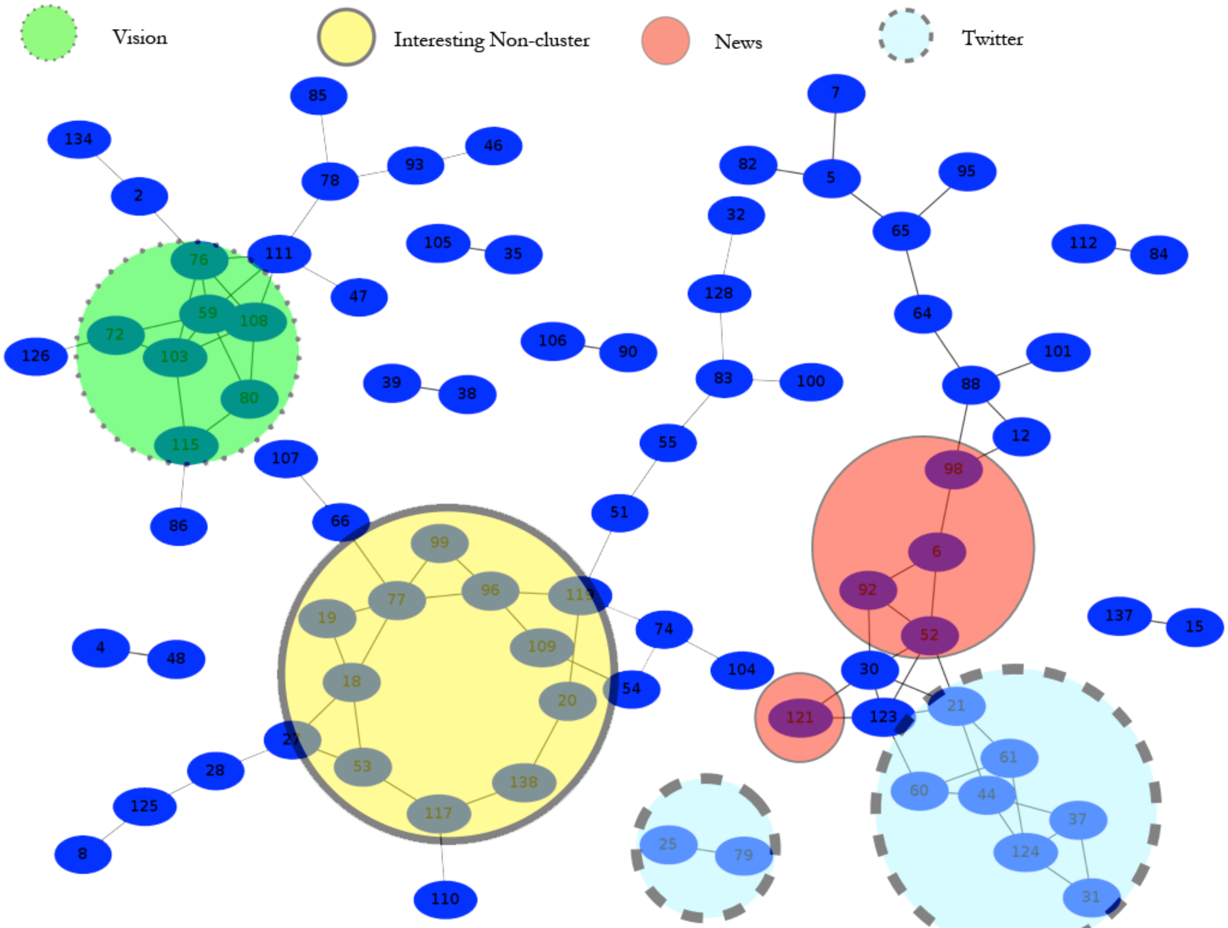
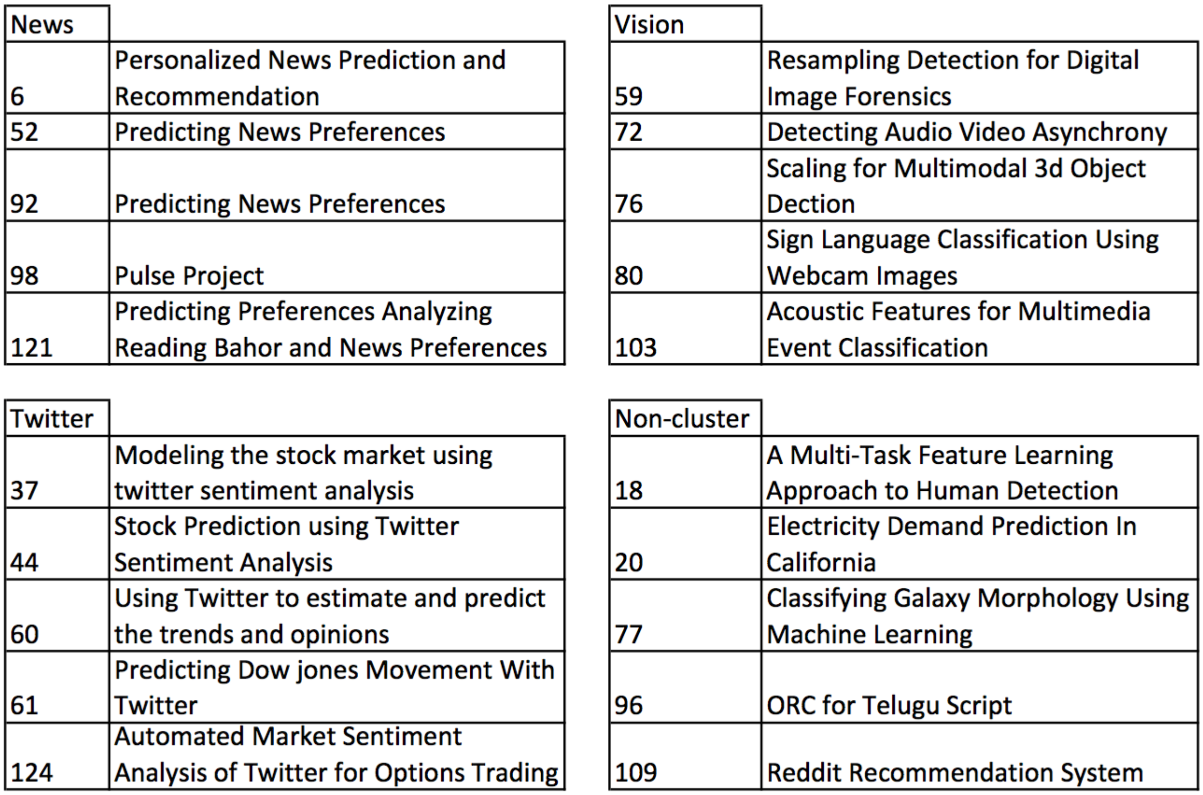


Figure 2: Graph of clustering results



News	
6	Personalized News Prediction and Recommendation
52	Predicting News Preferences
92	Predicting News Preferences
98	Pulse Project
121	Predicting Preferences Analyzing Reading Bahor and News Preferences

Vision	
59	Resampling Detection for Digital Image Forensics
72	Detecting Audio Video Asynchrony
76	Scaling for Multimodal 3d Object Dection
80	Sign Language Classification Using Webcam Images
103	Acoustic Features for Multimedia Event Classification

Twitter	
37	Modeling the stock market using twitter sentiment analysis
44	Stock Prediction using Twitter Sentiment Analysis
60	Using Twitter to estimate and predict the trends and opinions
61	Predicting Dow jones Movement With Twitter
124	Automated Market Sentiment Analysis of Twitter for Options Trading

Non-cluster	
18	A Multi-Task Feature Learning Approach to Human Detection
20	Electricity Demand Prediction In California
77	Classifying Galaxy Morphology Using Machine Learning
96	ORC for Telugu Script
109	Reddit Recommendation System

Table 5: Selected project titles

From the graph, we can see that our algorithm is able to detect a number of topic clusters. In particular, many projects about using Twitter to predict the stock market (labeled "Twitter") are connected to one another (we observed that M_{ij} for these projects tended to be in the 20s). Also, projects about vision were also connected, though M_{ij} values were slightly lower (around 10-15).

Another interesting feature of this graph is the ring, labeled "Interesting Non-Cluster." Although each project in the ring ranked its neighbors highly, the lack of edges that cross the ring suggests that these projects are not pairwise related in the same way as the other clusters. There was no noticeable topic relationship between projects in the ring.

4 Conclusion

The techniques we use are able to point out some interesting characteristics of CS 229 projects. We are able to distinguish between 229 projects and NIPS papers with very high accuracy, suggesting that 229 projects have a distinct place in work on machine learning, as they largely emphasize applications rather than new algorithms. Also, both our supervised and unsupervised techniques show that, while it is fairly easy to find similar projects, it is much more difficult to identify a significant fraction of the projects corresponding to a particular topic. This suggests that, although it may seem like many projects fall into the same general field, projects vary quite widely in how they apply machine learning, even within a single field. This fact, combined with the fact that some topics do not appear year after year, suggests that it would be rather hard to predict what sorts of projects students will come up with next year (for example, this one).