Building a Better Tour Experience with Machine Learning

Alan Guo, Chanh Nguyen, and Taesung Park

1. INTRODUCTION

The motivation of this project is to solve a problem that we currently face working on a project called 27bards, which seeks to revolutionize the tour industry by constructing a database of currently existing tour operators with the Airbnb-like twist of allowing the average person to make money as a part-time tour guide. We wanted to apply machine-learning techniques to make sense of the largely unanalyzed collection of tour operator websites and to ultimately create a rich tour-browsing experience for users.

One of the first challenges we faced was to build a classifier for detecting tour operator websites across the web. This was accomplished with relatively high accuracy using a Bag of Words model and an SVM classifier. We will not go into much detail of this task and instead discuss the more difficult problem of classifying the tour websites into multiple classes such as "cruise" or "hiking". The problem was not that the classes were too similar, but rather due to the nature of tour operator websites, which tend to mention affiliated tours on their websites, making it difficult to simply find a specific type of tour by simply looking for a keyword such as "hiking" since many tour websites will mention this term. Tours.com, one of the most comprehensive tour databases we found, lists a collection of tours across the world broken down by type. However, clicking on the first page of the "culinary" category yields a mixture of results—none of which are culinary tours. We wanted to build a system that can perform better than what's available on Tours.com.

Since the supervised data on Tours.com were unreliable, we turned to unsupervised clustering. We performed bisecting k-means clustering, and it worked reasonably well. For example, given three sets of handpicked websites on golf, culinary, and cruise, bisecting clustering gave 89.4% accuracy, which was better than the categorization by tours.com.

Lastly, we tried classifying websites based on its geographical location. There have been attempts to associate documents to geolocations. We devised a supervised classifier based on Wing et al. 2011. The nice thing about finding geolocation of a tour website is that we can recommend other tours in nearby regions, or show typical tour styles in a given region.

To summarize, we 1) apply Naïve Bayesian and SVM approach to simply build a binary classifier for tours and activities websites. 2) refine our training set and build both supervised/unsupervised multiclass classification algorithm for different type of tours (so we can 'tag'/categorize them), and 3) predict geolocation of a tour website using supervised learning.

2. BINARY CLASSIFICATION

Our first task was to build a system that allows for the collection of tour operator websites while rejecting the class of all other websites on the Internet. We scraped 885 known tour and activity sites and about 2,000 non-tour websites from Alexa and other sources.

We train both a Multinomial Naive Bayesian (MNB) classifier as well as SVM binary classifier. With regards to feature selection, we processed the data with stopwords and other techniques discussed in the following section on multi-class classification. We achieved fairly accurate results with the simple Bag of Words model (Table 1).

Training size	Features	Test Size	**NB Error**	**SVM Error**	Description
885	2173	885	**0.169**	**0.00**	Test on training set
885	2173	100	**.1900**	**.28**	Test on 100 samples
2887	12145	100	**.0300**	**.0300**	Increased training set size
2887	15501	200	**.0550**	**.0550**	Added more test data
2887	7555	200	**.0650**	**.0650**	Reduced number of features

Table 1: Results of binary classification show near equal performance between MNB and SVM.

3. MULTI-CLASS CLASSIFICATION

3.1) SUPERVISED CLASSIFICATION

Data and Model

In order to build our training data to categorize tour websites by type, we scraped all 2,000 websites from Tours.com. However, when we trained on this data and evaluated using a hold-out cross validation set of 30%, the error was somewhere near 56%, which we quickly found out was due to the mis-categorization of many sites on Tours.com. Other tour directories were too region-specific and so we had no ground-truth to work with. In order to reach our goal of building and evaluating a system with good performance, we chose to manually build a set of ground-truth samples for supervised learning. We collected 440 ground-truth samples in 8 categories by crawling through various tour directories on the web. When looking for tour websites of a certain category, our main criteria were the following:

1) Tour website must only offer a specific type of tour for a particular region.
2) Tour website must not be generic tour search service or directory, and must offer tours directly.
3) If a tour website met the above 2 criteria but did not fall into any of the 7 specific categories, it would be placed in the 8th category labeled "other."

Processing. Sometimes the main page of a tour website does not say much, whereas all the subpages may contain the majority of information. We followed the links to extract text from up to 20 pages on the same domain. We discard all mark-up and store the raw text only.

In order to improve the performance of all our learning systems, we stemmed the words using the Snowball library which runs the Porter stemming algorithm. This reduces related words such as "cruise," "cruises," "cruised," into "cruis" so that all three forms of the same word get recognized as the same feature.

We also noticed that many numbers and strange characters were slipping through as common features, and found a performance increase by restricting our word features to only alphabetical characters.

Feature Selection

Stopwords. We found a significant performance increase by removing common words such as "I" or "with" from the dictionary. The presence of these words, or lack thereof, usually have little to no correlation with the categories we are trying to distinguish and can only confuse the classifiers more.

Thresholding. In order to prevent rare terms from influencing the classifiers, we used a lower threshold and varied this number to determine the number of features. We also enforced an upper threshold on terms that appeared too often, which were bland terms such as "tour" and "travel." Thresholding also reduces the dimensionality of our feature vectors, reducing the average number of unseen features and thus the effects of smoothing.

A drawback of thresholding is that some features representing a certain category are more likely to fall under a threshold if that category is underrepresented in the training data. To counteract this, we gave documents with lower representation a boost directly proportional to category's training size.

Bigrams. Realizing that some of the most distinguishing features are phrases, we extracted bigrams and tuned their weighting such that they would be able to compete with single-word features, which have the advantage of appearing much more often. While increasing the number of classes to 8 brought the accuracy of our system down to about 70%, using bigrams helped bring it back up to about 86%.

Multinomial Naïve Bayes

In order to get a binary classifier to classify amongst several classes, we took an all-vs-one approach. For each class, we marked all samples in that class as positive, and samples in all other classes as negative. We then trained a Multinomial Naïve Bayes classifier whose job was to predict the probability of a sample belonging to that class. Then, for any given sample, we run it against each classifier and take the classifier with the highest confidence.

Multi-class Decision Trees

In order to discover possible improvements, we wanted to compare the performance of our all-vs-one classifier to that of a multi-class binary decision tree. As shown in Figure 1, we learned that while both classifiers struggled with the addition of classes, the decision tree performs better when there are fewer classes, but the Multinomial Naïve Bayes is preferred when there are more than five.

Table 1 Our system detects tour websites against other websites well in binary classification. In telling tour websites against each other, our manually labeled ground truth was critical for decent performance.

Description	Training size	Features	Test Size	Error
Binary classification	2887	7555	200	.065
8-class (Tours.com)	1843	1843	784	.56
8-class (Manually labeled)	312	1179	128	.14

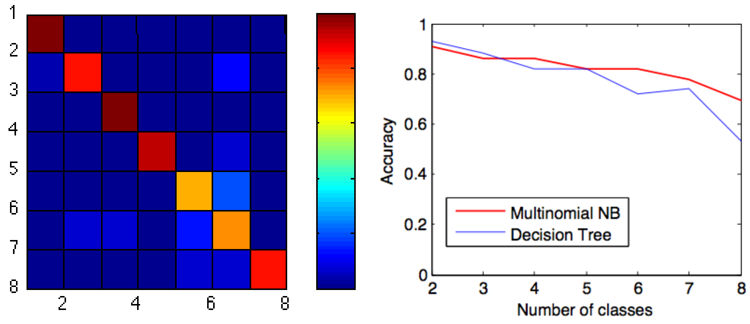

Figure 1: a) Classifier response to number of classes. b) Performance of final system with bigrams and feature threshold tuning.

Evaluation

We tested all of our classifiers using 30% hold-out cross validation and found that the system was able to perform reasonably well when trained on reliable ground truth data with good feature processing and selection. Understandably, accuracy begins to degrade sharply with the use of more than 8 classes due to higher chances of overlap.

3.2) UNSUPERVISED CLUSTERING

Since the labeled training set from tous.com were not reliable, we also tried unsupervised clustering for type classification. We initially started with basic k-means clustering on simple bag-of-words feature vectors with Euclidean distance measure. However, this approach did not yield high accuracy, because the Euclidean distances become indiscernible as dimensions increase (Beyer et al, 1999). Realizing that, our final implantation uses bisecting k-means clustering on refined feature vectors with cosine distance measure.

Feature Selection

First, we start with all words that appear in the websites listed on tours.com. Next, the words are stemmed. Then we chose 2000 words in the mid-frequency range. For example, we filtered away words like *the*, or *tour* that are too common to have discriminating power, and also rare words such as *Uvaggio* (a type of wine) that does not appear in most tour websites. Next was to sort the terms according to its variance on frequencies on documents. The idea is that the terms that appear equally frequently in all websites are not good classifiers. Let f_i be the frequency of term t in document d_j. The variance of term t is defined as

$$q_0(t) = \sum_{j=1}^{n} f_j^{\,2} + \frac{1}{n}\left(\sum_{j=1}^{n} f_j\right)^2.$$

We sorted the terms by variance in descending order, and took the first one thousand. Lastly, we manually throw away the terms that were meaningless or related to geographical locations. For example, the words we threw away include *sans-serif* (a font type), *buy*, and *???*(caused by text encoding error). We also filtered out geographical proper nouns because we did not want the clustering to be performed by regions, but tour types. In the end, we had 411 feature terms.

Feature Vector Representation

Each website is represented as a bag-of-words model on the feature dictionary we created. Then each entry is weighted by *tf-idf* to weight frequent words with less discriminating power. Finally, in order to account for documents of different lengths, each document vector is normalized so that it is of unit length. The distance measure in this space is calculated using cosine distance, not Euclidean.

Clustering Algorithm

We followed the suggestion by Steinbach et al 2000 on comparison of document clustering techniques, which recommends bisecting k-means clustering. The algorithm is very simple once we have the regular k-means clustering algorithm:

1. Pick a cluster with largest size.
2. Find 2 sub-clusters using the basic k-means algorithm. (Bisecting step)
3. Repeat steps 1 and 2 until the desired number of clusters is reached.

The number of clusters was chosen empirically. 3 to 8 clusters produced the most consistent websites collections.

Result

Using a hand-picked set of websites on three categories, *golf, culinary,* and *cruise*, we ran bisecting k-means algorithm into three clusters.

		Predicted		
		Golf	Culinary	Cruise
Actual	Golf	41	1	3
	Culinary	3	52	5
	Cruise	2	2	42
	Accuracy	0.891	0.964	0.840

Table 3: Confusion matrix of bisecting k-means

Furthermore, we collected a few terms that characterize each cluster. They were gathered by choosing the terms with highest values of (*frequency of the term in the centroid*) – (*maximum frequency of the term in all other centroids*). These terms can be used as an alternative to categories in supervised learning. Suppose you query the database of tour websites on tours.com, with the keyword "Mediterranean". 405 websites are returned in our training set. Clustering those into 5 categories yields characterization terms in Table 4. As you can see, the terms reveal the nature of the clusters. Thus, the characterization terms can be used as an alternative to the fixed categories in supervised learning. Moreover, they are more flexible than the fixed set of types in supervised learning.

Table 4 Characterization terms of clustered websites searched by "Mediterranean". The terms were stemmed.

Cluster 1	yacht, charter, sail, gullet, cabin, motor, boat
Cluster 2	Bike, wine, cycl, cook, hike, empir, ride
Cluster 3	Cruis, wildlife, dive, costa, masai, mara, pacif
Cluster 4	Overnight, dinner, templ, church, Christian, hotel, jewish
Cluster 5	Thank, testimony, axum, empir, camel, desert, women

4. Geolocation Identification

Another kind of categorization widely used in tour industry is classifying by location rather than types. Most of tour portals such as orbitz.com have that. Unfortunately, geographical categorization has been done manually by tour operators. Recently, there have been some attempts using machine learning that tried to classify documents by corresponding regions. We took the approach by Wing et al. 2011.

Supervised Data and Model

For the training set, we used the destination tagging on tours.com. Tours.com lists what regions the tour is offered for each tour website. We fetched the lists and used them as ground truth geolocations for the websites. The labeling on tours.com is not accurate, but manually identifying the geolocations of tour websites enough for training in all geolocations in the world was next to impossible. This list gives a mapping from website URLs to its geographical names. Then we fetched the list of countries and cities with their latitude and longitude on various internet sources. Now we can construct a mapping from website URLs to latitudes/longitudes. Note that this is a one-to-many relationship because a tour website usually offers tours in more than one geolocation.

Feature Selection and Feature Vector Representation

We took the same set of feature terms we used in bisecting k-means clustering. The only difference is that we did not filter out the terms referring to geolocations, because they are good indicators of identifying tour regions.

Method

First, we create a grid of latitude/longitude. We used a square grid of size one degree. For each grid, we can find the websites associated with that grid in the training set. We concatenate all website contents for that grid so that we can create a frequency-based feature vector for the grid. The probability of a word w_j in document d_k is simply the count ratio.

$$\theta_{j,d_k} = \frac{\#(w_j, d_k)}{\sum_{w_l \in V} \#(w_l, d_k)}$$

Similarly for a grid cell c_i, we can compute the word distribution.

$$\theta_{c_i,j} = \frac{\sum_{d_k \in c_i} \#(w_j, d_k)}{\sum_{d_k \in c_i} \sum_{w_l \in V} \#(w_l, d_k)}$$

The distributions were smoothed by Laplace smoothing.

Now that we have word distribution of documents and cells, identifying the geolocation of a given website is just searching for the cell with most similar word distribution to that of the website. Therefore, we need to establish a similarity measure between two probability distributions. We used Kullback-Leibler divergence for that.

$$KL(\theta_{d_k}|\theta_{c_i}) = \sum_{w_j \in V} \theta_{j,d_k} \log \frac{\theta_{j,d_k}}{\theta_{c_i,j}}$$

The cell c_i with the smallest KL divergence is the predicted geolocation of the website.

Result and Applications

We tested on 74 handpicked websites that refer to no more than two countries. If the predicted geolocation is either one of the two countries, that prediction is marked as correct. 60 of the prediction were correct, which gives the accuracy of 81%. Errors were largely due to the fact that our training set was not perfect, as many labeling on tours.com were wrong. Also, some errors were caused by the websites that refer to regions our latitude/longitude collection did not cover, such as Cuba.

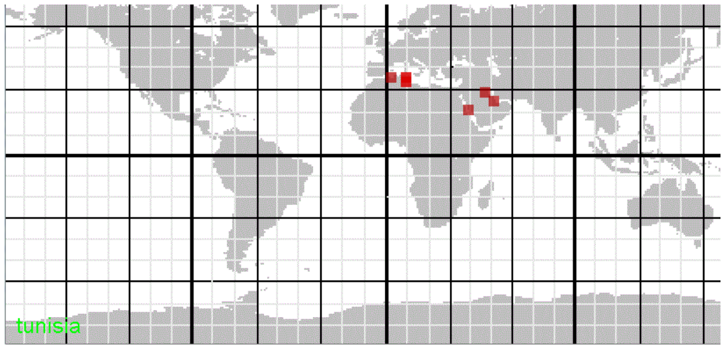

Figure 2 Distribution of likelihood of a website tunisiadiscoveries.com. Lighter red squares mean higher likelihood. Note that because of the term *desert*, other desert countries have high likelihood as well.

The word distribution has other applications. For example, you can visualize what the hot keywords for a tour in a specific region are by showing the terms with high probabilities, as in Figure 3. If you were wondering where to go for ultimate cruise/yacht tour, you can superpose the word distributions of *cruise, yacht,* and *dive*, as in Figure 4. Moreover, by storing the geographical location of websites by latitude/longitude rather than place names, we can recommend other popular tours in nearby regions.

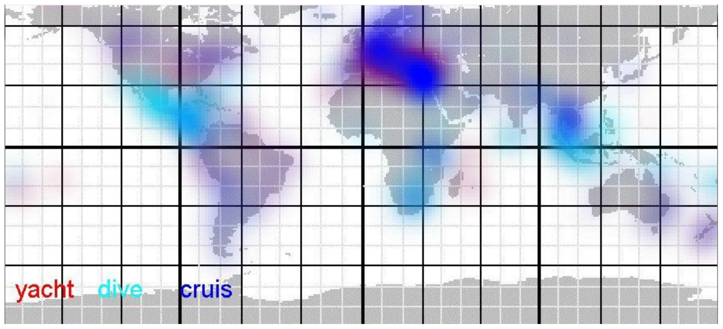

Figure 3 The word distribution of *yacht*, *dive*, and *cruise*. The distribution of *yacht* is in red. You can observe that these regions are gathered along popular beach destinations.

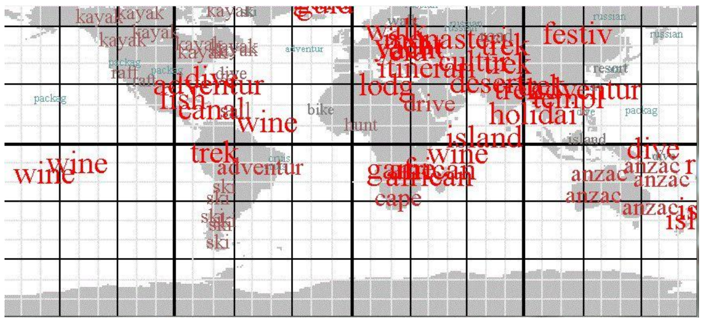

Figure 4 Hot keywords in different regions.

5. REFERENCES

[1] Qi & Davidson. *"Web Page Classification: Features and Algorithms"*. http://www.cse.lehigh.edu/~xiq204/pubs/classification-survey/LU-CSE-07-010.pdf

[2] Shen et al. *"Web-page Classification through Summarization"*. Microsoft Research in Asia. http://research.microsoft.com/pubs/67806/18.pdf

[3] B. Wing, J. Baldridge. *"Simple Supervised Document Geolocation with Geodesic Grids."* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 955–964, Portland, Oregon, June 19-24, 2011.

[4] I. Dhillon , J. Kogan , C. Nicholas. *"4 Feature Selection and Document Clustering."* http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.136.799

[5] M. Steinbach, G. Karypis, V. Kumar. *"A Comparison of Document Clustering Technique."* University of Minnesota.

[6] K.Beyer, J.Goldstein, R.Ramakrishnan, U.Shaft, *"When Is "Nearest Neighbor" Meaningful?",* 1999