Clustering a Customer Base Using Twitter Data

Vanessa Friedemann

***Abstract*—This paper presents a method for clustering customers of a company using social media data from Twitter. Retail and market analysis using social media has become a promising field for large enterprise companies. Applications include customizing advertising campaigns, localizing unexplored market segments, and projecting sales trends. The technique outlined in this paper scrapes publicly-accessible Twitter data and constructs features. These features are clustered using a similarity measure to produce groupings of users. This method performs well using the sample data set provided and exhibits potential to further improve given access to more data.**

***Keywords*—unsupervised learning, k-means, PCA, clustering, social media, customers, market segmentation, retail.**

I. Introduction

Applications of Clustering in Retail

There are numerous applications within the retail industry for clustering large populations. Clustering a company's customers allows marketing teams to tailor advertising messages for specific groups of like-minded people with similar interests. Clustering a competitor's customers, or the market as a whole, helps a company to identify untapped niches into which it can expand. Further, customer clustering can feed into recommendation systems to suggest items that "similar" users purchased. According to Forbes magazine, "89% of business leaders believe analytics will revolutionize business operations" [1].

A burgeoning area of research in the market analysis field involves using publicly-accessible social media data. The analytics website ResearchAccess states that "social media can be a value-add to traditional recruitment strategies" [2].

About This Paper

Our approach uses publicly available Twitter data to perform customer clustering for a chosen company, Nike. We first harvest the data from Twitter using the open source Tweepy package in Python [3]. For efficient storage and querying, we store this data into a local SQLite database.

We start with features selected from the data but then prune and transform into a lower dimensional feature space using principal component analysis (PCA). These features are passed into the k-means unsupervised learning algorithm to segment the samples into clusters. We then determine the appropriate number of clusters by performing a quantitative analysis of the resulting intra-class variances and inter-class distances.

Section II of this paper discusses related work involving social media data and improvements on the standard k-means algorithm. Section III details the data and features used in this paper. Section IV elaborates on the k-means clustering technique and parameter selection. In this section we also develop a quantitative metric to benchmark the quality of clustering. Section V presents the results of our algorithm.

Why Twitter data?

Large-scale private-sector data, such as sales history and loyalty account information, is prohibitively difficult to obtain for persons unaffiliated with the company to which the data pertains. For any company in need of information regarding customers other than its own, there is a need for an alternative. A key assumption we make is that a user who follows a brand on Twitter is a customer of that brand. Although Twitter accounts lack some basic information such as gender, they allow us to see other brands and public figures in which a customer has indicated interest. This information helps to create a more holistic view of the customer. We therefore consider Twitter data to be a reasonable proxy for customer data when the latter is unavailable.

II. Related Work

Significance of Social Media Data

Past work has found that data scraped from social media is a meaningful reflection of the human behind the account. Using Twitter data, Bergsma, Drezde, et al. were able to successfully predict hidden features such as gender and ethnicity by clustering on observed attributes such as first name, last name, and friends list [4]. A study from the IBM Haifa Research Lab demonstrated that "using the same tags, bookmarking the same web pages, [and] connecting with the same people" were all features that led to like-minded clusters [5]. A Pennsylvania State University research project partitioned users based on their levels of connectedness and engagement on social media, and showed that there was significant difference amongst the clusters regarding willingness to interact with a company online [6]. These studies set a precedent for the features we selected, which are discussed in Section III.

Improving on K-means

K-means is an efficient and flexible unsupervised learning algorithm. It can be adapted in a number of clever ways to suit various data sets including numerical, binary, and string features. Lingras and West use rough k-means, which estimates an upper and lower bound for each centroid rather than a single mean, to account for a bad or incomplete data set [7]. Ding and He present a strong argument for preprocessing with PCA [8]. Their analysis found a quality increase of more than 15% when reducing from 1000 dimensions to 5 prior to running k-means. The authors attribute this to the principal components being the features most indicative of cluster membership. Z. Huang points out that k-means is poorly suited to categorical data, and proposes the use of k-modes instead [9]. A drawback of this solution is that it forces centroids to take on the majority feature value without indicating whether the data points in that cluster are in strong agreement. Further, Pham et al. cautions

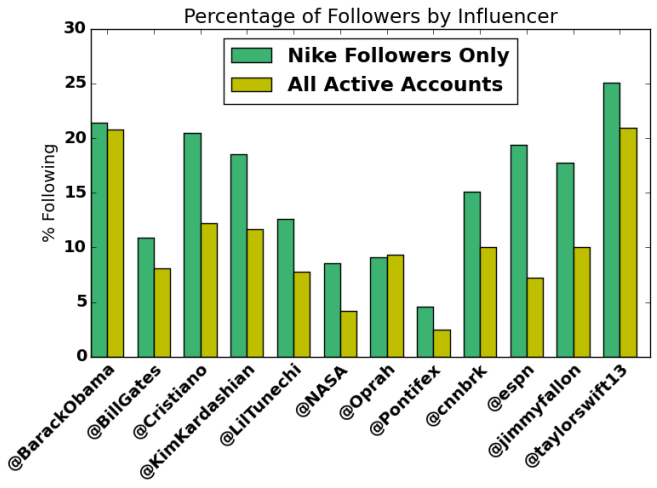

Fig. 1. Percentage of followers for a set of chosen influencers.

against using k-means as a black box and arbitrarily selecting the number of clusters [10].

III. Data Set and Features

User Data

Twitter's API rate limit constrains data gathering to a maximum limit of 720 data points per hour [11]. As such, we only consider a subset of 10,000 users from Nike's total 5.6 million followers. For each user, the data set includes a number of basic features including statuses posted, number of followers, number of accounts following, and language. In addition, we record whether each user is following one or more of a select list of popular Twitter accounts. We refer to these accounts as *influencers*. This set was hand-selected from a list of the 100 most-followed Twitter accounts and consists of: {Taylor Swift, ESPN, Bill Gates, Pope Francis, CNN, Barack Obama, Kim Kardashian, Cristiano Ronaldo, Jimmy Fallon, Oprah Winfrey, Lil Wayne, NASA} [12]. Figure 1 shows the percentage of users following each influencer for Nike's followers as well as the general Twitter population. Note that the distribution for Nike's followers is different than that for all Twitter users. For example, Nike's followers are more likely to follow ESPN than Barack Obama, while the opposite is true for the general Twitter population. Such differences are indicative of inherently distinct preferences for a chosen customer base. This further reaffirms the application of this analysis in targeted advertising.

Feature Similarity

The basic k-means algorithm requires features to have a numerical representation so that the chosen cluster centers' coordinates are well-defined. Specifically, it is important to preserve the meaning of the Euclidean distance between two samples as relating to similarity. In our case, all of the selected features are numerical except for the language of the Twitter user. The lexicographic proximity between the language acronyms for *en* and *es* is not indicative of actual similarity. To satisfy the similarity requirement, we convert language to a tuple of float values by mapping the language acronym to the latitude and longitude coordinates of the largest city in the country with the most people who speak this language. For example, the language acronym *th* is mapped to the geographic coordinates of Bangkok, Thailand (13.7563 N, 100.5018 E).

The k-means algorithm is isotropic with respect to all features. As a consequence, a feature with a larger range than another will indirectly receive more "weight" in the algorithm. One approach to alleviate this distortion is to map all features to be within the same range [13]. We choose to map the statuses posted, number of followers, number of accounts following, latitude, and longitude features to be within the range of the features output by PCA (described below).

Dimensionality Reduction

The original feature set includes two traits called *verified* and *utc offset*. The verified feature holds a boolean value to indicate if the user is famous or not. The utc offset field represents the user's timezone as an offset in seconds from GMT. Both of these features are shown to have low variance across the data set. The large majority of users have verified set to 0 and do not provide a utc offset value (possibly due to privacy concerns). Accordingly, these fields should be discarded from the final feature set.

We represent users following relationships towards influencers as a binary matrix with a 1 in the (i, j) position if user i follows influencer j. As previously mentioned, k-means does not work well on binary data. Therefore, as a pre-processing step, we perform PCA on the influencers matrix. We choose to reduce from 12 dimensions to 8. This corresponds to the lowest dimensionality that explains at least 85% of the variance, which is a common rule of thumb. Figure 2 illustrates how this minimum dimensionality is chosen.

IV. Methods

K-means

The k-means algorithm partitions the data by assigning each sample to a cluster for a predetermined number of clusters k. On initialization, k cluster centroids are randomly chosen. At each iteration, the algorithm assigns each sample to the cluster of the nearest centroid. It then recomputes the centroid to be the mean of the samples currently assigned to its cluster. The nearest centroid for a sample is defined to be the one with smallest Euclidean distance from that sample. K-means converges when the centroid values stabilize. The cluster centers c and labels are determined by minimizing

$$\arg\min_{\mathbf{c}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in c_i} \|\mathbf{x} - \boldsymbol{c}_i\|^2 \tag{1}$$

We employ k-means to perform the clustering because it produces acceptable experimental results and is considered to

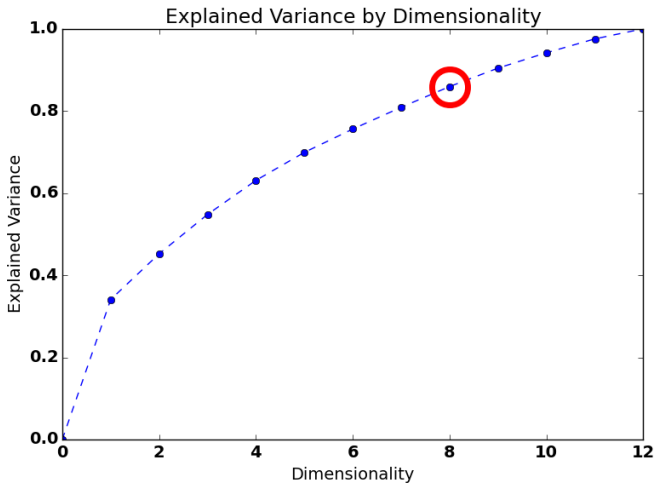

Fig. 2. Explained variance as a function of dimensionality.

be relatively computationally efficient. Our application requires clustering for a potentially massive social media data set. This suggests choosing k-means over slower alternatives such as hierarchical clustering [14].

The specific implementation of k-means we use in this paper is provided by Python's scikit-learn package [15].

Silhouette Coefficient

The remaining issue is to determine the number of clusters k. We begin by selecting the optimal number of clusters by maximizing the *silhouette coefficient* shown below. This metric is indicative of how well each object lies within its chosen cluster [16].

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & , a(i) < b(i) \\ 0 & , a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & , a(i) > b(i) \end{cases} \in [-1, 1]$$

The term $a(i)$ is the average dissimilarity of sample i to all other samples within the same cluster. It represents how well sample i "fits" in its cluster. And the term $b(i)$ is the smallest average dissimilarity of sample i to any other cluster, of which it is not a member. This represents the "next best fit" for sample i. Intuitively, the goal is to select clusters such that we maximize every sample's fit to its own cluster while minimizing the fit to the next best cluster. To achieve the maximum of $s(i) = 1$, we require $a(i) << b(i)$ for all samples.

One shortcoming of the silhouette coefficient is that it provides no preference for using less clusters. A maximum silhouette coefficient can trivially be obtained by selecting m clusters and assigning one to each sample. We attempt to overcome this deficiency by iteratively computing the silhouette coefficient for increasingly more clusters. We select the first cluster size that results in a silhouette coefficient of more than a chosen threshold $\Gamma = 0.7$.

Clustering Performance

Given the optimal value for k chosen above, it is necessary to measure the clustering performance on our data set. One feasible option is to simply use the output of the silhouette coefficient function corresponding to this value of k. However, one of our primary goals is to illustrate our clustering results in R^2, which is more conducive to visualization. To attain this, we compute a metric of clustering quality related to the intra-cluster variation and inversely proportional to the inter-cluster distance. This metric allows us to easily map to a lower dimensional space while still maintaining the same metric value. Intuitively, this will transform our clustering instance to a lower dimensional representation with the same ratio of intra-cluster variation to inter-cluster distance. We define the *clustering performance* as

$$q(x,k) = \frac{\sum_{i=1}^{k} \left(\frac{\sum_{j=1}^{m} 1\{x^{(j)} \in c_i\} ||x^{(j)} - c_i||_2}{\sum_{j=1}^{m} 1\{x^{(j)} \in c_i\}} \right)}{\sum_{i=1}^{k} \left(\frac{\sum_{j=1}^{k} 1\{i \neq j\} ||c_i - c_j||_2}{\sum_{j=1}^{k} 1\{i \neq j\}} \right)}, \quad (3)$$

where it is desirable to achieve a low score for the clustering performance q. The mean, rather than min or max, is chosen as the aggregate statistic for both intra-cluster variations and inter-cluster distances to be more robust to outliers.

V. Results

Selecting the Number of Clusters

In selecting the optimal number of clusters, we ignore the meaningless solution of $k = 1$. Further, we upper-bound the number of clusters at $k = 15$ since that seems reasonable given our particular application. As described above, we iteratively run k-means for the remaining candidate values of k and compute the corresponding silhouette coefficient. Figure 3 illustrates the silhouette coefficient for each value of k. The optimal value of k was experimentally determined to be $k = 5$.

Measuring Clustering Performance

Given the number of clusters k selected above, we next compute the clustering performance q for our data set. Figure 4 shows the relative clustering performance scores for randomized data, our data set, and perfectly clustered data. Low values of q correspond to better clustering performance. This result indicates that our clustering performance is in between that of ideal data and that of randomized data.

Visualizing Clusters

As described above, one key ambition is to visualize some representation of our clustering output in R^2. Figure 5 indicates one such visualization. The depicted clusters have the same ratio of average intra-cluster variation to average inter-cluster distance as our clustering output. This suggests that our data set can be cleanly clustered in our dimensionality space.

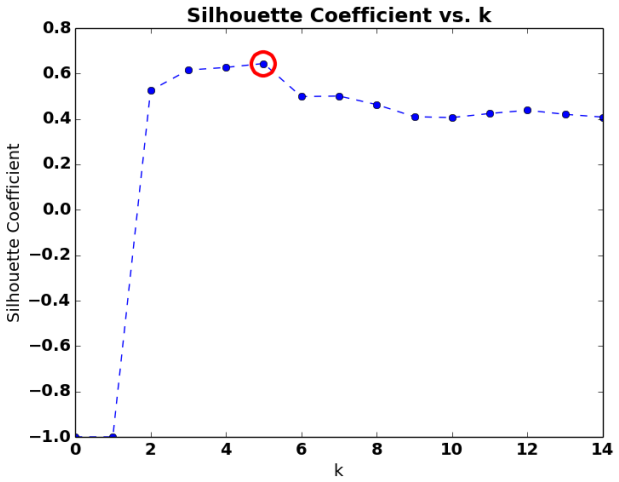

Fig. 3. Silhouette coefficient as a function of number of clusters.

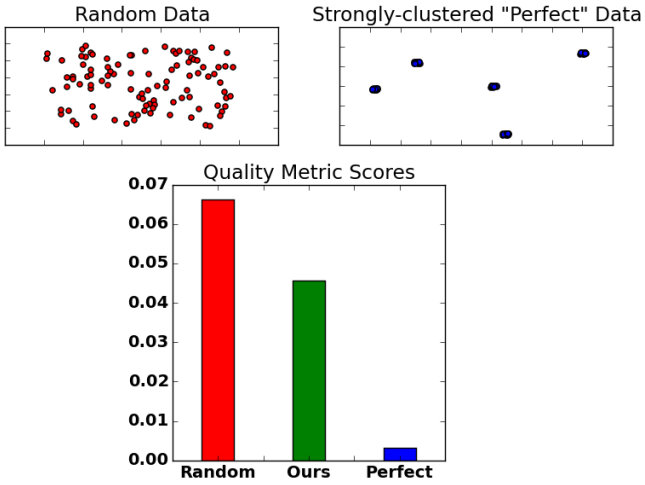

Fig. 4. Cluster performance q for randomized data, our data set, and perfectly clustered data.

Manual Labeling

We attempt to determine the mode user of each cluster by examining a randomly-selected subset of samples from each of the $k = 5$ clusters.

Cluster 1: These users speak English and follow many of the chosen influencers. This group represents Americans that are actively engaged in pop-culture.

Cluster 2: These users live in Europe, primarily in wealthier western countries such as France and Germany. They follow fewer of the influencers than Cluster 1. However, the influencers these users do follow are international figures such as the Pope or Barack Obama.

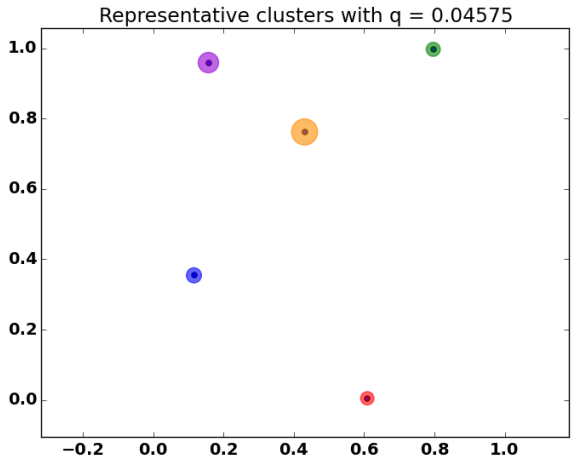

Fig. 5. Representative clusters in R^2 that share the same ratio of intra-cluster variations to inter-cluster distances as our clustering output.

Cluster 3: These users are based in Spanish or Portuguese speaking countries. They are not following the majority of the influencers. However, there is one noteworthy exception: nearly all follow soccer star Cristiano Ronaldo.

Cluster 4: These users are based in Asia. They appear slightly more active on Twitter than users in the other clusters. A number of accounts in this cluster have usernames that are a long series of letters and numbers, for example @1B8rSK3QaO92KKM and @BIKAOQta93k2nDb. This warrants further investigation and may correspond to bot Twitter accounts. If we suspect that a significant number of these users are indeed bots, we can try running k-means with $k = 2$ on just this cluster to separate the legitimate users from the rest.

Cluster 5: These users speak English, and follow few to none of the influencers we selected. However, further examination of their profiles shows that they are following other companies and public figures. These include Louis Vuitton (which indicates an interest in luxury and brand name goods), Kendall Jenner (whose fan base is younger than all the influencers on the current list), and sports brands such as Puma and Adidas (which implies an interest in all athletic wear and not Nike's products in particular). This suggests that our influencers list did not fully capture all interests common among Nike's followers. Expanding the influencers list will likely lead to better clustering accuracy.

An experiment was performed to validate the meaningfulness of these selected clusters. A human subject was presented with the chosen clusters and their descriptions. Then the subject was given a sample input feature and asked to assign it to the most fitting cluster. This process was performed repeatedly and the number of classifications that contradicted our clustering was tracked. Our empirical results show that the

human and our algorithm were in agreement approximately 80% of the time.

VI. Conclusion

Summary

This paper has shown a method for extracting and processing publicly-available social media data for the purposes of customer clustering. Section III showed how to use dimensionality reduction techniques to sanitize the extracted features. Section IV developed metrics for selecting the optimal number of clusters k and evaluating the overall clustering performance q. Lastly, in Section V we illustrated a representative clustering output for Nike's Twitter followers in R^2.

Discussion

Figure 4 indicates that the quality of clustering achievable using our techniques is midway in between that of ideal and randomized data. Further, the result shown in Figure 5 shows that the clusters created are remarkably pronounced and well-defined. The manual labeling experiment demonstrates that the selected clusters have recognizable meaning to a human observer. Overall the outcome of these experiments further reinforces the credibility of using social media data for market analysis.

Future Work

Future research on this topic should focus on applying these algorithms to a more reliable data set. One assumption we make in this paper is that Twitter users are accurately reporting their preferences through their following actions. This could be skewed because following preferences are publicly-visible. More factual data sources might include credit card transactions, newsletter subscriptions, and Amazon shopping cart history. Researchers with access to these types of proprietary information would likely discover clusters with a better clustering performance than presented here.

References

[1] L. Colombus. Roundup Of Analytics, Big Data Business Intelligence Forecasts And Market Estimates. 2015. http://www.forbes.com/sites/louiscolumbus/2015/05/25/roundup-of-analytics-big-data-business-intelligence-forecasts-and-market-estimates-2015/

[2] G. Timpany. 6 Ways Market Researchers Can Use Social Media Analytics. *ResearchAccess* 2014. http://researchaccess.com/2014/12/social-media-analytics/

[3] J. Roesslein. Tweepy. http://tweepy.readthedocs.org/en/v3.2.0/, retrieved 2015.

[4] S. Bergsma, M. Dredze, et al. Broadly Improving User Classification via Communication-Based Name and Location Clustering on Twitter. 2013. http://www.clsp.jhu.edu/ sbergsma/TwitterClusters/

[5] I. Guy, M. Jacovi, et al. Same Places, Same Things, Same People? *IBM Haifa Research Lab*. 2010. http://dl.acm.org/citation.cfm?id=1718928

[6] B. Jansen, K. Sobel, et al. Classifying ecommerce information sharing behaviour by youths on social networking sites. *Journal of Information Science*. 2011.

[7] P. Lingras, C. West. Interval Set Clustering of Web Users with Rough K-Means. *Journal of Intelligent Information Systems*. 2004.

[8] C. Ding, X. He. K-means clustering via principal component analysis. *ACM*, page 29. 2004.

[9] Z. Huang. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. 1998.

[10] D. Pham, S. Dimov, et al. Selection of K in K-means clustering. 2004.

[11] Twitter. API Rate Limits, retrieved 2015. https://dev.twitter.com/rest/public/rate-limiting

[12] TwitterCounter, retrieved 2015. http://twittercounter.com/

[13] A. Jain. Data Clustering: A Review. *ACM Computing Surveys*. Volume 31 Issue 3, Pages 264-323. 1999.

[14] M. Kaur, U. Kaur. Comparison Between K-Mean and Hierarchical Algorithm Using Query Redirection. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2013.

[15] scikit-learn.org, retrieved 2015. http://scikit-learn.org/stable/

[16] P. Rousseeuw. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Computational and Applied Mathematics*. 20: 5365. 1987.