CS999: Web Data and Text Mining
Annotated Bibliography

Justin Kamerman 3335272

February 9, 2011

Bibliography

[1] R. Cooley, B. Mobasher, and J. Srivastava, "Data preparation for mining world wide web browsing patterns," *KNOWLEDGE AND INFORMATION SYSTEMS*, vol. 1, pp. 5–32, 1999.

[2] A. Ghorbani and I.-V. Onut, *Y-Means: An Autonomous Clustering Algorithm*, ser. Hybrid Artificial Intelligence Systems. Springer Berlin / Heidelberg, 2010, vol. 6076, pp. 1–13.

[3] C. H. and Z. D., "Ai and opinion mining," *IEEE Intell.Syst.IEEE Intelligent Systems*, vol. 25, no. 3, pp. 74–76, 2010.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput.Surv.*, vol. 31, no. 3, pp. 264–323, September 1999. [Online]. Available: http://doi.acm.org.proxy.hil.unb.ca/10.1145/331499.331504

Paper is an overview of data clustering concepts and techniques. Clustering is an exploratory undertaking (unsupervised), as opposed to classification (supervised). During clustering, a collection of patterns are organised based on a notion of their similarity to one another. Patterns are typically represented as feature vectors and their organisation occurs within this feature space. Pattern selection involves feature selection as well as feature extraction, transforming input features to produce new salient features. All clustering algorithms will produce clusters regardless of the underlying data so how do we evaluate a cluster algorithm ? Cluster validation studies can be *external*, comparing the recovered structure to an *a priori* structure; *internal*, which examines whether the structure is intrinsically appropriate for the data; or *relative*, which compares two structures and measures their relative merit.

A measure of the similarity between two patterns is essential to most clustering procedures. The most common measure is the Euclidean distance. It works well when a data set has compact, isolated clusters but large scale features tend to dominate unless weighted or normalized. Salient cluster algorithm

properties include: agglomerative vs divisive; monothetic vs polythetic; hard vs fuzzy; deterministic vs stochastic; incremental vs non-incremental; and hierarchical vs partitioning.

Hierarchical algorithms produce a *dendrogram* representing nested groupings of patterns and the similarity thresholds at which they change. Most hierarchical cluster algorithms are variants of the single-link, *single-link* (distance between clusters is the minimum between any two patterns drawn from different clusters); *complete-link* (distance between clusters is the maximum between any two patterns from different clusters); and *minimum-variance* algorithms. Partitioning algorithms are less demanding computationally compared to hierarchical algorithms. A problem of these algorithms is the choice of the number of desired output clusters. The most common and intuitive criterion function used in partitional clustering is the *squared-error* criterion of which *K-Means* is the simplest and most commonly used. *K-Means* is one of the most efficient in terms of execution time and one of the few methods appropriate for use on large data sets. *K-Means* requires one to specify the number of clusters to create which is difficult to do optimally. Variants of *K-Means* have been proposed which dynamically merge and/or spilt clusters based on a variance threshold.

Clusters are typically represented by their centroid, a simple scheme if clusters are compact and iso-tropic. If clusters are elongated or non-isotropic, then this representation weak, better replaced by a collection of points.

Search based cluster techniques can be either deterministic or stochastic. Deterministic techniques guarantee an optimal partition by performing exhaustive enumeration. Stochastic search techniques generate near optimal partitions reasonably quickly and guarantee asymptotic convergence to optimal partition.

Clustering is subjective by nature. Subjectivity is usually incorporated into some phase of clustering, whether it be in selection of a pattern representation, choosing a similarity measure, or cluster representation. The incorporation of domain knowledge consists of ad-hoc approaches with little in common.

Clustering of large data sets is computationally demanding and many clustering algorithms do not scale adequately. The emerging discipline of data mining has spurred developments and optimizations in this area. Clustering is used in the data mining process for segmentation of databases into homogeneous groups,

predictive modelling, and visualization. If the data set is too large to fit in main memory, techniques like *divide-and-conquer*, incremental clustering, and parallel algorithm implementations have been used.

The paper review several application domains in which clustering has been successfully employed: image segmentation, object and character recognition, information retrieval, and data mining.

[5] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 6/1 2010.

Although *K-Means* was devised in 1955, it is still widely used because of its simplicity, efficiency, and empirical success. The paper looks at the difficulties of developing better algorithms. Clustering algorithms can be broadly divided into *hierarchical* and *partitional*. Hierarchical algorithms recursively find nested clusters in either *agglomerative* (bottom up) or *divisive* (top down) mode, taking an $n * n$ similarity matrix as input. Partitional algorithms take as input an $n * d$ pattern matrix or a similarity matrix.

The *K-Means* algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. The goal is to minimize the squared error over all clusters however this problem is NP-hard so, out of necessity, *K-Means* is a greedy algorithm which converges to a local minima. Research does show however that if the clusters are well separated, the algorithm will converge with high probability to the global optimum. The main steps of *K-Means* are

1. Select an initial partition and repeat steps 2 and 3 until cluster membership stabilizes.
2. Generate a new partition by assigning each pattern to its nearest cluster centre.
3. Compute new cluster centres.

K-Means requires three user-specified parameters: number of clusters, cluster initialization, and distance metric. Selection of number of clusters is difficult and usually based on heuristics and/or repeated execution with different number of clusters, adjudicated by a domain expert. *K-Means* typically uses the Euclidean distance metric and as a result, finds hyperspherical shaped clusters.

Clustering algorithms have been developed that model pattern density by a probabilistic mixture model viz. EM algorithm

and several Bayesian approaches. These methods are attractive because of their ability to deal with arbitrary shaped clusters but have difficulty dealing with high dimensional data the feature space is characteristically sparse, making it difficult to distinguish high density regions from low. Graph theoretic clustering is another class of clustering algorithms. These algorithms represents data points as nodes in a graph with connecting edges weighted by their pair-wise similarity. The central idea is to partition the nodes into two groups such that the weights of the edges between the two groups is minimized. All the variables involved in a clustering project make it inherently difficult. One of the most important decisions is that of data representation. A good data representation will result in compact, well separated clusters however there is no universally good representation and the process must be guided by domain knowledge. Another variable is the number of clusters. Automatic determination of this variable has been one of the most difficult problems in clustering. Alternatively, the optimal number of clusters must be determined through trial and error. Since clustering algorithms tend to find clusters irrespective of whether they exist, it is important to objectively evaluate whether the data has a natural tendency to cluster. *Cluster validation* is the formal evaluation of clustering results in a quantitative and objective manner. Cluster validity measures can be *internal*, *external*, or *relative*. Internal measures asses the fit between the structure imposed by the algorithm and the data itself. Relative measures compare the structure imposed by different algorithms on the same data. External measures compare cluster structure to some a priori information, namely "true" class labels.

Stability of a clustering solution is a measure of how much variation occurs in the structure imposed over different subsamples drawn from the input data. Different measures of variation can be used to obtain different stability measures. Since many algorithms are asymptotically stable, it may be important to consider the rate at which stability is reached. Some recent clustering trends include:

- **Clustering Ensembles**: combine the resulting partitions resulting from application of differing clustering methods on the same data.
- **Semi-Supervised Clustering**: a subset of the data is labelled and these are used to impose pairwise constraints (*must-link* and *cannot-link*) on the cluster algorithm.
- **Large-Scale Clustering**: algorithms developed to handle large data sets can be classified as: efficient nearest neighbour

(NN), data summarization, distributed computing, incremental clustering, or sampling-based methods.

[6] B. Liu and S. O. service), "Web data mining exploring hyperlinks, contents, and usage data," " 2007.

[7] T. M. Mitchell, *Machine Learning.* New York: McGraw-Hill, 1997.

ID: 36417892

[8] M. Rosell, "Introduction to Information Retrieval and Text Clustering," " 2006.

This is a collection of chapters adopted from the authors licentiate theses *Clustering in Swedish*. The first chapter introduces the field of Information Retrieval (IR), as a large and growing field within Natural Language Processing (NLP). IR is the theoretical foundation of text search engines. Texts are represented as vectors, each dimension corresponding to a distinct word in the set of words appearing in all texts. The vector fields are weights which model how important the corresponding word is deemed to be in the context of the text. There are many weighting schemes but in the most common the weights are the product the *term-frequency* (tf) and *inverse document frequency* (idf). The term frequency is a function of the number of occurrences of a particular word in a document divided by the number of words in the entire document. The inverse document frequency models the distinguishing power of the word in the text set; the fewer documents that contain the word, the more information about the text int he text set it gives.

In a text query, a search is conducted for texts similar to the search vector which is represented in the same way as the texts. The most common measure of similarity is the *cosine measure*, the cosine of the angle between the query and texts. The texts are returned are ranked by similarity. It is difficult to evaluate search results. In a controlled text set, query results can be compared against results of human opinion. By comparing these perspectives, we may define performance measures for the search engine. *Precision* and *recall* are common measures. To further characterize search engine performance over a range of operating conditions, the precision at different levels of recall can be plotted in a graph.

Modifications can be made to the vector space model described to improve search performance:

- **Stoplist and Word Classes**: stoplist words are excluded from the model, usually very common words whose occurrence do not separate one text significantly from another.
- **Phases**: treat phrases as separate dimensions for phrase based searches.
- **Lemmatizing and Stemming**:
- **Related Words**: the vector model does not account for the fact that words may be related (synonyms, homonyms etc). Many attempts have been made to attempt to address this phenomenon viz. word sense disambiguation, query expansion.
- **Statistically Related Words**: statistical examination of the word-by-document matrix gives information regarding words that appear together often. This information can be used by search engines to improve performance. *Latent Semantic Analysis* (LSA) is such a technique but is computationally heavy. *Random Indexing* (RI) is a much faster, less memory intensive alternative but does not use the entire word-by-document matrix.
- **Meta-data**: meta-data found in web pages provides additional information that can be used when indexing.

Text clustering can be used to discover structures within a text set that were not previously known. This is as opposed to text categorization where texts are assigned to predefined categories. IR and text clustering are related in that they both employ the same pattern representation and search function. Researchers believe that credible text clustering could make search times shorter by retrieving clusters of texts instead of individual documents. Similar (clustered) documents are probably relevant to the same queries but that does not mean that pre-clustering of the entire text set can take all future queries into account (I think this means that clustering is coarse grained relative to search queries). The authors argue for text clustering after ordinary search engine retrieval and have shown through experimentation that this can improve search result quality.//// It is hard to objectively evaluate clustering results since the value thereof is subjective. It is common to distinguish between intrinsic and external measures. Intrinsic measure use no external knowledge other than what was available to the cluster algorithm. External measures use external knowledge.

[9] G. P. Zhang, "Neural networks for classification: a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 30, no. 4, pp. 451–462, 2000.

This paper is a review of the use of Artificial Neural Networks (ANN) for classification tasks. It compares ANNs to statistically based classification procedures and explains advantages and disadvantages of ANN relative to these more traditional approaches. The paper shows how ANNs are able to estimate posterior classification probabilities by virtue of the fact that ANNs are typically trained by attempting to minimize mean squared errors. This provides a direct link between ANN classifications and statistical methods, particularly Bayesian.

Direct comparison of ANN and statistical classifiers may not be possible because ANNs are non-linear and model-free, while statistical methods are linear and model-based. However, by appropriately encoding ANN outputs, we can use ANNs to directly model some high order discriminant functions. Analysis along these lines has shown that the hidden layers of an MLP project the data onto different clusters in a way that these clusters can be further aggregated into different classes. However, the added flexibility of ANNs due to hidden layers does not automatically guarantee their superiority over logistical regression due to possible overfitting and other inherent problems.

Due to the variables associated with constructing ANN classifiers and the local minima problem associated with training ANNs, there is an inherent error between true posterior probabilities and the least square estimates provided by ANN. This prediction error is composed of two components, the *approximation error* and the *estimation error*. The *approximation error* reflects a inherent irreducible consequence of the randomness of the training data. The *estimation error* is a reflection of the effectiveness of the ANN to approximate the target function.

The paper describes how a bias-plus-variance decomposition of the ANN prediction error provides useful information on how the estimate differs from the target function. The model bias quantifies how the average estimates over all possible data sets of the same size differ from the target function. Bias is an indication of the limitations of the model itself. Model variance is an indication of the sensitivity of the estimation function to the training data set. Bias and variance are generally conflicting goals. ANNs are flexible and tend to have low bias but high variance.

Ensemble methods are described where classifiers are combined by averaging or voting prediction results from multiple ANNs. Improvements in prediction results are attributed to

reduction of variance. The technique seems to work best when the voting models disagree with one another strongly i.e. are biased. Averaging seems to offset this bias and reduce sensitivity to the data. Methods of constructing biased models include statistical resampling techniques and using different feature variables.

Feature selection methods for ANNs are mostly heuristic in nature and and lack statistical justification.

Taking misclassification costs into account seems to improve the performance of ANNs in terms of classification and feature selection. Various techniques are described for incorporating misclassification cost information and prior knowledge of relative class importance, however, little research has been done in the this area.