Automatic Cluster Stopping with Criterion Functions and the Gap Statistic

Ted Pedersen and **Anagha Kulkarni**
Department of Computer Science
University of Minnesota, Duluth
Duluth, MN 55812 USA
{tpederse,kulka020}@d.umn.edu
http://senseclusters.sourceforge.net

Abstract

SenseClusters is a freely available system that clusters similar contexts. It can be applied to a wide range of problems, although here we focus on word sense and name discrimination. It supports several different measures for automatically determining the number of clusters in which a collection of contexts should be grouped. These can be used to discover the number of senses in which a word is used in a large corpus of text, or the number of entities that share the same name. There are three measures based on clustering criterion functions, and another on the Gap Statistic.

1 Introduction

Word sense and name discrimination are problems in unsupervised learning that seek to cluster the occurrences of a word (or name) found in multiple contexts based on their underlying meaning (or identity). The assumption is made that each discovered cluster will represent a different sense of a word, or the underlying identity of a person or organization that has an ambiguous name.

Existing approaches to this problem usually require that the number of clusters to be discovered (*k*) be specified ahead of time. However, in most realistic settings, the value of *k* is unknown to the user. Here we describe various *cluster stopping measures* that are now implemented in SenseClusters (Purandare and Pedersen, 2004) that will group N contexts into *k* clusters, where the value of *k* will be automatically determined.

Cluster stopping can be viewed as a problem in model selection, since a number of different models (i.e., clustering solutions) are created using different values of *k*, and the one that best fits the observed data is selected based on a criterion function. This is reminiscent of earlier work on sequential model selection for creating models of word sense disambiguation (e.g., (O'Hara et al., 2000)), where it was found that forward sequential search strategies were most effective. These methods start with simpler models and then add to them in a stepwise fashion until no further improvement in model fit is observed. This is in fact very similar to what we have done here, where we start with solutions based on one cluster, and steadily increase the number of clusters until we find the best fitting solution.

SenseClusters supports four cluster stopping measures, each of which is based on interpreting a clustering criterion function in some way. The first three measures (PK1, PK2, PK3) look at the successive values of the criterion functions as *k* increases, and try to identify the point at which the criterion function stops improving significantly. We have also created an adaptation of the Gap Statistic (Tibshirani et al., 2001), which compares the criterion function from the clustering of the observed data with the clustering of a null reference distribution and selects the value of k for which the difference between them is greatest.

In order to evaluate our results, we sometimes conduct experiments with words that have been manually sense tagged. We also create *name con-*

Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume, pages 276–279,
New York City, June 2006.

flations where some number of names of persons, places, or organizations are replaced with a single name to create pseudo or false ambiguities. For example, in this paper we refer to an example where we have replaced all mentions of *Sonia Gandhi* and *Leonid Kuchma* with a single ambiguous name.

Clustering methods are typically either partitional or agglomerative. The main difference is that agglomerative methods start with 1 or N clusters and then iteratively arrive at a pre–specified number (*k*) of clusters, while partitional methods start by randomly dividing the contexts into *k* clusters and then iteratively rearranging the members of the *k* clusters until the selected criterion function is maximized. In this work we have used K-means clustering, which is a partitional method, and the $H2$ criterion function, which is the ratio of within–cluster similarity ($I2$) to between–cluster similarity ($E1$).

2 Methodology

In word sense or name discrimination, the number of contexts (N) to cluster is usually very large, and considering all possible values of *k* from $1...N$ would be inefficient. As the value of *k* increases, the criterion function will reach a plateau, indicating that dividing the contexts into more and more clusters does not improve the quality of the solution. Thus, we identify an upper bound to *k* that we refer to as *deltaK* by finding the point at which the criterion function only changes to a small degree as *k* increases.

According to the $H2$ criterion function, the higher its ratio of within–cluster similarity to between–cluster similarity, the better the clustering. A large value indicates that the clusters have high internal similarity, and are clearly separated from each other. Intuitively then, one solution to selecting *k* might be to examine the trend of $H2$ scores, and look for the smallest *k* that results in a nearly maximum $H2$ value.

However, a graph of $H2$ values for a clustering of the 2 sense name conflation *Sonia Gandhi* and *Leonid Kuchma* as shown in Figure 1 (top) reveals the difficulties of such an approach. There is a gradual curve in this graph and there is no obvious *knee point* (i.e., sharp increase) that indicates the appropriate value of *k*.

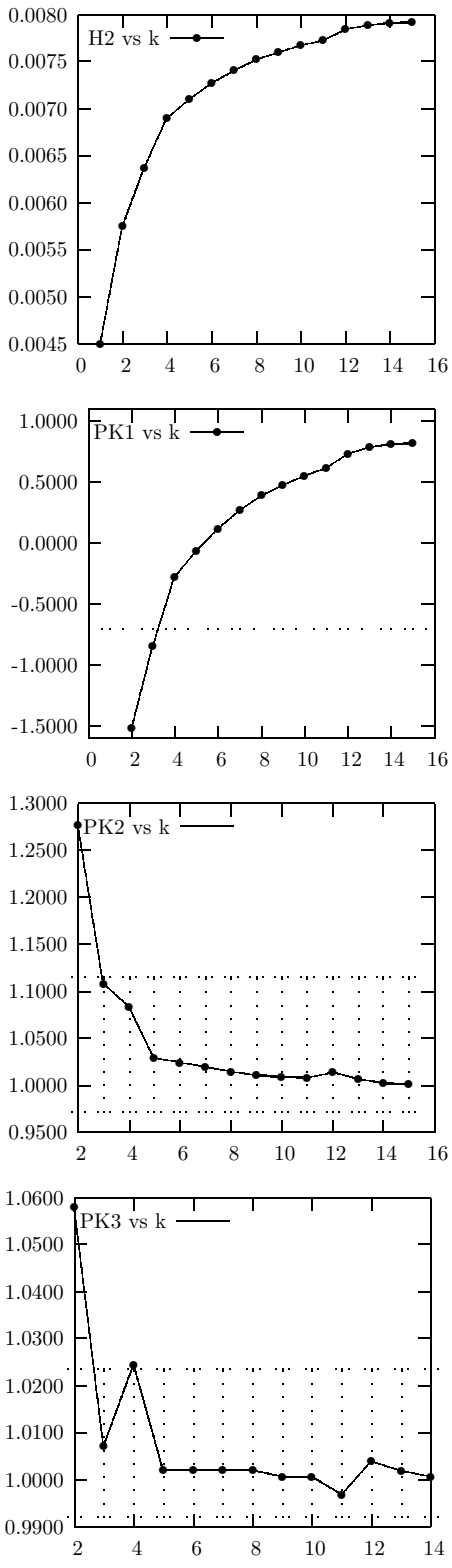

Figure 1: H2 (top) and PK1, PK2, and PK3 for the name conflate pair *Sonia Gandhi* and *Leonid Kuchma*. The predicted number of senses is 2 for all the measures.

2.1 PK1

The $PK1$ measure is based on (Mojena, 1977), which finds clustering solutions for all values of k from $1..N$, and then determines the mean and standard deviation of the criterion function. Then, a score is computed for each value of *k* by subtracting the mean from the criterion function, and dividing by the standard deviation. We adapt this technique by using the $H2$ criterion function, and limit *k* from $1...deltaK$:

$$PK1(k) = \frac{H2(k) - mean(H2[1...deltaK])}{std(H2[1...deltaK])} \quad (1)$$

To select a value of *k*, a threshold must be set. Then, as soon as $PK1(k)$ exceeds this threshold, *k-1* is selected as the appropriate number of clusters. Mojena suggests values of 2.75 to 3.50, but also states they would need to be adjusted for different data sets. We have arrived at an empirically determined value of -0.70, which coincides with the point in the standard normal distribution where 75% of the probability mass is associated with values greater than this.

We observe that the distribution of $PK1$ scores tends to change with different data sets, making it hard to apply a single threshold. The graph of the $PK1$ scores shown in Figure 1 illustrates the difficulty : the slope of these scores is nearly linear, and as such any threshold is a somewhat arbitrary cutoff.

2.2 PK2

$PK2$ is similar to (Hartigan, 1975), in that both take the ratio of a criterion function at *k* and *k-1*, in order to assess the relative improvement when increasing the number of clusters.

$$PK2(k) = \frac{H2(k)}{H2(k-1)} \quad (2)$$

When this ratio approaches 1, the clustering has reached a plateau, and increasing *k* will have no benefit. If $PK2$ is greater than 1, then we should increase *k*. We compute the standard deviation of $PK2$ and use that to establish a boundary as to what it means to be "close enough" to 1 to consider that we have reached a plateau. Thus, $PK2$ will select *k* where $PK2(k)$ is the closest to (but not less than) 1 + standard deviation(PK2[1...*deltaK*]).

The graph of $PK2$ in Figure 1 shows an *elbow* that is near the actual number of senses. The critical region defined by the standard deviation is shaded, and note that $PK2$ selected the value of *k* that was outside of (but closest to) that region. This is interpreted as being the last value of *k* that resulted in a significant improvement in clustering quality. Note that here $PK2$ predicts 2 senses, which corresponds to the number of underlying entities.

2.3 PK3

$PK3$ utilizes three *k* values, in an attempt to find a point at which the criterion function increases and then suddenly decreases. Thus, for a given value of *k* we compare its criterion function to the preceding and following value of *k*:

$$PK3(k) = \frac{2 \times H2(k)}{H2(k-1) + H2(k+1)} \quad (3)$$

The form of this measure is identical to that of the Dice Coefficient, although in set theoretic or probabilistic applications Dice tends to be used to compare two variables or sets with each other.

$PK3$ is close to 1 if the $H2$ values form a line, meaning that they are either ascending, or they are on the plateau. However, our use of *deltaK* eliminates the plateau, so in our case values of 1 show that *k* is resulting in consistent improvements to clustering quality, and that we should continue. When $PK3$ rises significantly above 1, we know that *k+1* is not climbing as quickly, and we have reached a point where additional clustering may not be helpful. To select k we select the largest value of $PK3(k)$ that is closest to (but still greater than) the critical region defined by the standard deviation of $PK3$.

$PK3$ is similar in spirit to (Salvador and Chan, 2004), which introduces the L measure. This tries to find the point of maximum curvature in the criterion function graph, by fitting a pair of lines to the curve (where the intersection of these lines represents the selected *k*).

2.4 The Gap Statistic

SenseClusters includes an adaptation of the Gap Statistic (Tibshirani et al., 2001). It is distinct from the measures PK1, PK2, and PK3 since it does not attempt to directly find a knee point in the graph of a criterion function. Rather, it creates a sample of reference data that represents the observed data as if it had no meaningful clusters in it and was simply made up of noise. The criterion function of the reference data is then compared to that of the observed data, in order to identify the value of *k* in the observed data that is least like noise, and therefore represents the best clustering of the data.

To do this, it generates a null reference distribution by sampling from a distribution where the marginal totals are fixed to the observed marginal values. Then some number of replicates of the reference distribution are created by sampling from it with replacement, and each of these replicates is clustered just like the observed data (for successive values of *k* using a given criterion function).

The criterion function scores for the observed and reference data are compared, and the point at which the distance between them is greatest is taken to provide the appropriate value of *k*. An example of this is seen in Figure 2. The reference distribution represents the noise in the observed data, so the value of *k* where the distance between the reference and observed data is greatest represents the most effective clustering of the data.

Our adaption of the Gap Statistic allows us to use any clustering criterion function to make the comparison of the observed and reference data, whereas the original formulation is based on using the within–cluster dispersion.

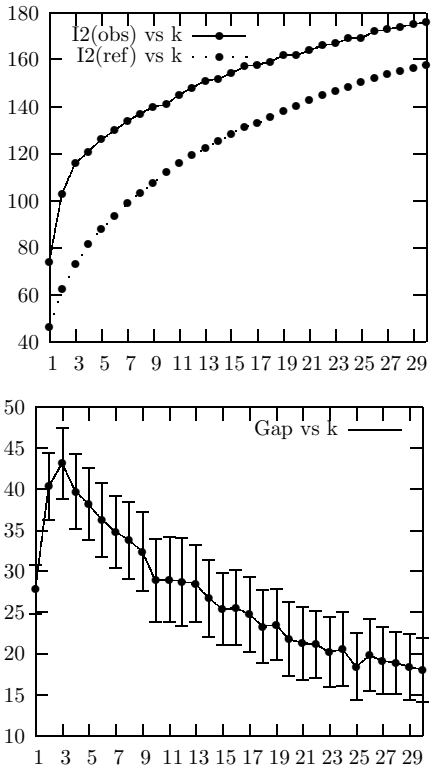

Figure 2: $I2$ for observed and reference data (top) and the Gap between them (bottom) for the name conflate pair *Sonia Gandhi* and *Leonid Kuchma*. The predicted number of senses is 3.

3 Acknowledgments

This research is supported by a National Science Foundation Faculty Early CAREER Development Award (#0092784).

References

J. Hartigan. 1975. *Clustering Algorithms*. Wiley, New York.

R. Mojena. 1977. Hierarchical grouping methods and stopping rules: An evaluation. *The Computer Journal*, 20(4):359–363.

T. O'Hara, J. Wiebe, and R. Bruce. 2000. Selecting decomposable models for word-sense disambiguation: The grling-sdm system. *Computers and the Humanities*, 34(1–2):159–164.

A. Purandare and T. Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 41–48, Boston, MA.

S. Salvador and P. Chan. 2004. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE International Conference on Tools with AI*, pages 576–584.

R. Tibshirani, G. Walther, and T. Hastie. 2001. Estimating the number of clusters in a dataset via the Gap statistic. *Journal of the Royal Statistics Society (Series B)*, pages 411–423.