An Objective Evaluation Criterion for Clustering

Arindam Banerjee
Dept of ECE
University of Texas at Austin
Austin, TX, USA
abanerje@ece.utexas.edu

John Langford
Toyota Technological Institute
Chicago, IL, USA
jl@tti-c.org

ABSTRACT

We propose and test an objective criterion for evaluation of clustering performance: How well does a clustering algorithm run on unlabeled data aid a classification algorithm? The accuracy is quantified using the PAC-MDL bound [3] in a semisupervised setting. Clustering algorithms which naturally separate the data according to (hidden) labels with a small number of clusters perform well. A simple extension of the argument leads to an objective model selection method. Experimental results on text analysis datasets demonstrate that this approach empirically results in very competitive bounds on test set performance on natural datasets.

Categories and Subject Descriptors: I.5.3 [Pattern Recognition]: Clustering

Keywords: Clustering, Evaluation, PAC bounds, MDL

1. MOTIVATION

Clustering is perhaps the most widely used exploratory data analysis tool using in data mining. There are many clustering algorithms that have been proposed in the literature based on various requirements. Each one optimizes some unsupervised internal quality measure such as minimizing the intra-cluster distances, maximizing the inter-cluster distances, maximizing the log-likelihood of a parametric mixture model, minimizing the cut-value of a pairwise similarity graph, etc. The differences between these internal quality measures make it practically impossible to objectively compare clustering algorithms. Due to the lack of an objective measure, choosing the "right" algorithm for a particular task is confusing.

We propose a method which eliminates some of these confusions in some settings. Clustering is often used as an intermediate exploratory data analysis step for a prediction problem. Hence, the answer to *what is a good clustering algorithm for my problem?* often depends on how much prediction power the clustering step provides, or, more directly, *what is the predictive accuracy of a clustering algorithm?* In other words, since clustering is often used as a method for gaining insight into a dataset, our method here evaluates the degree to which clustering aids the understanding of a dataset *for the purpose of classification.* On natural semi-supervised learning datasets, if the clustering "agrees well" with a set of hidden labels using a small number of clusters, we say that the clustering algorithm is good for prediction. The precise definition of "agrees well" is mediated by the PAC-MDL bound [3] on the accuracy of a test set. Our proposed criterion has several natural properties:

1. It applies to every clustering algorithm.
2. It is inherently normalized to the interval $[0, 1]$.
3. All possible values are exercised on natural datasets.
4. The metric can flexibly incorporate prior information by proper design of a *description language.*
5. It can be used for model selection.
6. It is directly related to a concrete goal (good prediction).

It is important to note that this is not the only possible objective criterion for evaluation of clustering algorithms. Clustering algorithms may be used for other purposes, and when so used, other measures may be more appropriate. Our results here are only directly applicable when the goal in clustering is related to prediction.

The rest of the paper is organized as follows. First, we explain the PAC-MDL bound [3] in section 2. In section 3, we discuss how the PAC-MDL bound can be applied to a clustering setting for performance evaluation and model selection. We present experimental results on benchmark text datasets in section 4 to demonstrate the proposed approach. Related work is presented in section 5. We end with a discussion on the proposed criterion in section 6.

2. THE PAC-MDL BOUND

The PAC-MDL bound is the core mechanism used to trade off between a sufficiently rich representation to capture the data and over-fitting on the data. Clustering algorithms with a small bound must have a small number of clusters which agree well with a set of (hidden) labels.

Consider the following learning setting: Let D be any distribution over (X, Y) where X denotes the input and Y denotes the label. We assume Y can take one of $\ell > 1$ possible values, i.e., $Y \in \{1, \cdots, \ell\}$. Consider a train set $S = (X^m, Y^m)$ and a test set $S' = (X^n, Y^n)$, where (X^i, Y^i) denotes the set of i independently drawn samples from the (unknown) joint distribution D over (X, Y). The PAC-MDL bound applies to a transductive learning algorithm, $T : (X \times Y)^m \times X^n \mapsto \sigma$, where $\sigma : X^{m+n} \mapsto \hat{Y}^{m+n}$ is a transductive classifier which produces a simultaneous labeling $\hat{Y}^{m+n}$ for all of the data points X^{m+n}. Given the description complexity of the transductive classifier measured by its bit description length $|\sigma|$, and the prediction error count on the train set, $\hat{\sigma}_S = |\{Y^m \neq \hat{Y}^m\}|$, the bound limits the error count on the test set $\hat{\sigma}_{S'} = |\{Y^n \neq \hat{Y}^n\}|$.

The precise bound is defined in terms of a cumulative hypergeometric distribution. To understand this distribution, imagine a bucket with m red balls and n blue balls, from which $(a+b)$ balls are drawn without replacement. Now, define $\text{Bucket}(m,n,a,b)$ to be the probability that at least b blue balls are drawn. That is,

$$\text{Bucket}(m,n,a,b) \;=\; \sum_{t=b}^{a+b} \frac{\binom{n}{t}\binom{m}{a+b-t}}{\binom{n+m}{a+b}}.$$

The bound is actually defined in terms of a "worst case" over the value of b defined according to:

$$\text{bmax}(m,n,a,\delta) \;=\; \max\{b : \text{Bucket}(m,n,a,b) \geq \delta\}$$

Thus, for any $b > \text{bmax}(m,n,a,\delta)$, if $(a+b)$ balls are drawn out of the $(m+n)$ balls, the probability of getting at least b blue balls is less than δ. In the PAC-MDL bound, a plays the role of $\hat{\sigma}_S$, the number of errors on the train set, and b plays the role of $\hat{\sigma}_{S'}$, the number of errors on the test set, which is exactly the number we wish to bound.

Theorem 1 (PAC-MDL bound [3]) *For any distribution D, for any label description language $L = \{\sigma\}$, with probability at least $(1-\delta)$ over the draw of the train and test sets $S, S' \sim D^{m+n}$:* $\forall\sigma, \quad \hat{\sigma}_{S'} \leq \text{bmax}(m,n,\hat{\sigma}_S, 2^{-|\sigma|}\delta)\}$

Intuitively, the theorem says that if a transductive classifier with a short description length achieves few errors on the train set, then the number of errors on the test set is small with high probability. For details on this theorem, its proof, and its connection to other PAC bounds, see [3]. For now, it is important to note that the applicability of this theorem is only limited by the assumption that the train and test sets are each drawn independently from the distribution D, and that $L = \{\sigma\}$ is a "valid" description language.

3. APPLICATION TO CLUSTERING

In this section, we discuss the application of the PAC-MDL bound to clustering. Two important issues are relevant for clustering bounds:

A. How does a clustering algorithm produce a prediction?

B. What is a "valid" description language for transductive classifiers?

For A, recall that the PAC-MDL bound is applicable to a transductive classifier. It turns out that *any* clustering can be converted into a transductive classifier. Given the entire train set (X^m, Y^m), and X^n from the test set, consider X^{m+n} as the input to the clustering algorithm. Say the clustering algorithm partitions X^{m+n} into k subsets. To find the labels $\hat{Y}^{m+n}$ on all points, first compute the most common label in each cluster using Y^m. Then,

i. if the most common is of class i, $i \in 1, \cdots, \ell$, label *all* points in that cluster as class i;

ii. if there is a tie between two or more class labels, pick one of them uniformly at random and label *all* points in that cluster with that label;

iii. if there are no labeled points in a cluster, choose a label $i \in \{1, \cdots, \ell\}$ uniformly at random and label *all* points in that cluster with that label.

After the assignment of the new labels, all points in each cluster have the same label.

For issue B above, note that the description σ of a classifier can be considered a binary code that contains all the relevant information for generating the labels $\hat{Y}^{m+n}$. Hence, formally speaking, there is a Turing machine that takes σ as an input, produces the labels $\hat{Y}^{m+n}$ as output and halts. Therefore, the description language $L = \{\sigma\}$ must be a prefix-free code (by the halting property) and hence satisfy Kraft's inequality (see [4], Theorem 5.2.1, Lemma 7.3.1, for details), i.e., $\sum_{\sigma\in L} 2^{-|\sigma|} \leq 1$. This is the *only* condition a label description language $L = \{\sigma\}$ must satisfy for the proposed bound to hold.

3.1 PAC-MDL Bound for Clustering

Consider the problem of clustering a dataset X^{m+n}, where m is train-set size and n is the test-set size. For any fixed[1] clustering algorithm, we can construct a description language $L = \{\sigma\}$ by letting each description σ have a constant label on each cluster, as discussed earlier. Given the clustering, this description is sufficient to generate the new labels $\hat{Y}^{m+n}$ over the entire data. Now, if c is the number of clusters and ℓ is the number of labels, then the set of descriptions, i.e., the language $L = \{\sigma\}$, has size at most ℓ^c which can be indexed using only $|\sigma| = c\log\ell$ bits ("fractional bits" are ok here). Since $|L| \leq \ell^c$, it is straightforward to see that Kraft's inequality is indeed satisfied, and hence L is a valid description language. On a more general note, if one assigns a probability measure $p(\sigma)$ to all $\sigma \in L$, since

$$\sum_{\sigma\in L} p(\sigma) = 1 \quad\Rightarrow\quad \sum_{\sigma\in L} 2^{-\log\left(\frac{1}{p(\sigma)}\right)} = 1\ ,$$

$|\sigma| = \log\left(\frac{1}{p(\sigma)}\right)$ always gives a valid bit description complexity for σ. Here, since $|L| = \ell^c$, we have essentially assigned a uniform probability of $p(\sigma) = \frac{1}{\ell^c}$, $\forall\sigma \in L$.

We call the above description language Simple. Using a direct application of the PAC-MDL bound, we get: With probability at least $(1-\delta)$ over the draw of the train and test sets $S, S' \sim D^{m+n}$, the test-set error is bounded by:

$$\hat{\sigma}_{S'} \leq \text{bmax}\left(m, n, \hat{\sigma}_S, \frac{\delta}{l^c}\right) \qquad (1)$$

In practice, clustering algorithms are highly dependent upon random initializations, so the algorithm is run multiple times with the best[2] performing run chosen. Note that the description length of this scheme is higher since the description language must specify which random initialization to use. If the best initialization is chosen from r random initializations, the description length is $|\sigma| = c\log\ell + \log r$. We call this language Init. Applying the PAC-MDL bound for Init, we get with probability at least $(1-\delta)$ over the draw of the train and test sets:

$$\hat{\sigma}_{S'} \leq \text{bmax}\left(m, n, \hat{\sigma}_S, \frac{\delta}{rl^c}\right) \qquad (2)$$

3.2 The Right Number of Clusters

[1]We mean fixed initialization and fixed cluster number here.

[2]"Best" might be defined with respect to some algorithm-specific metric or with respect to bound performance, depending on what you want to evaluate.

Most clustering algorithms need the number of clusters as an input to the algorithm. The PAC-MDL bound provides a natural mechanism for cluster number selection when some label information is available. Consider running a clustering algorithm on a dataset over a range of cluster numbers and picking the cluster number with the tightest bound on the test-set error. This process increases the size of our set of descriptions again. We use a description language for the cluster number c requiring $\log c(c+1)$ bits. This is "legal" because it does not violate the Kraft inequality since: $\sum_{c=1}^{\infty} \frac{1}{c(c+1)} = 1$. One slight optimization is possible here: the nature of our metric disallows the $c = 1$ case, implying that we can substitute $c \rightarrow c - 1$. With this description language, the length of our description is $|\sigma| = c \log \ell + \log r + \log(c(c-1))$ bits. We call the language Cluster. Applying the PAC-MDL bound for Cluster, we get that with probability at least $(1 - \delta)$ over the draw of the train and test sets, the optimal cluster number c^* achieves a test-set error of:

$$\hat{\sigma}_{S'} \leq \text{bmax}\left(m, n, \hat{\sigma}_S, \frac{\delta}{rl^{c^*} c^*(c^*-1)}\right) \tag{3}$$

3.3 The Right Algorithm

In practice, for a given dataset, it is not clear which clustering algorithm is appropriate for use. Typically an algorithm is chosen using domain knowledge, and there is normally no objective way to verify whether the choice made was good or bad. To cope with this, we can extend our description language to specify one of multiple algorithms. More precisely, if we are choosing the best among s clustering algorithms, an extra $\log s$ bits are required to send the index of the clustering algorithm that performs the best. Hence, $|\sigma| = c^* \log \ell + \log r + \log((c^*-1)c^*) + \log s$. Being the best over all algorithms, we call this language Algo. Applying the PAC-MDL bound, we see that with probability at least $(1 - \delta)$ over the draw of the train and test sets, the best algorithm with the optimal cluster number and optimal initialization achieves a test-set error of:

$$\hat{\sigma}_{S'} \leq b_{\max}\left(m, n, \hat{\sigma}_S, \frac{\delta}{r\ell^{c^*} c^*(c^*-1)s}\right) . \tag{4}$$

4. EMPIRICAL RESULTS

We present an empirical study of the proposed evaluation technique on the problem of text clustering for several benchmark datasets using various algorithms.

4.1 Datasets

The datasets that we used for empirical validation and comparison of our algorithms were carefully selected to represent some typical clustering problems: (a) classic3 is a well known collection of documents that contains 3893 documents, among which 1400 Cranfield documents are from aeronautical system papers, 1033 Medline documents are from medical journals, and 1460 Cisi documents are from information retrieval papers; (b) classic2 is a subset of 2860 documents from the classic3 collection formed with the 1400 Cranfield documents and the 1460 Cisi documents; (c) cmu-newsgroup-clean-1000, or, the CMU 20 newsgroups dataset is a widely used text analysis dataset that is a collection of approximately 20,000 messages from 20 different USENET newsgroups, with approximately 1000 messages per group; (d) cmu-newsgroup-clean-100 was formed by sampling 100 messages per group from the full 20 newsgroup dataset; (e) cmu-different-1000 is a subset of the original 20 newsgroups dataset consisting of 3 groups on very different topics: *alt.atheism*, *rec.sport.baseball*, *sci.space*; (f) cmu-different-100 is a subset of (e) formed by sampling 100 documents per topic; (g) cmu-similar-1000 is a subset of the original 20 newsgroups dataset consisting of 3 groups on similar topics: *talk.politics.guns*, *talk.politics.mideast*, *talk.politics.misc*; (h) cmu-similar-100 is a subset of (g) formed by taking 100 documents per topic; (i) cmu-same-1000 is a subset of the original 20 newsgroups dataset consisting of 3 groups on the same topic viz computers, with different subtopics : *comp.graphics*, *comp.os. ms-windows*, *comp.windows.x*; (j) cmu-same-100 is a subset of (i) formed by sampling 100 documents per topic; and, (k) yahoo, or, the Yahoo News (K-series) dataset that has 2340 Yahoo news articles from 20 different categories.

4.2 Algorithms

We experiment with 6 algorithms that have been applied to text datasets with varying degrees of success. Since the motivation behind the experiments is to establish the efficacy of the proposed criterion in evaluation, comparison and model selection for clustering, we have not tried to be exhaustive in the list of algorithms that have been used. However, we have chosen algorithms that represent the state-of-the-art and have been applied to text clustering in the literature. The algorithms we consider are: `SPKMeans` [6], better known as spherical kmeans, that employs the widely used cosine similarity; `FSKMeans` [2], a frequency sensitive version of spherical kmeans; `Hard-moVMF` [1], a generative model based clustering that uses a mixture of von Mises-Fisher (vMF) distributions to model the data; `Soft-moVMF` [1], that also uses a mixture of von Mises-Fisher distributions to model the data with soft-assignments that are finally converted to hard assignments by the standard method assigning a data point to the highest probability cluster `KMeans` [9], the standard kmeans clustering algorithm; and `KLKMeans` [5], better known as information theoretic clustering, that uses KL-divergence between L_1 normalized document vectors instead of squared Euclidean distance in the kmeans framework.

4.3 Methodology

Before performing any experiments, an independent train-test split needs to be made. All experiments reported in this paper were performed on 5 different train-test splits: 10-90, 30-70, 50-50, 70-30, 90-10. On each train-test split, we performed 4 sets of experiments for each dataset corresponding to the 4 bounds discussed in section 3:

(a) Experiments for a particular algorithm, with a fixed cluster number, with a particular initialization. The test-set error-rate bound is computed using (1).[3]

(b) The best results for a particular algorithm with a fixed cluster number, where the best is computed over all the r possible initializations. The test-set error-rate bound is computed using (2).

(c) The best results for a particular algorithm, where the best is computed over all the r possible initializations and all the $c_{\max}$ possible cluster numbers. The test-set error-rate bound is computed using (3).

[3]Code for computing the PAC-MDL bound is available at http://hunch.net/~jl/projects/prediction_bounds/info_theory_n_learning_theory/pac-mdl_bound.tar.gz

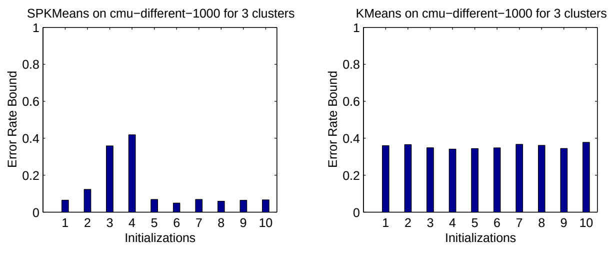

Figure 1: Test set error rate bounds for KMeans and SPKMeans on cmu-different-1000 and cmu-same-1000: 10 runs with different initializations for 3 clusters

(d) The best performance for a given dataset, where the best is computed over all algorithms over all possible cluster sizes and all possible initializations. The bound on the test-set error-rate is computed using (4).

4.4 Results

Now we are ready to present results on the various datasets. We start with the simplest language and using (1) present representative results comparing individual runs of a particular algorithm for a fixed cluster number with different random initializations (Fig 1). Taking the best over 10 initializations for each cluster number, we then compare performance of a particular algorithm over a range of cluster numbers (Fig 2) using (2). Next, by taking the best over the entire cluster number range considered, we compare the performance of various algorithms (Fig 3) using (3). Finally, by taking the best over the best of all the algorithms considered, using (4), we present the best results on particular datasets for various train-test splits (Fig 4-5). Unless otherwise stated, all results are on a 50-50 train-test split.

In Fig 1, we present test-set error-rate bounds for KMeans and SPKMeans on cmu-different-1000 and cmu-same-1000 for 10 runs with different initializations for 3 clusters. All bounds are computed using (1). cmu-different-1000 is a relatively easy dataset in that its labels are reasonably separated being samples from 3 quite different newsgroups. As a result, both algorithms achieve low bounds on the error-rate. SPKMeans performs particularly well since it was designed to be a text clustering algorithm [6]. Over the 10 runs, KMeans achieves a lowest bound of 34.13 % with probability 0.9 in run 4, and SPKMeans achieves a lowest bound of 4.93 % with probability 0.9 in run 6. The best constant class has error-rate 66.67 %. cmu-same-1000 is a relatively difficult dataset since the true labels have significant overlaps. Again, SPKMeans achieves lower bounds than KMeans in most runs, although the bounds are in general higher than those for cmu-different-1000. Over the 10 runs, KMeans achieves a lowest bound of 50.8 % with probability 0.9 in run 7, and SPKMeans achieves a lowest bound of 31 % with probability 0.9 in run 3, the best constant classifier has error-rate 66.67 %.

Next, we consider the best performance over all the 10

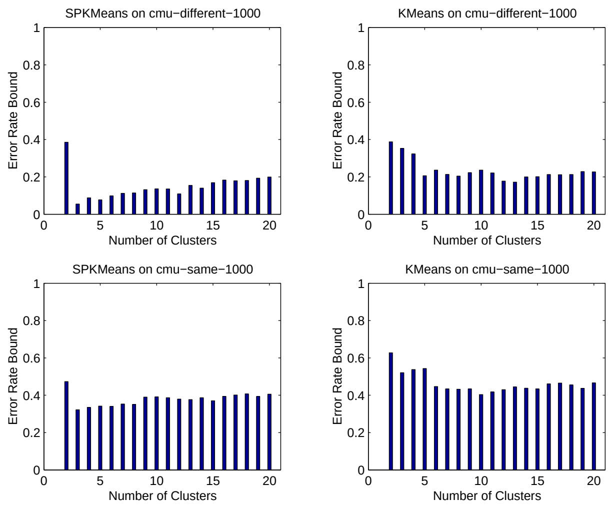

Figure 2: Test-set error-rate bounds for KMeans and SPKMeans on cmu-different-1000 and cmu-same-1000: Best over 10 runs with different initializations over a cluster number range of (2 to 20).

iterations for each cluster number and compare them over a range of cluster numbers. The bound calculation is done using (2). In Fig 2, we present test-set error-rate bounds for KMeans and SPKMeans on cmu-different-1000 and cmu-same-1000 over a range of cluster numbers (2 to 20). We observe that SPKMeans achieves a lower bound than KMeans for most cluster numbers for reasons described above. Over the entire range, SPKMeans achieves a lowest bound of 5.46 % with a probability of 0.9 for 3 clusters. This is marginally higher than the lowest bound of 4.93 % in Fig 1, since the bound calculation uses (2) after incorporating the extra $\log 10$ bits required to index the best over the 10 runs with different random initializations. This is the extra cost for not knowing upfront which random initialization is going to perform the best. This demonstrates the trade-off between improvement in prediction accuracy and considering more runs with different random initializations. On the other hand, KMeans achieves a lowest bound of 17.2 % with a probability of 0.9 for 13 clusters. For cmu-same-1000, over the entire range, SPKMeans achieves a lowest bound of 32.2 % with probability 0.9 for 3 clusters, which is marginally higher than the lowest of 31 % in Fig 1 due to extra description complexity of $\log 10$ bits in indexing the best. KMeans achieves a lowest bound of 40.3 % with probability 0.9 for 10 clusters.

From the results in Fig 2, we make an interesting observation. Note that for SPKMeans, the *optimal number of clusters*, as dictated by the lowest bound over the entire range of cluster numbers considered, is 3 for both the datasets. Interestingly, the number of true labels in both the datasets is 3. This demonstrates how the proposed criterion can be used for *model-selection*, the “right” number of clusters in this particular case, for a given algorithm and a dataset.

Next, we compare the best performance of each of the algorithms, with best taken over all cluster numbers and initializations, on various datasets. This comparison is of great practical interest since this determines the appropriateness of an algorithm for a given dataset. Since the best performance of each algorithm over cluster numbers and initializations is considered, (3) is used to compute the bounds in Fig 3. In Fig 3, we compare the test-set error-rate bounds

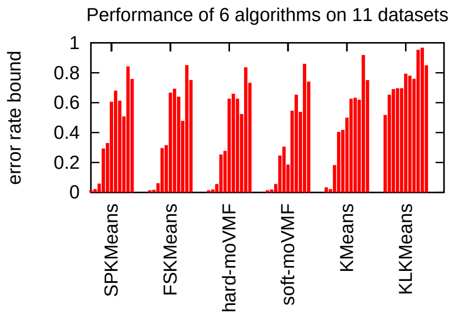

Figure 3: Test-set error-rate bounds for each algorithm on all the 11 datasets: classic2, calssic3, cmu-different-1000, cmu-similar-1000, cmu-same-1000, cmu-different-100, cmu-similar-100, cmu-same-100, cmu-newsgroup-clean-1000, cmu-newgroup-clean-100, yahoo: **Best over all number of clusters and initializations.**

for all the 6 algorithms on all the 11 datasets under consideration. For datasets such as cmu-different-100, cmu-similar-100, cmu-same-100, the low number of samples in high-dimensions make the clustering problem hard for most algorithms. Among the algorithms considered, `Soft-moVMF` performs quite well, e.g., it achieves the lowest test-set error-rate bound of 18.6 % with a probability of 0.9 on cmu-different-100. On the other hand, datasets such as cmu-different-1000 has more samples from the same distribution that makes the clustering problem reasonably simple for quite a few algorithms. We note that 4 algorithms have comparative performances, with `Soft-moVMF` achieving the lowest bound of 5.67 % with a probability of 0.9. It is interesting to note that `SPKMeans` achieves a bound of 5.87 % which is marginally higher than the bound of 5.46 % we observed in Fig 2. The marginal increase is due to the extra description length of $\log((3-1)3)$ bits used to describe the fact that the optimal cluster number is 3. As for the relative performance of the algorithms, as expected, there is no clear winner across all datasets although `Soft-moVMF` appears to win quite often.

We now present best bounds on the test-set error-rate by taking the best over all algorithms, cluster numbers and initializations. We use (4) to compute the bound. Results over 5 different train-test splits on all the datasets considered are presented in Figs 4-5.

classic2 is a relatively simple dataset with only 2 reasonably separate classes. As we see in Fig 4, for all train-test splits, the bound on the test-set error-rate is very low. For the 50-50 train-test split, with probability 0.9 we get a PAC bound of 1.68% on the error-rate on the test-set. (The best constant classifier has error-rate of 48.95%.) This is a remarkably low error-rate bound by PAC standards[4]. classic3 is also a relatively simple dataset with 3 classes. As shown in Fig 4, for the 50-50 train-test split, with probability 0.9 we get a bound of 2% on the error-rate on the test-set. (The best constant classifier has error-rate 62.50%.)

Although cmu-different-100 consists of samples from 3 relatively different classes, the small number of samples and high dimensionality make the problem difficult. As shown in Fig 4, with a probability of 0.9, we get a bound of 20.6% on the test-set error-rate for the 50-50 train-test split. (The

[4]Note that increasing the train-set fraction need not increase prediction accuracy since the clustering algorithms never actually look at the labels.

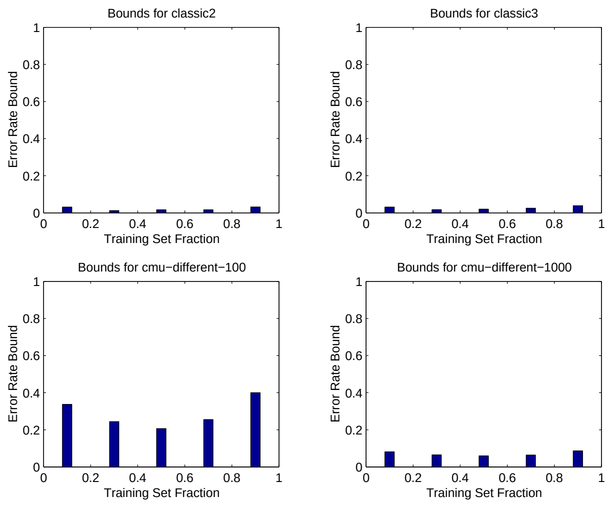

Figure 4: Test set error rate bounds for classic2, classic3, cmu-different-100, cmu-different-1000**: Best over all algorithms, cluster numbers, initializations**

best constant classifier has error-rate 66.67%.) In cmu-different-1000, the extra samples make finding structure in the data easier — a bound of 6 % is obtained.

Due to a large overlap between the underlying labels, both cmu-same-100 and cmu-same-1000 are difficult datasets to get good predictions on by just clustering. As shown in Fig 5, with a probability of 0.9, we achieve error-rate bounds of 64% and 28.4% respectively, while the best constant classifier has error-rate 66.67%.

cmu-newsgroup-clean-100 is a difficult dataset since there are 20 underlying classes with significant overlaps and small number of samples per class. As Fig 5 shows, the lowest bound on the test-set error-rate is 84 % with probability 0.9 on a 50-50 train test split, whereas the best constant classifier has an error-rate of 95 %. For cmu-newsgroup-clean-1000, with increased number of samples from the same problem, a lowest bound of 47.94 % is achieved with probability 0.9.

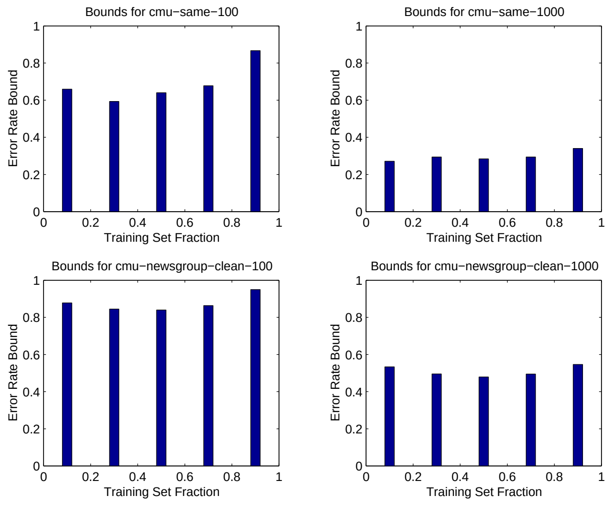

Figure 5: Test set error rate bounds for cmu-same-100, cmu-same-1000, cmu-newsgroup-clean-100 **and** cmu-newsgroup-clean-1000**: Best over all algorithms, cluster numbers, initializations**

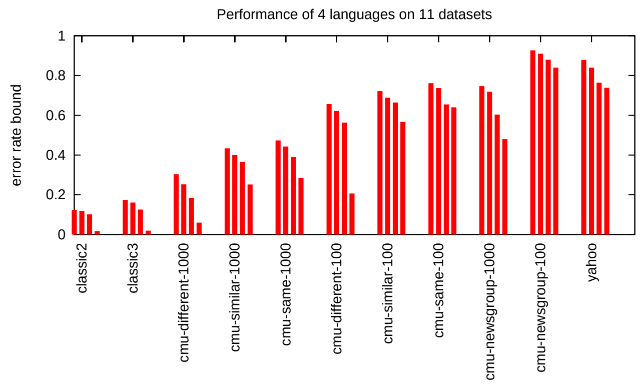

Figure 6: Test-set error-rate bounds on 11 datasets for 4 languages: Simple, Init, Cluster, and Algo.

yahoo is a dataset with 20 underlying classes and 2340 examples. The class portions are highly skewed ranging from as low as 0.0038 to as high as 0.2111. Naturally, unsupervised prediction is nontrivial. The best performance (not shown) achieves a bound of 73.84% with probability 0.9 on the 50-50 train-test split. (The best constant classifier has error-rate 78.89%).

At this point, we make two observations: (i) on all the datasets considered, the test-set error-rate bound is *always* better than the best constant classifier; and, more interestingly, (ii) for most datasets, the bound on the test-set error-rate for the best clustering algorithm is *comparable*, perhaps even *better* than many supervised learning bounds. Clustering appears to capture the structure of labeling in natural datasets.

Finally, we present a comparison between the bounds obtained from the 4 language families considered: Simple, Init, Cluster and Algo, corresponding to Eqns (1)-(4). It is difficult to directly compare languages because each optimizes over a different set of possibilities. For example, it would be unfair to compare the best bound (across all possibilities) for Init to the best bound (across all possibilities) for Cluster, since Cluster takes into account the optimization over all number of clusters while Init does not. To make the comparison fair, we compare the average bound of Init to that for Cluster. Similar arguments apply to other languages. Fig 6 displays comparisons of the 4 languages on all the datasets under consideration. As shown in Fig 6, there seems to be some advantage to using a more complicated language, i.e., trying to optimize over several iterations, cluster numbers and algorithms. In practice, using a more complicated language of course implies more computational effort.

5. RELATED WORK

A clustering algorithm typically tries to optimize its internal quality measure, thereby making objective comparison of various algorithms practically impossible. Note that several unsupervised methods for comparing clusterings, e.g., Jaccard index, Rand index, Fowlkes-Mallows index, Mirkin metric, variation of information etc., exist in the literature (for details, see [9, 10] and references therein). These completely unsupervised methods are incapable of measuring performance in supervised tasks, such as prediction.

Since the predictive ability of clustering algorithms is often key to their successful application, external prediction-related quality measures are often appropriate. Several supervised measures such as purity, entropy, normalized mutual information, supervised F-measure etc. have been used in the literature (see [8] for details). However, it is not clear how these measures are related to the error-rate of the clustering algorithm when used for prediction. Furthermore, none of these supervised measures help with cluster number selection, which is often a big issue for these supervised measures.

An information theoretic external validity measure motivated by the *minimum description length* (MDL) principle has been recently proposed [7]. In spite of having several desirable properties, this measure has a few drawbacks with respect to the measure proposed here: (i) the measure is not normalized to the interval [0,1] (and not easily normalized to exercise that interval) which is desirable in several settings, and many other quality measures actually satisfy this; and, (ii) the measure does not provide guarantees of prediction ability for test data. On a more general note, by attempting to measure the bits required to precisely specify all the class labels in the dataset, the method [7] overlooks the more general possibility of lossy compression, which appears useful and is heavily utilized in our proposed criterion.

6. DISCUSSION

The PAC-MDL bound provides an objective criterion for evaluation, comparison and model selection for clustering that is applicable when the goal of clustering is related to prediction. Experimental results show that this criterion is practically and flexibly useful.

It is particularly striking (and perhaps even shocking) to notice test-set error-rate bounds achieved by the best clustering algorithms are very competitive with various supervised learning bounds on the true error rate of learned classifiers. This good performance suggests that clustering algorithms are doing something fundamentally "right" for prediction purposes on natural datasets.

7. REFERENCES

[1] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Generative model-based clustering of directional data. In *KDD*, pages 19–28, 2003.

[2] A. Banerjee and J. Ghosh. Frequency sensitive competitive learning for balanced clustering on high-dimensional hyperspheres. In *IEEE Transactions on Neural Networks*, May 2004.

[3] A. Blum and J. Langford. PAC-MDL bounds. In *COLT*, 2003.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[5] I. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3(4):1265–1287, 2003.

[6] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[7] B. E. Dom. An information-theoretic external cluster-validity measure. Technical Report RJ 10219, IBM Research Report, 2001.

[8] J. Ghosh. Scalable clustering. In Nong Ye, editor, *The Handbook of Data Mining*, pages 247–277. Lawrence Erlbaum Assoc., 2003.

[9] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, New Jersey, 1988.

[10] M. Meilă. Comparing clusterings by the variation of information. In *COLT*, 2003.