THESIS

EVALUATING CLUSTER QUALITY FOR VISUAL DATA

Submitted by

Maggie Wigness

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2013

Master's Committee:

Advisor: Bruce Draper

Ross Beveridge
Adele Howe
Chris Peterson

ABSTRACT

EVALUATING CLUSTER QUALITY FOR VISUAL DATA

Digital video cameras have made it easy to collect large amounts of unlabeled data that can be used to learn to recognize objects and actions. Collecting ground-truth labels for this data, however, is a much more time consuming task that requires human intervention. One approach to train on this data, while keeping the human workload to a minimum, is to cluster the unlabeled samples, evaluate the quality of the clusters, and then ask a human annotator to label only the clusters believed to be dominated by a single object/action class. This thesis addresses the task of evaluating the quality of unlabeled image clusters.

We compare four cluster quality measures (and a baseline method) using real-world and synthetic data sets. Three of these measures can be found in the existing data mining literature: Dunn Index, Davies-Bouldin Index and Silhouette Width. We introduce a novel cluster quality measure as the fourth measure, derived from recent advances in approximate nearest neighbor algorithms from the computer vision literature, called Proximity Forest Connectivity (PFC). Experiments on real-world data show that no cluster quality measure performs "best" on all data sets; however, our novel PFC measure is always competitive and results in more top performances than any of the other measures. Results from synthetic data experiments show that while the data mining measures are susceptible to over-clustering typically required of visual data, PFC is much more robust. Further synthetic data experiments modeling features of visual data show that Davies-Bouldin is most robust to large amounts of class-specific noise. However, Davies-Bouldin, Silhouette and PFC all perform well in the presence of data with small amounts of class-specific noise, whereas Dunn struggles to perform better than random.

TABLE OF CONTENTS

Chapter 1

Introduction

Digital video cameras have become ubiquitous, and large collections of digital videos are now easily collected and freely available. This provides a vast amount of data for computer vision techniques that learn to recognize objects and actions in video. Unfortunately, while raw video is essentially free, ground-truth labels are time consuming to obtain. The collection of labels, however, is a necessary task for supervised learning techniques, and also for unsupervised learning in order to bridge the gap between learned abstract concepts and natural language. As a result, there is growing interest in techniques that learn to recognize objects and actions from raw videos, with little or no human involvement.

Systems that learn from raw videos typically begin by focusing attention on objects that move. For fixed cameras, this is commonly done using background subtraction [25]. When the camera is in motion, more sophisticated motion segmentation techniques are used [3, 17]. Either way, source videos are divided into collections of smaller and shorter object-centered tracks, where a track is defined as a dynamic region of interest spanning consecutive frames in a video. Training samples are extracted from these tracks either as individual still images or short video snippets, depending on whether the goal is to recognize objects, actions or both. Figure 1.1 shows a single frame from a video with a track (illustrated as a blue rectangle) focused on a person walking.

This thesis is motivated by the need to quickly collect training labels for still images to be used in the context of object recognition. Digital video cameras often capture data at 30 frames per second, yielding a large amount of potential training data if images are extracted from every frame. Even with a sparser extraction rate, the set of training images will grow large for long videos with many tracks. One way to reduce the labeling effort is to cluster the unlabeled samples and have a human annotator label the clusters. This reduces the labeling from $O(\#samples)$ to $O(\#clusters)$.

Figure 1.1: A track (blue rectangle) bounding a person.

Clustering visual data, however, is not a trivial task because images from the same object class have variations in viewpoint, lighting, appearance and background clutter. These variations, combined with high inter-class similarity, conspire such that many clusters may correspond to a single object class, while other clusters may not correlate to any class, but are instead random collections of unrelated objects. To help address these variations in visual data, a fine-grain clustering approach is taken, resulting in a many-to-one mapping between clusters and object classes. However, even this clustering approach is unable to perfectly segment visual data.

Figure 1.2 provides examples of clusters that exhibit varying degrees of quality. True cluster quality is simply a measure of purity, where purity is defined as the percentage of images that come from the dominating object class in the cluster. Thus, we refer to high quality clusters as being "pure" because they can be visually inspected and associated with a single object class label. Conversely, clusters of poor quality contain a mixed set of images, and cannot be accurately described with a single meaningful object class label. In order to collect meaningful information during the labeling process, pure clusters should be selected and displayed to a human annotator for labeling.

This thesis addresses the task of identifying image clusters that are likely to be dominated by instances of a single object class. This task is analogous to cluster validation in data mining, which aims to evaluate the "goodness" of a resulting set of clusters produced by a clustering algorithm. Measures used to evaluate a set of clusters on the whole are adapted to produce individual cluster quality scores to meet the needs of our task. We evaluate three

(a) Highest quality cluster.

(b) High quality cluster.

(c) Poor quality cluster.

Figure 1.2: Examples of learned clusters from the CalTech256 data set with varying degrees of quality.

well-known cluster quality measures from the cluster validation literature: Dunn Index [7], Davies-Bouldin Index [6] and Silhouette Width [22].

This thesis also presents and evaluates a novel cluster quality measure, called Proximity Forest Connectivity (PFC). A Proximity Forest is a forest of randomized metric trees used for approximate nearest neighbor indexing of general metric spaces [21]. It has been shown to be highly accurate for indexing vector data as well as points on Grassmann manifolds [20]. PFC is derived from this neighbor-preserving indexing structure.

1.1 Thesis Contributions

The contributions of this thesis come from the evaluation of three existing cluster quality measures, and the introduction and evaluation of a novel cluster quality measure derived from approximate nearest neighbor indexing. We evaluate the cluster quality measures using the real-world CalTech256 [8] image benchmark data set. Experimental results indicate that

there is no single "best" measure; however, our novel PFC measure is broadly competitive with the existing data mining measures, and is the top performer on more data sets than any of the other measures.

Using synthetic data we model common visual data properties that provide deeper insight regarding which cluster quality measure to select for different structures of data. We show that all three data mining measures from the literature are sensitive to class overlap and over-segmentation, whereas these are PFC's biggest strengths. However, Davies-Bouldin is the most robust in the presence of large amounts of class-specific uniform noise. When smaller amounts of class-specific uniform noise are present, Davies-Bouldin, Silhouette and PFC perform well, whereas Dunn performs poorly.

1.2 Thesis Organization

The rest of this thesis will be organized as follows. Chapter 2 will discuss other approaches used to reduce the labeling workload, and how cluster quality measures have traditionally been used in the data mining community. Next, Chapter 3 discusses properties of visual data and how these properties affect clustering output. A detailed description of the cluster quality measures is provided in Chapter 4, followed by the methodology used to evaluate the measures in Chaper 5. Chapters 6 and 7 present the results of the cluster quality measures on real-world and synthetic data, respectively. Finally, concluding remarks and future work are discussed in Chapter 8.

Chapter 2

Related Work

This chapter reviews sets of literature that either share an underlying motivation or have an analogous task to this thesis. Specifically, we first review machine learning and data mining techniques commonly used on visual data to help reduce the labeling workload. Second, we review data mining literature that compares cluster quality measures in the context of cluster validation, which is used to judge the performance of a clustering algorithm on the whole instead of at the level of individual clusters.

2.1 Reducing Human Labor

Active learning frameworks have been used in many domains to reduce labeling by successfully learning from a subset of training data. A general introduction and survey of the literature is provided by Settles [24]. The key idea associated with active learning is to allow a system to actively choose which unlabeled training samples to learn from. The selection of unlabeled training samples is done iteratively, often through uncertainty sampling [13] based on classifier output. The iterative sample selection halts when the refined classifier exhibits little to no uncertainty. The reduction seen in the labeling workload is defined by the fraction of samples that were left unlabeled. These frameworks have been shown to yield high performance in image classification while significantly reducing the number of samples to label [10, 11], and have helped automate the labeling of video [30, 31].

In the multi-label image domain (i.e., each image may contain multiple objects), an active learning framework was introduced that predicts the tradeoff between annotation cost and information gain [29]. Annotation in this framework may include providing an object label for a system-defined segmented region in the image, or completely segmenting the image by hand. The workload associated with these two types of annotation are very different. Thus,

the framework tries to determine when a difficult annotation task is more worthwhile than an easy annotation task.

Active learning frameworks and the long term goal of this thesis are both centered around reducing the labeling workload, but the general approach is quite different because the selection criteria try to accomplish different tasks. Active learning frameworks attempt to reduce the labeling effort by quickly finding a smaller subset of training data that does not compromise performance. Cluster quality evaluation, however, only takes place after the clustering algorithm has finished learning how to group samples, and the labeling effort is reduced by identifying pure clusters.

Research in unsupervised object discovery also helps reduce the labeling workload, and has many similarities to this thesis work. Object discovery is the task of learning object-level models from a pool of unlabeled images. This task is motivated by the desire to quickly summarize the number of object classes that exist in an unlabeled data set. Most unsupervised object discovery approaches are based on either clustering [5, 26] or latent topic models [14, 23, 26]. Both techniques produce groups of images, where each group is intended to represent a different object class. This act of grouping by object class is exactly the approach that we take in this thesis to help reduce the labeling workload.

An iterative object discovery approach [12] attempts to learn object-level models easiest to hardest. At each iteration the unlabeled samples are agglomeratively clustered until the compactness of clusters start to degrade, and then a single cluster with the best Silhouette width is selected for discovery. This iterative discovery approach emphasizes that some object classes are easier to discover (or cluster) than others. This is a similar view that we share in this thesis and one that we attribute to properties of visual data that degrade intra-class similarity (discussed in Chapter 3).

Although the grouping approach is common between unsupervised object discovery and this thesis work, the biggest difference can be seen at the granularity of the grouping. The above mentioned approaches try to force a one-to-one mapping between clusters and object classes. In theory, this results in the optimal solution to reduce the labeling effort as each

class would only need to be labeled once, but in reality this perfect mapping cannot be achieved for large visual data sets. We recognize that clustering visual data has many challenges, and try to overcome these challenges with a fine-grained clustering of images (i.e., an over-segmentation of the data). However, even over-segmenting the data does not produce perfect clusters, which further motivates the need for cluster quality measures in order to later guide the labeling process. Chapter 3 provides a detailed discussion of the visual data challenges that motivate our fine-grain clustering approach.

2.2 Cluster Validation

Clustering is a common technique used to group data based on common patterns or similarities. Many clustering algorithms have emerged; each modeling clusters slightly differently. For example, clusters can be modeled using distance, density, or statistical distributions. Often, clustering is used on unlabeled data as a form of unsupervised learning. Unlabeled data, however, provide very little context for the appropriate number of clusters to learn, which makes it hard to determine whether or not the clustering was successful.

Cluster validation has emerged in the data mining community to help determine if a set of data was clustered successfully. For unlabeled data sets, cluster validation is performed using internal evaluation. This simply means that there is no ground truth that can be used to directly evaluate cluster purity. Instead, evaluation is based on the data set itself. Internal cluster validation measures often look for high intra-cluster similarity and low inter-cluster similarity. This indicates that the formed clusters are compact and well-separated from other clusters.

Internal measures can be used to compare the output of different clustering algorithms, but more often the measures are used to help identify the true number of clusters in a data set (i.e., find the best value of k). The data mining literature often refers to this comparison of different partitions of the data as relative cluster validation, instead of internal cluster validation. This is a necessary task because many clustering algorithms require a priori

knowledge of the number of clusters to group the samples into. For unlabeled data sets, this is a difficult parameter to define, but relative measures can guide this selection.

Just as there are many clustering algorithms, there are also many relative cluster quality measures. Comparisons of relative cluster quality measures have been conducted to determine if a single "best" measure exists. Halkidi, Batistakis and Vazirgiannis [9] compare four measures (including Davies-Bouldin and variance) on four synthetic data sets: three 2-dimensional sets and one 6-dimensional set. The parameters used to construct these four synthetic data sets are omitted from the paper. However, the 2-dimensional data sets are visually presented as figures, and contain clusters of data that are mostly well-separated. The experimental results indicated that there was no clear best cluster validation measure. Each of the measures identified a clustering output that had a justifiable number of clusters for each data set. The authors conclude that the best measure may be application dependent, depending on whether there should be an emphasis on cluster separation or compactness.

Vendramin, Campello and Hruschka [28] provide a fuller comparison by evaluating 40 relative cluster quality measures (many of which are variations of a single measure), including Davies-Bouldin, Dunn and Silhouette. The comparison is conducted on 108 synthetic data models, each replicated 9 times. The models are differentiated by three parameters. The first is data dimensionality: 2, 3, 4, 22, 23, and 24. The second is the number of ground truth clusters: 2, 4, 6, 12, 14, or 16. Finally, the last parameter defines the three potential distributions of samples per cluster. Either the 500 samples were distributed as evenly as possible among the clusters, or one cluster was constructed with a percentage (10% or 60%) of the samples while the remaining were split as evenly as possible. The data samples were drawn from a multivariate normal distribution, with the exception that cluster boundaries were not permitted to overlap in the first dimension of the data, resulting in each cluster occupying a disjoint region in feature space.

Results from the 108 data models suggest that Silhouette width is the most robust to all data set variations: number of samples, number of clusters and dimensionality. However, many other measures performed better than Silhouette under specific conditions. Vendramin,

Campello and Hruschka conclude that their results are likely to extend to similar data sets, but not necessarily to different structures of data.

The results from the above mentioned comparative cluster validation papers leave us with the belief that common data mining cluster quality measures are capable of identifying successful clustered output, but that different measures handle sources of variation within the data differently. Although our motivation is different, the use of such measures can be adapted to describe the quality of individual clusters. In this thesis we evaluate three commonly used data mining measures to determine if we can identify the sources of variation in visual data that each measure handles best.

Chapter 3

Clustering Visual Data

When clustering data as a means of reducing labeling effort, the optimal labeling reduction occurs when 1) each cluster contains data from a single class, and 2) no two clusters represent the same class. This means that an annotator only has to provide one label for every object class. This is an ideal labeling situation, and may be realistic with well-separated data with high intra-class similarity (e.g., data used in much of the cluster validation literature from data mining [9, 28]), but this thesis is predicated on the idea that the structure of visual data typically does not exhibit those properties.

The task of a clustering algorithm is to identify a common pattern of behavior, and then group data based on these learned patterns. Feature representation plays a crucial role in helping the algorithm learn the patterns that we are interested in identifying. When clustering visual data in the context of object recognition, the desired learned similarity would be the common object class. Unfortunately, regardless of feature representation, many sources of variation in visual data can cause low overall intra-class similarity for object classes. Four main sources of variation are viewpoint, illumination, appearance and background clutter. In addition to these intra-class variations, inter-class similarity likely increases as the number of object classes in a data set grows. The remainder of this chapter discusses these variations in greater detail, and how we attempt to deal with these challenges when clustering visual data.

3.1 Intra-Class Variations

Objects from the same class often look different under various viewpoints. For example, the profile or frontal view of a person, and the side view or rear view of a car. Research by Murase and Nayar [18] indicates that images from the same object class do not trivially fall

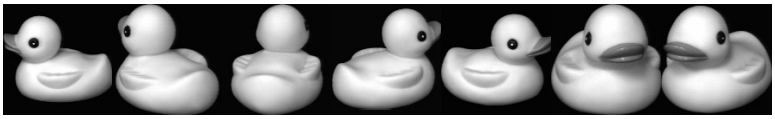


Figure 3.1: Example images from an object in the COIL-100 data set. The selected images have 50° pose variations.

into the same area of feature space. Thus, attempting to learn a single cluster per object class could be difficult given that the images are scattered throughout feature space and not in a tightly formed group.

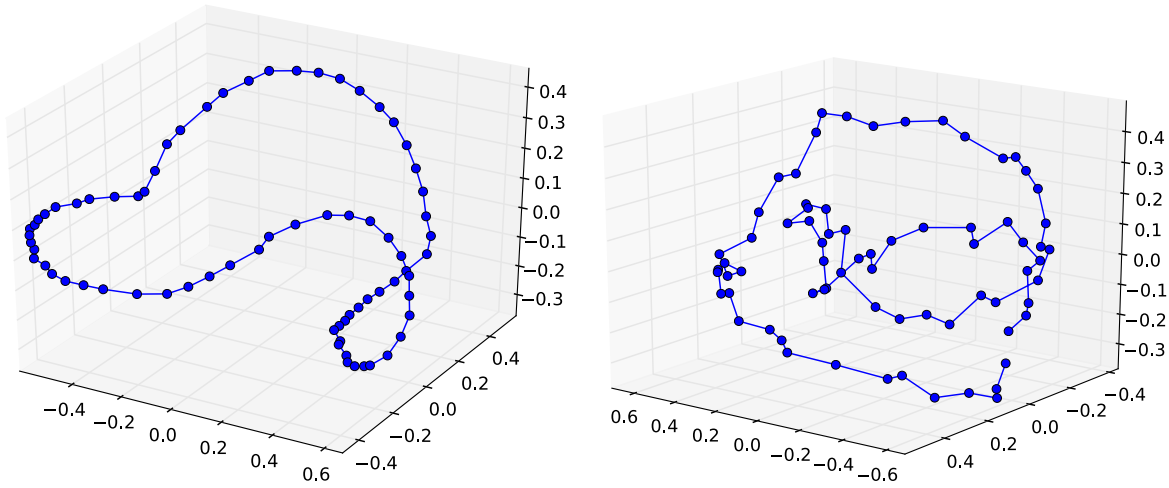
The experiments of Murase and Nayar included putting objects on turntables and considering the set of images generated as the objects rotated through 360°. They showed that the images form a closed 1D curve in high-dimensional image space (or PCA projections thereof). The curve is closed because a full rotation brings the object back to its original position. If the object rotates in 2 dimensions, the images form a closed 2D surface. Since rotating an object is equivalent to changing the viewing angle, the set of images of a single object under changes in viewpoint is a closed 2D surface in image space.

We illustrate and confirm the findings of Murase and Nayar using the images of a single object from the COIL-100 data set[1] [19]. This data set contains images of 100 objects. Each image represents a viewpoint at 5° intervals as the object is rotated through 360° on the turntable, resulting in 72 images per object. Image sizes are normalized to 128x128 pixels. A subset of the images for the object used in our mini experiment can be seen in Figure 3.1.

In the first part of this replicated experiment, images are represented as a vector of their raw pixel values (i.e., a point in a 16,384-dimensional space), and are normalized by subtracting the mean image value. The images are then PCA projected into a 3-dimensional object eigenspace; the results can be seen in Figure 3.2a. As Murase and Nayar describe, the images form a closed 1D curve, demonstrating that images of the same object class lie in different locations of feature space due to variations in viewpoint.

---

[1] http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php



(a) Raw pixel values.

(b) 50-dimensional feature vector.

Figure 3.2: 3-dimensional eigenspace representation of the object from Figure 3.1 with different feature representations.

Feature representation can help with this challenge because certain features try to extract consistent properties that minimize the effects of viewing angle. Unfortunately, Burns and Weiss [4] proved that there are no general case perspective-invariant features. Instead, affine-invariant features produce nearly constant values over a range of viewpoints, effectively breaking the 2D surface into small patches with nearly constant feature values.

In the second part of the replicated experiment, we use the same object images from the COIL-100 data set, but instead use a Bag of Features representation (discussed in greater detail in Chapter 6) with SIFT descriptors [15], and create a 50-dimensional feature vector. Note that SIFT descriptors are affine-invariant. Figure 3.2b shows the formed 1D curve that the images make up after again being projected into a 3-dimensional object eigenspace. Notice that the curve is not as smooth as the one seen in Figure 3.2a when using the raw pixel values. Although it is hard to tell from the static 3D plots, the projected samples in Figure 3.2a are more evenly spaced, whereas some of the projected samples in Figure 3.2b tend to be more closely bunched together with larger distances between these bunches. Therefore, even with features that attempt to address the challenges associated with

different viewing angles, images from the same object class under different viewpoints are still scattered throughout feature space, which makes it difficult to group all these images into a single cluster.

Illumination is a second source of variation that is seen in visual data. Objects are viewed under many different light sources, both artificial and natural. Basri and Jacobs [1] showed that changes in illumination produce images that span a 9 dimensional subspace, given matte surfaces and a fixed viewpoint. Therefore every point on the 2D surface in image space representing object viewpoints is actually a sample from a 9 dimensional space of possible illuminations. Since features are not insensitive to illumination, every view of an object produces distributions of feature values due to lighting.

A third variation seen among visual data is appearance, and is the most diverse of the variations discussed thus far. Appearance in the context of images most often pertains to color and shape of objects. The particular source of appearance variation depends on the object class. Artificial objects such as airplanes and bicycles come in almost every color, so color features are unreliable while shape features may be robust. Natural objects such as plants, on the other hand, assume a variety of shapes, so shape features tend to vary while color features may be more stable.

Finally, background clutter in visual data introduces features that can distract from the primary object of interest. Background clutter can be introduced when objects are not centered and tightly cropped in image samples. When extracting samples from raw video for example, tracker failures may create tracks that do not tightly bound the object of interest. However, system failures are not the only time that background clutter is introduced. Many inanimate objects are often put in motion by regularly mobile objects such as people. For example, people ride bikes and carry boxes. Thus, it is not uncommon to see parts (e.g., arms) of these motion activators in image samples of inanimate objects. Also, some objects are commonly found together although they represent different object classes. For example, baseballs are often seen near baseball gloves and bicycles often have accessories such as saddle bags. These extra objects are not necessarily always seen with the object of interest though.

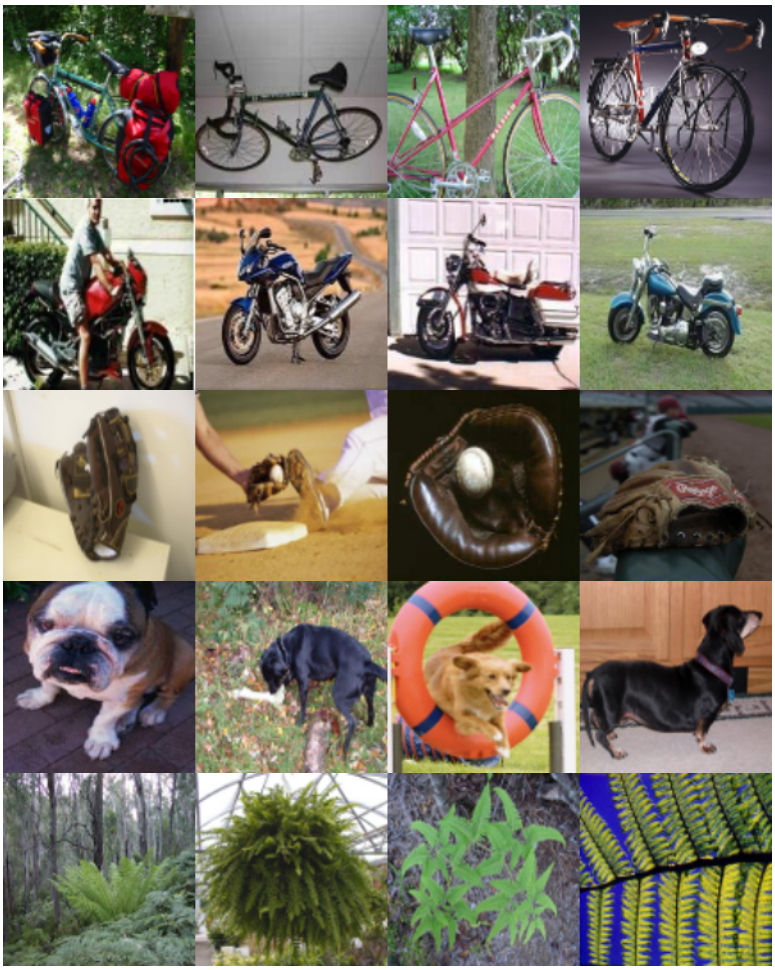


Figure 3.3: Example images from five object classes in the CalTech256 image data set. Notice the visual data variations that each class exhibit.

Figure 3.3 shows example images from five object classes (one class per row) in the CalTech256 data set. These examples illustrate some of the intra-class variations that visual data often exhibit, and that cause images from the same object class to reside in different areas of feature space.

3.2 Inter-Class Similarity

Intra-class variations make it difficult to cluster visual data, but high inter-class similarity creates further confusion during clustering. High inter-class similarity arises when different object classes have common features. For example, bicycles and motorcycles (rows one and two in Figure 3.3) both have two wheels and handlebars. In large data sets, many classes

may fall under the umbrella of a general object class which almost necessitates that those classes have common features. For example, chimps, bears, camels, fish, and dogs are all animals (these classes are all found in the CalTech256 data set). Some features will not be common to all animal classes, e.g., only fish have gills. However, all of the animal object classes have eyes, and most have legs.

3.3 Fine-Grain Clustering

The discussion of intra-class variations, particularly those caused by changes in viewpoint, illumination and appearance, indicates that images from the same object class can be spread throughout different areas of feature space. This spread of samples from the same object class makes it extremely difficult to force a one-to-one correspondence between clusters and object classes. The addition of noise caused by background clutter and class overlap from high inter-class similarity further complicates the one-to-one mapping task.

With this in mind, we adopt a hierarchical clustering technique that produces a many-to-one correspondence between clusters and object classes. This over-segmentation of the data provides greater opportunity to differentiate between classes with high inter-class similarity, and allows images from the same object class to create groups in various parts of feature space because of intra-class variations. Using a hierarchical clustering algorithm allows us to construct clusters without specifically defining a value of k. This hierarchy can be cut at different levels to achieve different clustering granularities.

Chapter 4

Cluster Quality Measures

We now describe the five cluster quality measures used for evaluation. First, we establish a baseline method that measures intra-cluster variance. Next, the data mining cluster quality measures, Davies-Bouldin Index, Dunn Index, and Silhouette Width, are defined, followed by the novel Proximity Forest Connectivity measure. Each measure is defined so as to produce a quality score for individual clusters. A measure's sorted cluster scores, highest to lowest quality, can be used as guidance for selectivity when labeling. The five measures will be compared and evaluated based on where they place pure clusters in their suggested labeling order.

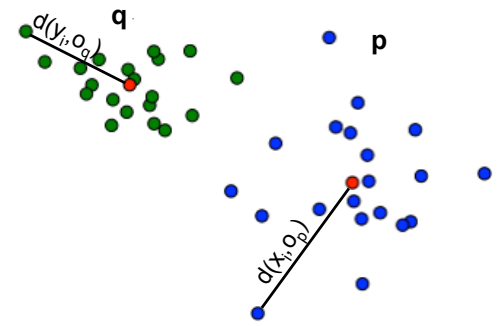
The following are some common variable definitions used in the descriptions below. It is assumed that n samples are clustered producing a set of clusters, C, such that $|C| = k$ and $p, q \in C$. For the purposes of explanation, p is the cluster of interest when calculating a quality score, and q may be any other cluster within the partitioned set. Cluster p contains a set of data points $\{x_i | x_i \in p\}$, and the cluster centroid, o_p, is the mean of these points. Cluster q's data points will be denoted as $\{y_i | y_i \in q\}$. $d(a, b)$ denotes the Euclidean distance between points a and b. Finally, the nearest neighbor of p is defined as $nn_p = min_{q \in C, p \neq q}\{d(o_p, o_q)\}$.

4.1 Variance

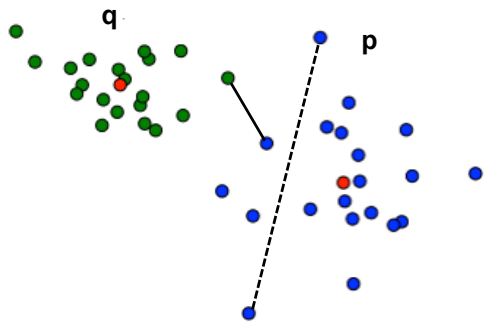
The baseline measure used in this evaluation is variance, and is defined as the average squared distance of each data point to the cluster centroid. More formally:

$$V_p = \frac{1}{|p|} \sum_{i=1}^{|p|} d(x_i, o_p)^2 \tag{4.1}$$

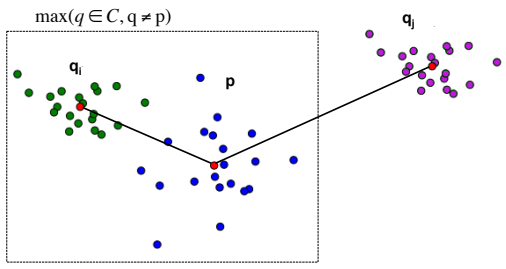
This baseline measures intra-cluster similarity as a way to indicate the compactness of a cluster. Highly compact clusters comprise samples that are near each other in feature space, which could be a good indicator that the samples come from the same object class. Clusters



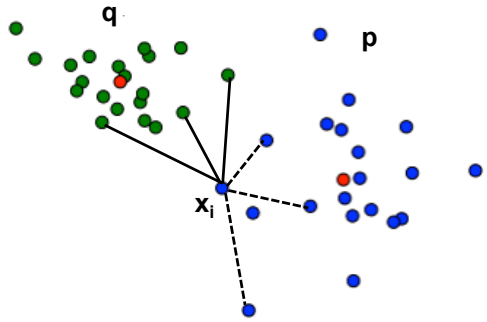
(a) Variance illustration where cluster q is more compact than p, indicating higher cluster quality.



(b) Dunn illustration of the maximum intra-cluster distance and minimum inter-cluster distance.



(c) DB illustration where clusters q_i and q_j have the same scatter resulting in cluster q_i producing the maximum DB score based on its dispersion from p.



(d) Silhouette illustration of some intra-cluster and inter-cluster similarities that would be used to calculate the silhouette width for point x_i.

Figure 4.1: Illustrations of cluster quality measures.

with lower variance are considered to be more compact and therefore would suggest a higher cluster quality (e.g., see Figure 4.1a).

4.2 Dunn Index

There are many variations of Dunn indices [7], but at the core, each one is the ratio of inter-cluster to intra-cluster similarity. Formally, the measure is defined as

$$D_p = \min_{q \in C, p \neq q} \{ \frac{\delta_{p,q}}{\Delta_p} \} \tag{4.2}$$

where $\delta_{p,q}$ measures inter-cluster similarity between clusters p and q, and Δ_p measures the intra-cluster similarity of cluster p. The variations of Dunn indices are differentiated by their definitions of δ and Δ. We use the original suggested definitions [7], where inter-cluster

similarity is the minimum distance between any pair of data points across clusters, i.e. $\delta_{p,q} = \min_{x_i \in p, y_j \in q}\{d(x_i, y_j)\}$, and intra-cluster similarity is the maximum distance between a pair of data points within the same cluster, i.e. $\Delta_p = \max_{x_i, x_j \in p}\{d(x_i, x_j)\}$. The intra-cluster distance could also be thought of as the diameter of cluster p.

To compute Equation 4.2 we need the pair-wise distances between all n samples in the data set. This can be computationally expensive for large data sets, so we do not exhaustively search for the minimum score produced for all $q \in C$. Instead, we restrict the computation to the cluster that is the nearest neighbor of p. Thus, the final measure is defined as

$$D_p = \frac{\delta_{p,nn_p}}{\Delta_p} \tag{4.3}$$

To confirm that only using the nearest neighbor of p does not drastically change the Dunn index scores, a set of experiments were run on small data sets comparing the results of Equations 4.2 and 4.3. It was found that $\sim 23\%$ of the time, the equations returned the same results, meaning that the cluster that produced the minimum score selected in Equation 4.2 was actually nn_p. However, when different scores were computed, the absolute difference between the scores was very small, suggesting that calculating the Dunn index using Equation 4.3 does not drastically change the results. Details of this experiment can be found in Appendix A.

The ratio of inter-cluster to intra-cluster similarity will produce larger scores for clusters that are well-separated and compact. Thus, larger scores indicate higher cluster quality. Figure 4.1b illustrates the minimum inter-cluster distance (solid line) and maximum intra-cluster distance (dotted line) when calculating the Dunn Index.

4.3 Davies-Bouldin Index

The Davies-Bouldin (DB) index [6] captures the average similarity of a cluster and its most similar cluster. It is formally defined as

$$DB_p = \max_{q \in C, p \neq q} \frac{S_p + S_q}{M_{pq}} \tag{4.4}$$

where S_p is the measure of scatter or intra-cluster similarity for p, and M_{pq} is the measure of dispersion or inter-cluster similarity between clusters p and q. Just as with Dunn, there are many variations of the DB index depending on the definitions of scatter and dispersion. We again use the original suggested definitions [6], where dispersion is the distance between cluster centroids, i.e., $M_{pq} = d(o_p, o_q)$, and scatter is the average distance of each data point to its cluster's centroid, i.e., $S_p = \frac{1}{|p|} \sum_{\forall x_i \in p} d(x_i, o_p)$.

The computational complexity of DB is much less burdensome than what was seen when calculating Dunn since inter-cluster distance is defined using cluster centroids, not individual data samples. Thus, we do exhaustively search for a score against all $q \in C$. Overall, smaller DB scores indicate better cluster quality since DB is a ratio of intra-cluster to inter-cluster similarity. However, cluster p takes the maximum score for all $q \in C$ because the maximum score will identify the cluster most similar to p. For example, if the scatter S_q were equal for all clusters, the most similar cluster would be defined as the nn_p. This is shown in Figure 4.1c where clusters q_i and q_j have the same variance, but the centroid of cluster q_i is closest to the centroid of p, making it the most similar to p. Alternatively, if there were a choice between two clusters with centroids equidistant from p, then the cluster with larger dispersion S_q would produce the maximum score.

4.4 Silhouette Width

The Silhouette width [22] measure defines how well a single data point x_i fits within its cluster,

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max(a(x_i), b(x_i))} \tag{4.5}$$

where $a(x_i)$ is the average intra-cluster similarity between $x_i \in p$ and the remaining data points in p, and $b(x_i)$ is the average inter-cluster similarity between data point $x_i \in p$ and all $y_i \in q$ where q is a neighboring cluster. Similarity between two elements is defined as $d(x_i, x_j)$. Figure 4.1d illustrates some of the similarities calculated for $a(x_i)$ as dotted lines, and some of the similarities calculated for $b(x_i)$ as solid lines.

Typically, the cluster with the lowest average similarity is defined as the neighboring cluster, which again, like Dunn index, requires the computation of pair-wise distances between all n samples. Thus, we define the neighboring cluster as nn_p. Appendix A also includes detailed experimental results regarding how the selection of the nearest neighbor affects the true Silhouette width score. However, $\sim 42\%$ of the time the nearest neighbor returned the true Silhouette width score so we conjecture that this modification would not drastically change our results.

Silhouette values close to 1 mean that the data point is well matched in its cluster, where values close to -1 indicate the data point may fit better in a different cluster. The final quality score for cluster p is the average of its elements' Silhouette widths:

$$S_p = \frac{1}{|p|} \sum_{i=1}^{|p|} s(x_i) \tag{4.6}$$

4.5 Proximity Forest Connectivity

Our novel cluster quality measure is derived using the relative locality of data points to gauge cluster quality. The idea is that, in general, the neighbors of a sample should be in the same cluster as the sample, and neighboring samples in feature space likely represent the same object class. Thus, clusters containing many neighboring samples in feature space are likely to be pure. Approximate Nearest Neighbor (ANN) queries can be exploited to quickly identify neighboring data in feature space. Well known ANN indexing structures include forests of randomized kd-trees and hierarchical k-means, but both methods assume data is from a vector space.

O'Hara and Draper [21] introduced a new ANN indexing structure called a Proximity Forest which is a set of randomized metric trees. The metric tree structure was defined by Uhlmann [27] for indexing general metric spaces, a generalization that cannot be made by forests of kd-trees and hierarchical k-means, whose hyperplane partitioning requires a vector space. Although visual data is commonly represented using vector data, not all visual data representations fall into this category. One example includes videos of human actions

represented as points on Grassmann manifolds [16]. Each point on the manifold defines a subspace, making common vector data ANN indexing structures inadequate. O'Hara and Draper, however, show that a specialized Proximity Forest, called the Subspace Forest, is capable of efficiently and accurately providing approximate neighbor queries for subspaces [20].

Generalizing to metric spaces allows the Proximity Forest ANN indexing structure to be used for a wider variety of data; however, within this thesis, images are still represented as vector data. Thus, any ANN indexing method could be used to encode relative locality of our image data. However, we suspect that the success of our novel cluster quality measure will, in part, be due to the reliability of the ANN retrieval. O'Hara and Draper presented experimental results showing that the Proximity Forest yielded substantially more accurate querying results than kd-trees and hierarchical k-means on three real-world vector data sets [21]. The Proximity Forest's superior querying performance was seen in experiments that controlled for data dimensionality, data set size, distance functions and the number of nearest neighbors returned by the query. We used this superior performance of the Proximity Forest as our motivation for its selection as the ANN indexing structure for our novel cluster quality measure, Proximity Forest Connectivity (PFC).

A Proximity Forest is simply a set of Proximity Trees. Each tree will randomly partition the entire data set by splitting samples based on distance to a randomly selected pivot element. To construct a Proximity Tree, data samples are randomly added to an initially empty node in the tree. τ defines the splitting threshold, at which a random pivot point is selected from the elements in the node. The distance between the pivot element and the samples is calculated, and the elements are split into two child nodes: one containing samples with distance less than the median, and the other containing the remaining samples. This process is done recursively until all samples have been added to the tree.

Once all the data has been added, the leaf nodes of a tree define approximate neighborhoods, and thus encode relative locality information of the data. We use the Proximity Forest to determine how often a cluster's samples are grouped together in the leaf nodes of the t trees. Samples that often reside in the same leaf nodes are neighbors in feature space,

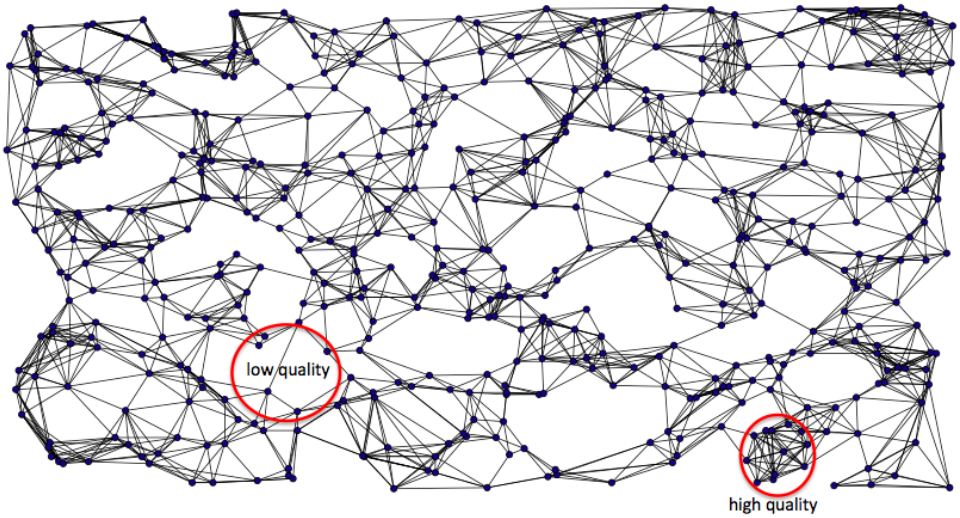


Figure 4.2: 2-dimensional illustration (generated using the publicly available Proximity Forest source code[1]) of highly connected samples, denoted with an edge, within the Proximity Forest. This connectivity model is used to generate PFC scores for clusters of data points.

indicating that they may represent the same class. The connectivity score between two data samples is the number of trees in the forest where they reside in the same leaf node:

$$c(x_i, x_j) = \sum_{k=1}^{t} l_k(x_i) == l_k(x_j), \tag{4.7}$$

where $l_k(x_i)$ is the leaf node of tree k that contains element x_i, and the $==$ comparison returns 1 if true and 0 if false. A sample's cluster connectivity score is the average connectivity between it and all other samples in the cluster:

$$cc(x_i, p) = \frac{1}{|p|} \sum_{j=1}^{|p|} c(x_i, x_j) \tag{4.8}$$

Finally, the PFC measure of a cluster is the average of its samples' cluster connectivity

$$PFC_p = \frac{1}{|p|} \sum_{i=1}^{|p|} cc(x_i, p) \tag{4.9}$$

where higher connectivity scores indicate better cluster quality.

[1]http://sourceforge.net/projects/proximityforest/

Figure 4.2 shows randomly generated 2-dimensional data points, where edges between points denotes highly connected data in the Proximity Forest. This visual representation of PFC was generated using a forest of 27 trees, and high connectivity is defined as a pair of samples residing in at least 9 of those trees, i.e. $c(x_i, x_j) \geq 9$. The figure illustrates two hypothetical clusters that could be formed, indicated with red circles. These clusters are intended to show how clusters with data samples that are highly connected would indicate high cluster quality, whereas clusters with few highly connected data samples would be considered a low quality cluster.

In the experiments conducted for this thesis 21 trees are generated, and the splitting threshold, τ, is set to 20. These parameter values were found to work well when using the Subspace Forest for action recognition [20] so we extend them to our application as well.

Chapter 5

Evaluation Methodology

The remainder of this thesis is devoted to the evaluation of the five cluster quality measures described in Chapter 4. This work is motivated by the need to selectively label clusters extracted from a large collection of unlabeled videos, namely the Mind's Eye Year 2 data set[1]. Unfortunately, the lack of ground truth labels in the Mind's Eye data set makes it a poor choice for evaluation. Labeled data allow us to compute the purity of clusters which can be used to quantitatively evaluate the performance of the cluster quality measures. Thus, the evaluation is performed using the labeled CalTech256[2] [8] image benchmark data set, and synthetic data sets that are designed to model visual data properties.

For each experiment, the following steps are taken to evaluate the cluster quality measures. First, the data samples are agglomeratively clustered using Ward's linkage and Euclidean distance. In keeping with the theme of over-clustering to handle the challenges discussed in Chapter 3, data are grouped until every cluster contains at least 10 but not more than 20 samples. Next, each of the five measures computes a quality score for the k clusters. The ranked ordering of these scores (highest to lowest quality) serves as each measure's suggested labeling order.

The primary form of evaluation will be presented as precision-recall curves. We calculate precision and recall at every index in the ordered list of scores, i.e., for growing subsets of clusters that could be returned for labeling. Precision is the percentage of clusters in the

[1] http://www.visint.org

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech256/

subset that are pure, and recall is the percentage of all pure clusters that exist in the subset. Formally, precision and recall are defined using true/false positives/negatives (tp,tn,fp,fn):

$$precision = \frac{tp}{tp + fp}$$
$$recall = \frac{tp}{tp + fn}$$

In this evaluation a true positive is the selection of a pure cluster, where a cluster is considered to be pure if at least 50% of its samples come from the same class. Problem difficulty is defined by the number of pure clusters that exist; if most of the clusters are pure then almost any selection order will perform well. However, as the number of pure clusters starts to drop, the selection order becomes more important. The percentage of pure clusters is provided in each precision-recall plot to provide context for the degree of difficulty. Note that the percentage of pure clusters should be a good indicator of the precision for a random selection measure; however, for completeness we include a curve in the precision-recall plots that illustrates the performance of a random ordering of the clusters. The random curve is always an average over 100 trials.

Additionally, the average F-measure for each precision-recall curve is provided in the legend of the figures. F-measure is the harmonic mean of precision and recall:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

The value of F_1 will fall in the range $[0.0, 1.0]$, but we present an average F-measure across the entire precision-recall curve, and even a perfect curve will not receive a 1.0 value. Thus, these measures should be used as relative comparisons of the cluster quality measures, and should not be directly compared to the range $[0.0, 1.0]$.

We realize that the hard threshold used to define a pure cluster is not ideal. Any number of other thresholds could have been used. However, we provide alternative evaluation figures in Appendix B that show curves of the ground truth cluster purity using each measure's ordering of clusters. A good ordering of clusters would result in the early part of the curve remaining high while the end of the curve drops low. The general trends of these alternative plots are consistent with the trends seen in the precision-recall plots. This indicates

that although the purity threshold influences the precision-recall curves, a change to that threshold would likely affect all the curves in a similar fashion, resulting in the same relative ordering of the measures. The figures in Appendix B are provided only for the synthetic data experiments. The real-world data experiments for which we present precision-recall plots have a high degree of difficulty, a large number of clusters, and are not smoothed over 100 trials like the synthetic data, making the alternative figures for these data sets difficult to read. Thus, they are omitted.

Chapter 6

Evaluation of Cluster Quality Measures Using Real-World Data

The evaluation of cluster quality measures on real-world data uses the CalTech256 data set which contains 30,607 still images of 256 object classes plus a "clutter" class composed of background images (e.g., trees, sky, grass, walls). Although taken from the web, these images are small and mostly object-centered, with variations in viewpoint, scale, illumination and appearance. They are therefore similar to images that might be sampled from video tracks. The images from Figure 3.3, which depict examples of variations of visual data properties in different object classes, are from the CalTech256 data set.

The first part of this chapter evaluates the cluster quality measures using different feature representations for the CalTech256 data set. Feature representation plays a significant role in determining the output of a clustering algorithm. If the feature representation is unable to encode the visual concepts in the data set, then the images will not be grouped by object class very effectively. Most feature representations selected for image data sets attempt to be robust to common visual data variations such as those discussed in Chapter 3, but feature relevance is often class specific, meaning that any representation will face weaknesses on data sets with many object classes. We explore two different feature representations on the CalTech256 data set to see how the cluster quality measures perform under different qualities of clustered output. Both feature representations, however, have a difficult time clustering the entire data set with great success.

The second part of this chapter evaluates the cluster quality measures on subsets of 20 classes from the CalTech256 data set. We introduce the subsets because of the poor clustering output observed using the entire data set on both feature representations (details of this output are discussed in the next two sections). Using subsets of object classes makes the

clustering problem easier, and ensures that the performance of the cluster quality measures seen using the entire data set did not face a floor effect because of the extreme difficulty of the task. The subsets, however, each display varying degrees of difficulty, providing further insight as to when measures perform best.

6.1 Bag of Features Representation

A Bag of Features (BoF) model is a common representation for visual data. This representation models an image as an unordered collection of local features extracted at points of interest. The model is adapted from the Bag of Words representation used in textual information retrieval tasks where documents are represented by a histogram of their word frequencies. Analogously, images are represented by the frequency of their visual features. A set of local features from the training set are clustered to learn a discrete visual vocabulary, often called a dictionary or codebook. The feature vector for an image is the histogram describing how many of its interest points fell into each codebook entry.

To collect the visual words, the image is broken into small image patches. The extraction of local features from a patch represents a single visual word. A variety of local feature descriptors and codebook sizes can be used, but we use publicly available feature vectors[1] [26] that densely sample local features as SIFT [15] descriptors. These features are matched to a codebook with $1,000$ entries.

Clustering the CalTech256 images using the BoF representation generates 1,724 clusters. Note that this is roughly a factor of seven over-segmentation relative to the 256 object classes. Only 15% of clusters are considered to be pure, and Figure 6.1 shows the precision-recall curves indicating the performance of the five cluster quality measures.

Dunn and PFC perform best of all the measures, but Dunn displays a superior performance after about 15% of the pure clusters are identified (i.e., recall). Although Silhouette

[1] http://homes.esat.kuleuven.be/~tuytelaa/unsup_features.html

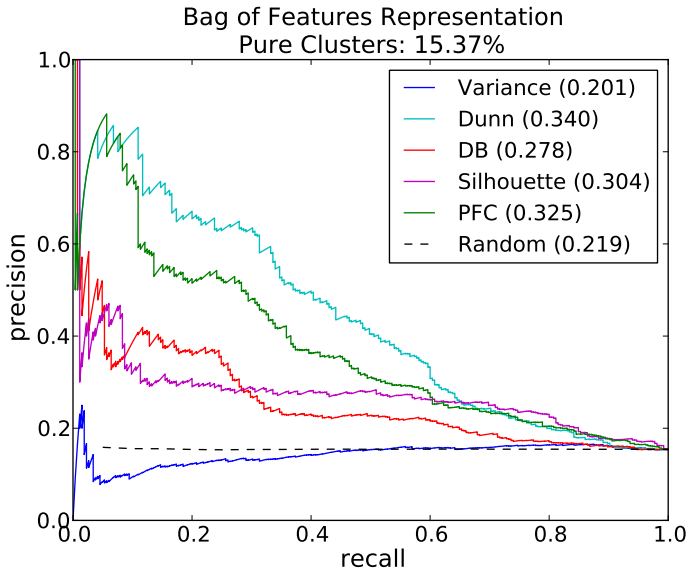


Figure 6.1: Precision-recall curves for CalTech256 using a $1,000$-dimensional BoF representation.

and DB are less successful than Dunn and PFC, they clearly outperform the variance baseline measure which performs slightly worse than random. If we were content to draw conclusions from a single experiment, we would be tempted to conclude that Dunn and PFC are the best cluster quality measures. Unfortunately, the relative ordering of the measures depends on the choice of features.

6.2 Global Color Histogram Representation

The SIFT descriptors used in the previous experiment capture local shape information. This experiment uses features that capture global color information by mapping pixels into the 11-dimensional named color space [2], and representing every image by its 11-dimensional color histogram. With these feature vectors, 1,888 clusters are generated, of which only about 5% are considered pure. It is not surprising that the number of pure clusters drops significantly. Many of the objects in the CalTech256 data set are artificial and do not have standard colors (e.g., bicycles). This provides little chance for a clustering algorithm to group by object class when only color information is available. Not surprisingly, the classes

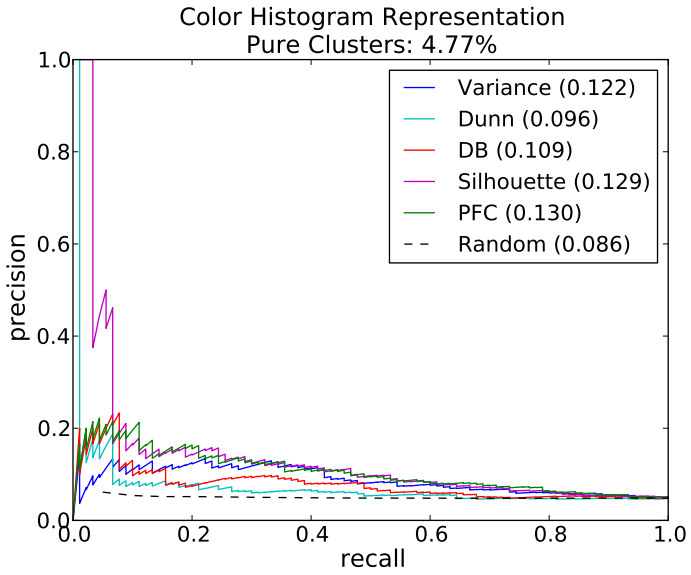


Figure 6.2: Precision-recall curves for CalTech256 using an 11-dimensional color histogram representation.

represented by the pure clusters have a consistent color; examples include the flowers iris and hibiscus.

Figure 6.2 shows the precision-recall curves for the quality measures applied to the clusters generated using only color features. The relative rankings of the quality measures are quite different than those seen with the BoF representation. Silhouette and PFC perform best, but PFC's consistency receives a slightly better average F-meausure even though it starts out weaker than Silhouette. Dunn and DB start competitively, but drop below the variance baseline measure after only 10% of pure clusters are identified.

Between the two experiments there is little consensus as to which cluster quality measure performs best. At this point we can only conclude that the data mining measures and PFC are able to identify pure clusters better than a random selection criteria. These two experiments, however, had a high degree of difficulty; very few clusters that were formed were pure. In the next section, we simplify the real-world data set to see how this affects the performance of the measures.

Table 6.1: List of the 20 object classes hand selected for easy discrimination.

American flag	diamond ring	dice
fern	fire extinguisher	fireworks
French horn	ketch	killer whale
leopards	mandolin	motorbikes
pci card	rotary phone	roulette wheel
tombstone	Pisa tower	zebra
airplanes	faces-easy	

6.3 CalTech256 Subsets using BoF Representation

To further evaluate the cluster quality measures, 14 subsets of the CalTech256 data set are used for evaluation. The BoF representation is used for these subsets since it produced superior clustering results relative to the 11-dimensional color histogram representation. All but one subset comprise 20 object classes. These subsets are used by Tuytelaars et al. [26] when they compare several unsupervised object discovery techniques. One subset is hand selected to represent 20 "easy" classes. These classes (listed in Table 6.1) are considered to have low inter-class similarity, making them the easiest to differentiate. The remaining subsets are simply put together based on the order in which the object classes appear in the benchmark data set class listing.

The average F-measure results for all 14 subsets can be found in the top part of Table 6.2. The subset ordering in the table is sorted by problem difficulty, easiest to hardest, as defined by the percentage of pure clusters produced for the subsets. The highest F-measure (i.e., top performance score) for each subset can be seen in bold. For each subset, the top performing measure was evaluated for statistical significance. We ran the Wilcoxon rank-sum test to obtain p-values, and any statistically insignificant results are italicized in the table. The level of significance is evaluated at $\alpha = 0.01$. The bottom part of the table provides a summary of each measure across all subsets. This summary includes the average and standard deviation (σ) of the results, and Spearman's rank correlation coefficient, ρ, measuring the relationship between the F-measure values and the percentage of pure clusters.

Table 6.2: Average f-measure summary on subsets of the CalTech256 data set.

Subset	% Pure	Variance	Dunn	DB	Silhouette	PFC	Random
20 Easy	84.47	0.633 ± 0.255	**0.638 ± 0.257**	0.561 ± 0.248	0.609 ± 0.261	0.632 ± 0.259	0.598 ± 0.244
241-256	71.79	0.606 ± 0.236	**0.654 ± 0.242**	0.545 ± 0.233	0.583 ± 0.260	0.593 ± 0.231	0.559 ± 0.219
221-240	59.52	0.464 ± 0.209	0.532 ± 0.195	**0.566 ± 0.179**	0.547 ± 0.190	0.543 ± 0.185	0.513 ± 0.192
141-160	54.29	0.532 ± 0.171	**0.625 ± 0.179**	0.436 ± 0.175	0.532 ± 0.202	0.565 ± 0.179	0.492 ± 0.179
81-100	48.84	0.425 ± 0.172	0.499 ± 0.174	0.498 ± 0.169	0.486 ± 0.173	**0.540 ± 0.183**	0.465 ± 0.164
101-120	46.77	0.448 ± 0.172	0.460 ± 0.163	0.409 ± 0.172	0.451 ± 0.178	**0.475 ± 0.159**	0.451 ± 0.159
121-140	44.70	0.400 ± 0.163	0.437 ± 0.179	0.501 ± 0.139	0.438 ± 0.181	**0.544 ± 0.132**	0.444 ± 0.153
1-20	43.26	0.463 ± 0.156	0.471 ± 0.160	0.476 ± 0.140	0.490 ± 0.180	**0.508 ± 0.174**	0.436 ± 0.148
161-180	42.31	0.377 ± 0.166	0.431 ± 0.140	**0.498 ± 0.141**	0.452 ± 0.151	0.492 ± 0.122	0.433 ± 0.146
201-220	40.38	0.417 ± 0.134	0.402 ± 0.149	0.448 ± 0.122	0.439 ± 0.126	**0.505 ± 0.137**	0.421 ± 0.139
41-60	39.82	0.405 ± 0.129	0.354 ± 0.133	**0.559 ± 0.151**	0.437 ± 0.130	0.486 ± 0.136	0.417 ± 0.139
21-40	38.10	0.304 ± 0.132	0.380 ± 0.143	**0.430 ± 0.093**	0.366 ±0.167	*0.423 ± 0.122*	0.405 ± 0.132
61-80	35.45	0.357 ± 0.100	0.428 ± 0.126	0.418 ± 0.112	0.390 ±0.117	**0.443 ± 0.123**	0.391 ± 0.124
181-200	33.33	0.366 ± 0.131	0.364 ± 0.145	0.402 ± 0.125	0.428 ± 0.147	**0.467 ± 0.116**	0.378 ± 0.118
Average	48.79	0.443 ± 0.166	0.477 ± 0.170	0.482 ± 0.157	0.475 ± 0.176	**0.515 ± 0.161**	0.457 ± 0.161
σ	13.95	0.090	0.097	0.057	0.068	**0.056**	0.061
ρ		0.903	0.934	0.618	0.921	0.881	**1.000**

Variance and Silhouette fail to perform best on any of the subsets. However, in most cases Silhouette does perform better than random unlike the variance baseline measure which performs worse than random more times than not. This leaves Dunn, DB and PFC as the top performers for the different subsets. PFC performs best on half of the subsets, and is the best performer on average. Additionally, PFC displays the most consistent behavior across the subsets as its F-measure values have the smallest standard deviation, although DB displays a comparable consistency.

Dunn performs best on three of the four subsets that have the largest percentage of pure clusters, and its value of ρ indicates that its performance is the most correlated to the percentage of pure clusters in a data set (excluding the random selection method). This trend suggests that Dunn will perform well on easy data sets, and poorly on hard ones. Silhouette, variance and PFC also have high rank correlation coefficients, whereas DB is far less correlated. On the whole, the values of ρ confirm that the percentage of pure clusters found in a data set is a good indicator of the problem difficulty.

The results of the subset experiments still indicate that none of the cluster quality measures present themselves as the best for all data sets. However, variance is clearly not a good cluster quality measure candidate as it struggles to perform better than random on most data sets. Additionally, Silhouette is consistently out-performed by at least one of the remaining cluster quality measures, Dunn, DB and PFC. Finally, our novel PFC measure presents itself as a competitive cluster quality measure based on its superior performance on many of the subsets and highest overall average performance.

Chapter 7

Experiments Using Synthetic Data

The real-world data experiments from Chapter 6 are sufficient to conclude that no single measure universally retrieves better clusters than the alternatives. Therefore, it becomes essential to understand as best we can what aspects of a data set relate to one measure doing better than the others. This can be a difficult task, but controlled experiments on synthetic data with clear distinguishing attributes is a good first step.

We design several sets of synthetic data that model some of the most dominant characteristics that we have experienced with real-world visual data. Each model generates 20 different processes (simulating object classes) in a 50-dimensional space. Each class centroid is selected from a uniform distribution over the range $[-1.0, 1.0]$ in every dimension. In most instances this will result in some class centroids being near each other while others are far from all other classes. This simply models different levels of inter-class similarity that is common in visual data. Finally, 100 samples are assigned to each process whose values are drawn from a Gaussian distribution with σ being varied throughout the different experiments.

Since the synthetic data sets are randomly generated, we average 100 iterations of each experiment to ensure that evaluation is not performed on a particularly hard or easy data set. The above mentioned parameters produce a data set that is about one-tenth the size of the full CalTech256 data set, but is similar in size to the subsets used for evaluation in the previous chapter.

7.1 Spherical Gaussian Processes

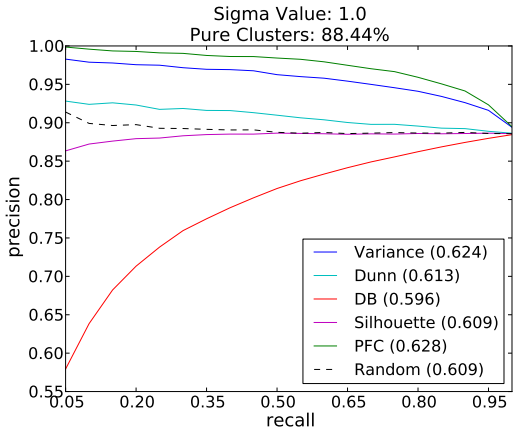
The first synthetic model consists of simple spherical Gaussian processes. We run five experiments using this model, each with a different value of σ when generating the class

samples. σ values are chosen to ensure that the generated data cannot be trivially clustered (i.e., every cluster perfectly represents a single class). As the value of σ increases in the different experiments, more overlap among processes can be seen, causing the number of pure clusters to decrease which in turn makes the evaluation of clusters more difficult.

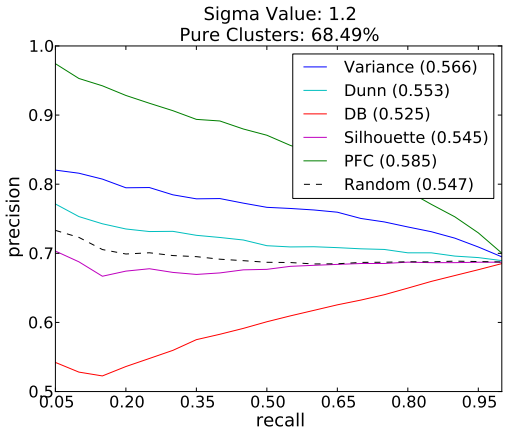
The first experiment uses $\sigma = 1.0$, and on average the data is over-segmented into 120 clusters to represent the 20 underlying processes. The clustering results are much better than those seen in the real-world data; by our definition, 88.2% of the clusters are considered to be pure. Figure 7.1a shows the precision-recall curves generated for the cluster quality measures for this first experiment. The curves look nothing like those for the CalTech256 data shown in Figures 6.1 and 6.2. PFC performs best, but is only slightly better than the baseline variance. DB performs worse than random, while Silhouette and Dunn perform essentially at random.

We posit that PFC and variance perform well on this first experiment because they do not require separation between clusters when evaluating cluster quality. PFC only requires that samples share a cluster with their nearest neighbors; variance requires that points be near the mean. DB, on the other hand, depends on the distance between cluster centers. When the data is over-segmented, cluster centers that represent the same class tend to be close to each other in feature space. As a result, DB performs worse than random on this data. Silhouette and Dunn also rely on inter-cluster distances, although at the level of samples, not cluster centers.

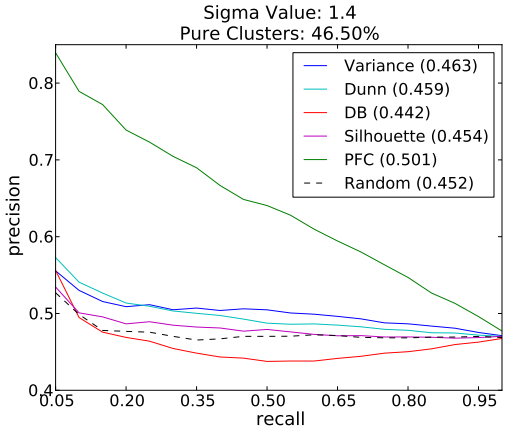
To test this hypothesis, we repeated the experiment for larger values of σ, whose precision-recall curves are shown in Figures 7.1b-7.1e. The relative ordering of measures remains the same when $\sigma = 1.2$ and $\sigma = 1.4$, while the distinction between PFC and variance becomes more significant. We suspect this is because relatively more of the pure clusters lie near the edges of the hypercube, where more diffuse samples may be generated by a unique process but are still nearest neighbors. To easily visualize this suspicion, Figure 7.2 depicts 2-dimensional synthetic data generated from increasing σ values, where each data sample color defines its ground truth class. Notice that as σ increases (from Figure 7.2a to 7.2c) the



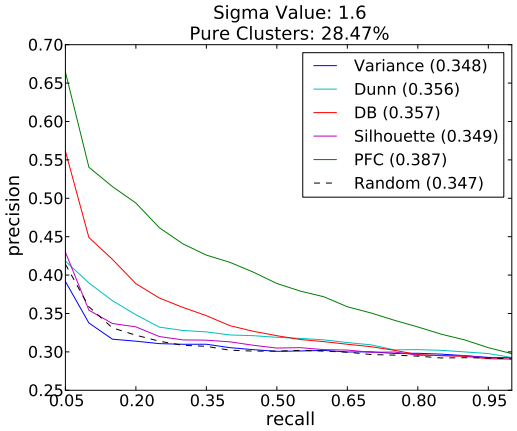
(a) $\sigma = 1.0$



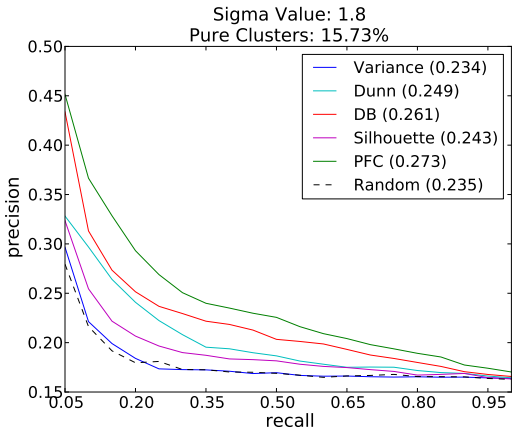
(b) $\sigma = 1.2$



(c) $\sigma = 1.4$



(d) $\sigma = 1.6$



(e) $\sigma = 1.8$

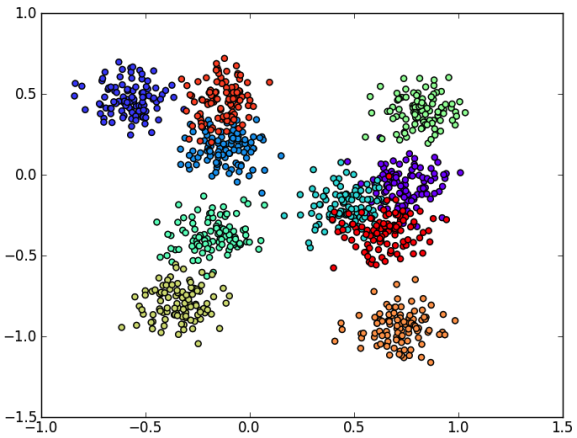
Figure 7.1: Precision-recall curves for 50-dimensional spherical Gaussian processes.

interior of the hypercube becomes very congested with samples from many classes, making it difficult for a clustering algorithm to learn pure clusters in the interior region. However, samples from the same class can be seen together near the edges of the hypercube, making it conceivable that pure clusters could be generated in these outer regions. Although this suspicion is verified using 2-dimensional data, we suspect that spherical Gaussian processes will exhibit similar behavior in higher dimensions.

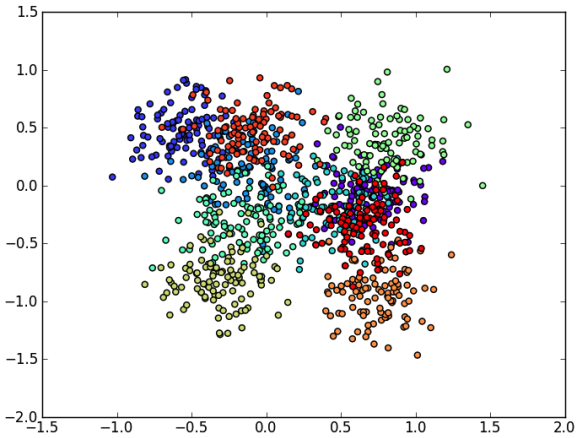
Once $\sigma = 1.6$, a shift in the ordering of the measures can be seen, sending variance to the bottom, and moving DB to the second best performing spot. This shift in ordering strengthens our hypothesis about DB being susceptible to over-segmentation. Again, the 2-dimensional data in Figure 7.2 shows that as σ increases, samples from a class are spread throughout a larger area of feature space, which creates more class overlap near the interior of the hypercube, and pure clusters lying near the edges of the hypercube. Even if the data are over-segmented, the pure cluster centers will drift further to the edges of the hypercube, creating more separation between clusters which is a property that DB uses to determine cluster quality.

Table 7.1 shows the average distance between centroids for a cluster and its nearest neighbor for the σ values used in this first synthetic data model. We report the average distance for pure and mixed (low quality) clusters separately. Notice that when σ is small, the average separation for pure clusters is lower than the separation seen for mixed clusters, which may have helped cause the poor performance of DB. Once $\sigma = 1.6$, however, the separation for pure clusters exceeds the separation seen among mixed clusters, and this is the point in which DB started to perform better than the other data mining measures. The increase in separation for pure clusters is slight as σ increases from 1.0 to 1.8, but the trend in the data does exist.

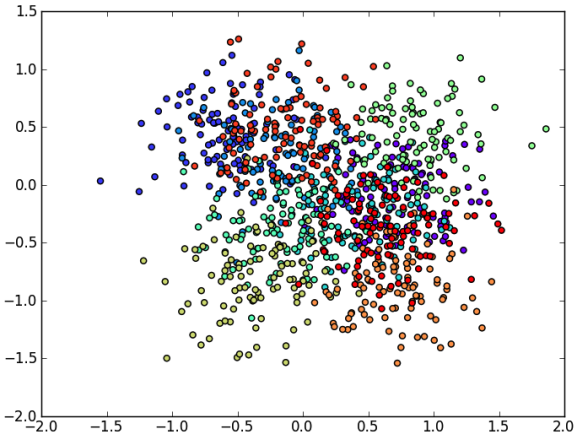
To summarize, this first synthetic data model has shown that PFC is the most robust to spherical overlapping data. We hypothesize that the poor performance from the data mining measures is due to the fact that the data set is over-segmented. This hypothesis is tested more directly with the second synthetic data model.



(a) Small σ.



(b) Larger σ.



(c) Largest σ.

Figure 7.2: 2-dimensional visualization of spherical Gaussian processes generated using our synthetic data model. Sample color defines its ground truth class. As σ increases, more class overlap is seen, and pure clusters will most likely be formed from the diffuse samples near the edges of the hypercube.

Table 7.1: Average centroid distance between a cluster and its nearest neighbor.

σ Value	Pure Clusters	Mixed Clusters
1.0	3.700	4.125
1.2	3.950	4.168
1.4	4.131	4.197
1.6	4.235	4.217
1.8	4.290	4.232

7.2 Less Over-Segmentation

The second synthetic model generates data in the same manner as the first spherical Gaussian model, but decreases the amount of over-segmentation by setting a minimum cluster size of 40 (instead of 10). This roughly quarters the number of clusters and the amount of over-segmentation. Figure 7.3a shows the precision-recall curve for the first experiment of the second model that uses $\sigma = 1.0$. The relative ordering of the measures is the same as was seen in the first experiment of the first model (Figure 7.1a), however, the performance of all the measures improves. The most significant improvement can be seen in DB which is now performing far above random. The problem difficulty (percentage of pure clusters) is approximately the same in both models, leading to the conclusion that the amount of over-segmentation produced during clustering plays a significant role in the performance of the cluster quality measures, specifically DB.

This conclusion is further strengthened with three more experiments using different sigma values. These results are seen in Figures 7.3b-7.3d. DB's performance, relative to the other measures, improves as the value of σ increases, just as in the first model, but is now performing much better than random and is highly competitive with the performance of PFC. The performance of PFC does not change in terms of the relative ordering of the cluster quality measures. However, the F-measure of PFC also improves with less over-segmentation, although the degree of improvement is less drastic than what is seen for DB.

Table 7.2 summarizes and compares the F-measure scores between the first and second synthetic data models, which directly shows the effects that over-segmentation has on all the cluster quality measures. All measures improve with less over-segmentation for $\sigma = 1.0, 1.2$

Table 7.2: Summary of the effects that over-segmentation has on all cluster quality measures. The F-measures for each cluster quality measure improve with less over-segmentation (i.e., models where $40 \leq |p| \leq 80$) given that the problem difficulty is similar.

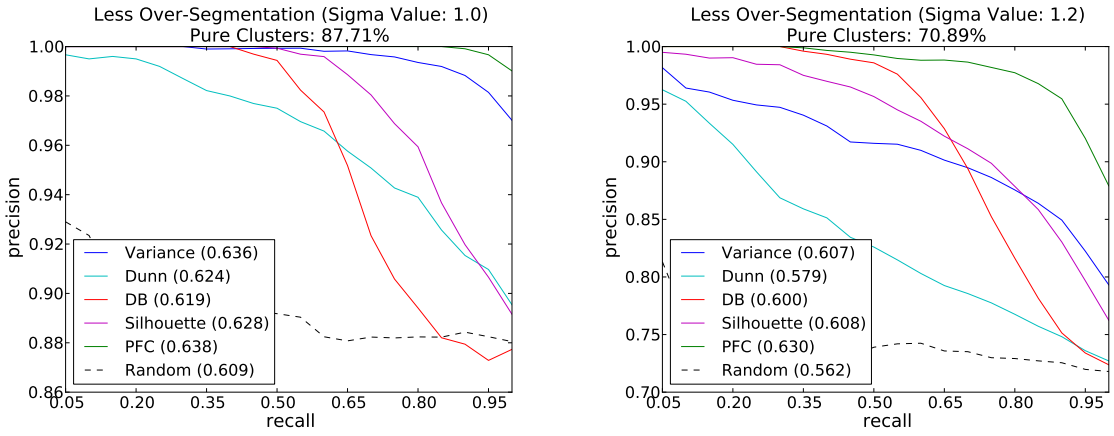
Model	% Pure	Variance	Dunn	DB	Silhouette	PFC
$\sigma = 1.0, 10 \leq \|p\| \leq 20$	88.44	0.624	0.613	0.596	0.609	0.628
$\sigma = 1.0, 40 \leq \|p\| \leq 80$	88.55	0.637	0.625	0.621	0.630	0.638
$\sigma = 1.2, 10 \leq \|p\| \leq 20$	68.49	0.566	0.553	0.525	0.545	0.585
$\sigma = 1.2, 40 \leq \|p\| \leq 80$	70.89	0.607	0.579	0.600	0.608	0.630
$\sigma = 1.4, 10 \leq \|p\| \leq 20$	46.50	0.463	0.459	0.442	0.454	0.501
$\sigma = 1.4, 40 \leq \|p\| \leq 80$	43.72	0.477	0.461	0.523	0.497	0.560
$\sigma = 1.6, 10 \leq \|p\| \leq 20$	28.47	0.348	0.356	0.357	0.349	0.387
$\sigma = 1.6, 40 \leq \|p\| \leq 80$	12.86	0.232	0.238	0.296	0.254	0.319

and 1.4. The improved performance of all the measures suggests that over-segmentation should be minimized when possible. However, comparing the results for $\sigma = 1.6$ suggests that over-segmentation is inevitable, particularly for data coming from highly overlapping classes because the cluster quality measures perform better with more over-segmentation. Notice that the problem difficulty for the model with less over-segmentation is twice as hard, leading us to believe that as data classes overlap more, a fine-grained clustering of the data is necessary to learn pure clusters.

The third synthetic model steps away from the discussion of the effects of over-segmentation. Instead, the final synthetic data model addresses the visual data property of class-specific irrelevant features.

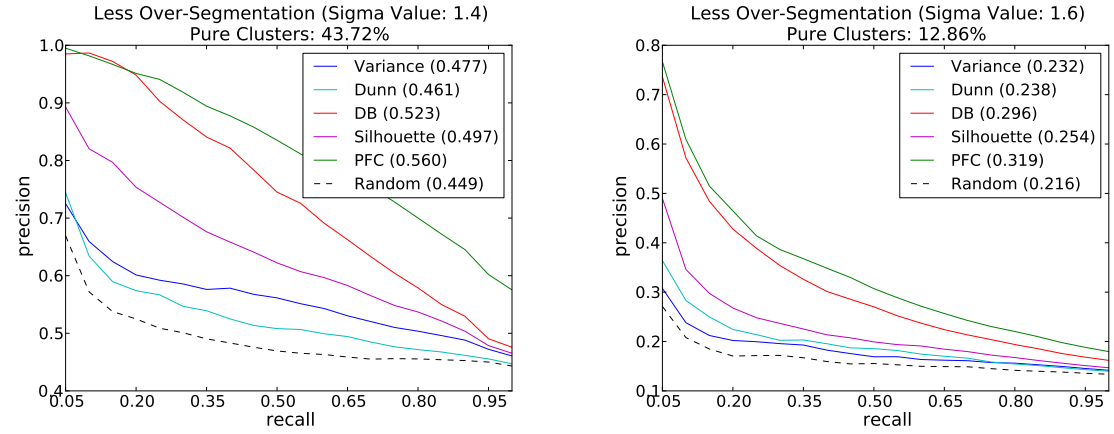
7.3 Class-Specific Noise

Collectively, the figures above suggest that the data mining measures may perform worse than PFC on over-segmented data sets with overlapping processes. This is useful, but the figures above still look different from the precision-recall curves computed from the CalTech256 data set. Something else is going on. In the discussion of visual data properties in Chapter 3, we noted that feature relevance is class specific. For example, color features may be meaningful for the class iris but not for the class bicycle. We therefore created a third model with class-specific noise. For every class, N dimensions are picked at random and



(a) $\sigma = 1.0$

(b) $\sigma = 1.2$



(c) $\sigma = 1.4$

(d) $\sigma = 1.6$

Figure 7.3: Precision-recall curves for 50-dimensional spherical Gaussian samples with minimum cluster size of 40.

given values selected from a uniform distribution in the range $[-5, 5]$ to simulate an irrelevant feature. Figure 7.4a shows the precision-recall curves for data with 5 noisy dimensions.

This time, the relative ordering of the cluster quality measures is totally different. DB, which performed so poorly before, becomes the best performing measure, while Silhouette goes from second-worst to second-best. Meanwhile, variance goes from highly competitive to near the bottom, PFC goes from first to the middle of the pack, and Dunn drops to last. Some of this re-ordering can be explained. It is not surprising that Dunn suffers from the added noise since its intra-cluster similarity is measured as the cluster diameter. Even a few irrelevant dimensions causes the diameter to expand, resulting in noisy measures.

Overall, the ordering is quite stable for this third model with class-specific uniform noise. Figures 7.4b - 7.4e shows the precision-recall curves after randomizing 10, 15, 20 and 25 dimensions, respectively. Overall performance drops with the increase in noise and decrease in signal, but the relative ordering of the quality measures is the same.

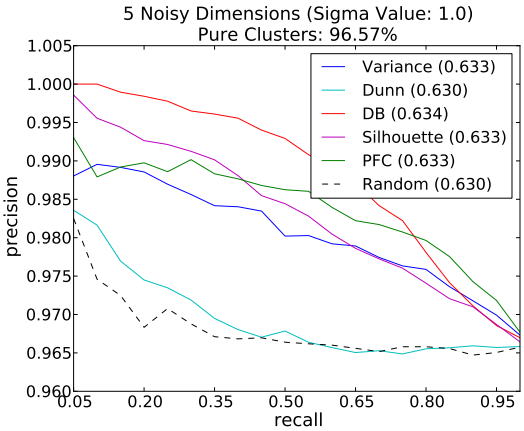
A similar explanation given for the improved DB performance in the first synthetic data model can be used to help explain DB's superior performance in this third synthetic data model. Table 7.3 shows the average distances between cluster centroids for pure and mixed clusters for the set of noisy dimension experiments. When there are no noisy dimensions added to the data, the mixed clusters have the best centroid separation relative to their nearest neighbor, which is why we hypothesized the DB performed so poorly in the first synthetic data model. However, as soon as noisy dimensions are added to the data, the separation for pure clusters exceeds that of the mixed clusters (although the difference is slight) which may be part of the reason that DB outperforms the other measures. Notice, however, that the average centroid separation for pure clusters does not continually increase as the number of irrelevant dimensions increases, suggesting that the irrelevant dimensions do cause additional noise instead of simply creating more separation for the pure clusters near the edges of the hypercube.

Irrelevant dimensions do seem to have an effect on the performance of PFC. The relative order suggests that PFC's performance drops significantly, but comparing the actual per-

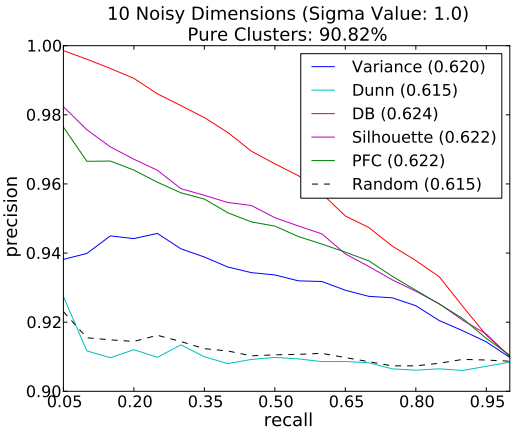
Table 7.3: Average distance between cluster centroids for experiments that add irrelevant feature dimensions per class.

Number Noise Dimensions	Pure Clusters	Mixed Clusters
0	3.700	4.125
5	4.436	4.214
10	4.540	4.365
15	4.485	4.358
20	4.432	4.333
25	4.410	4.315

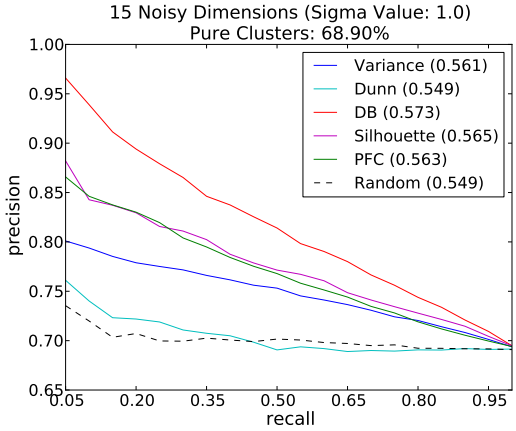
formance in terms of F-measure suggests that PFC is fairly consistent with a few irrelevant dimensions. For $\sigma = 1.0$ in the first synthetic data model PFC produced an F-measure of 0.628. This value increased to 0.638 in the second synthetic data model with less over-segmentation. In this third model, for 5 and 10 irrelevant dimensions PFC produce an F-measure of 0.633 and 0.622, respectively. Thus, with few irrelevant dimensions, PFC still yields high performance, but does not receive the performance boost that DB exhibits, causing the relative ordering to change. Once the number of irrelevant dimensions grows to 15, however, the performance difference between PFC and DB becomes more apparent, suggesting that DB is more robust to class-specific irrelevant features.



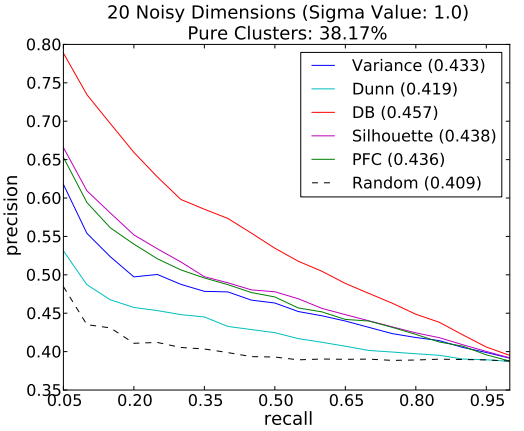
(a) Uniform Noise added to 5 Dimensions



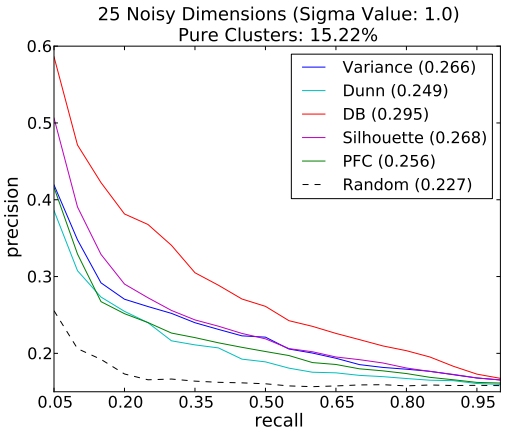
(b) Uniform Noise added to 10 Dimensions



(c) Uniform Noise added to 15 Dimensions



(d) Uniform Noise added to 20 Dimensions



(e) Uniform Noise added to 25 Dimensions

Figure 7.4: Precision-recall curves for 50 dimensional spherical Gaussian samples with $\sigma = 1.0$ and uniform noise added to randomly selected dimensions per cluster.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The rapid growth of technology makes it possible to capture a high volume of visual data with very little effort. This has helped large image and video corpora emerge for computer vision related tasks. However, large data sets require a significant amount of human labor to annotate with ground truth labels. Clustering the data can help alleviate the human workload as it allows for a group of samples to be labeled simultaneously instead of having to label individual instances. However, we discussed several visual data properties that make clustering image data by object class a challenging task.

The different viewpoints of a single object under fixed illumination form a closed 2D manifold in image space, and affine-invariant features divide the manifold into a set of points in feature space. Also, changes in illumination expand each viewpoint into a 9-dimensional subspace of image space. Further complicating the issue, some features describing object appearance (e.g., color and shape) are irrelevant for some classes, and some classes exhibit more background clutter than others. As a result, one does not expect that forcing a one-to-one correspondence between clusters and object classes will be very effective. Instead, many small clusters approximate a single object class.

Still, many resulting clusters are not representative of a single object class, but instead are random collections of images. This thesis evaluated five cluster quality measures on real-world and synthetic data sets to determine how well they could differentiate pure clusters from a mixed group of images. These measures included variance, three data mining measures, Dunn Index, Davies-Bouldin (DB) Index and Silhouette Width, and our novel cluster quality measure, Proximity Forest Connectivity (PFC). These five measures can largely be split into two distinct approaches. The first approach measures cluster quality as a combi-

nation of intra-cluster compactness and inter-cluster separation, which is used by all three data mining measures. The second approach is to rely solely on intra-cluster information, be that compactness or the relative locality of a cluster's samples, which are the approaches taken by variance and PFC, respectively.

The set of real-world experiments was designed to determine if a single cluster quality measure was able to consistently outperform the other measures in their task of identifying pure clusters from a set of partitioned data. However, in 16 real-world experiments, only variance and Silhouette failed to perform best on at least one data set according to the F-measure values. Although there is not a perfect consensus across all data sets to indicate a "best" performing cluster quality measure, PFC claimed half of the statistically significant top performances.

Using synthetic data models, we isolated certain properties often seen within visual data to determine their affect on the cluster quality measures. Most notably, we found that over-segmentation plays a significant role in the performance of all the cluster quality measures, particularly the data mining measures. Over-segmented data caused worse than random performance from the data mining measures, whereas with less over-segmentation, each of the data mining measures easily outperformed a random selection criteria. This seems to suggest that over-segmentation should be limited when possible. However, results also indicated that for highly overlapping data classes, an over-segmentation of the data is necessary to learn a set of pure clusters. Since the percentage of pure clusters was found to be highly correlated to the performance of most of the cluster quality measures in the real-world experiments, we conjecture that over-segmentation of visual data is necessary.

The synthetic model of feature relevance indicated that all the measures, except Dunn, were fairly robust to small amounts of class-specific noise. However, larger amounts of class specific noise proved to be more difficult for all the measures except DB, which is inferred because of the large performance gap seen between DB and the remaining measures. The addition of noise seemed to help create further centroid separation between pure clusters and their neighbors, which likely helped cause the performance boost of DB.

Given the real-world and synthetic data experiment results, we can say that our novel PFC measure is highly competitive with the existing cluster quality measures that have been evaluated in this thesis. A distinct advantage that PFC has over the data mining measures is the ability to be relatively unaffected by the properties of visual data that suggest an over-segmented, many-to-one mapping between clusters and object classes is essential to learn pure clusters. PFC's performance suggests that it could further improve the labeling workload by identifying only the pure clusters in the partitioned data set, and thus, only ask a human annotator to provide labels for these meaningful clusters.

8.2 Future Work

The evaluation presented in this thesis focused on how cluster quality measures can be used to identify clusters of images that represent a single object class. However, we hypothesize that these measures generalize to the evaluation of clusters of short video segments that represent a single action class. Evaluation on this secondary type of visual data would be useful in the long term to help label training data in the context of action recognition.

We have shown through synthetic data that cluster quality measures perform differently on different structures of data. The relative ordering of the cluster quality measures on the synthetic data, however, does not directly match the relative ordering of the measures on the CalTech256 data sets. Thus, we cannot yet say what the underlying structure of the CalTech256 data set is using the modeled synthetic data results. We assume that the structure is some combination of the structures we have looked at, along with other structures that were not directly tested in this thesis. In our future work we would like to draw better connections between the structures of the real-world and synthetic data.

Finally, we plan on merging these cluster quality measures with our labeling system, to more directly determine their effectiveness with respect to our underlying motivation of reducing the human labeling workload. This work has shown that existing cluster quality measures and our novel PFC measure can help distinguish between pure clusters and clusters

composed of a random collection of images. The next step is to directly test how much of an impact these measures provide during labeling.

References

[1] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

[2] Robert Benavente, Maria Vanrell, and Ramon Baldrich. Parametric fuzzy sets for automatic color naming. *Journal of the Optical Society of America A*, 25(10):2582–2593, 2008.

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 282–295, 2010.

[4] B. Burns, R. Weiss, and E. Riseman. The non-existence of general-case view-invariants. *Geometric invariance in computer vision*, 1:554–559, 1992.

[5] Dengxin Dai, Mukta Prasad, Christian Leistner, and Luc Van Gool. Ensemble partitioning for unsupervised image categorization. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 483–496, 2012.

[6] D.L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):224–227, 1979.

[7] J.C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[8] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[9] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering validity checking methods: part ii. *ACM Sigmod Record*, 31(3):19–27, 2002.

[10] A. Holub, P. Perona, and M.C. Burl. Entropy-based active learning for object recognition. In *Proceedings of Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1–8, 2008.

[11] A.J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 2372–2379, 2009.

[12] Y.J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1721–1728, 2011.

[13] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. pages 3–12. Springer-Verlag, 1994.

[14] D. Liu and T. Chen. Unsupervised image categorization and object localization using topic models and correspondences between images. In *Proceedings of the 11th International Conference on Computer Vision (ICCV)*, pages 1–7, 2007.

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[16] Y. Lui, R. Beveridge, and M. Kirby. Action classification on product manifolds. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2010.

[17] Q. Mo and B. Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. In *European Conference on Computer Vision (ECCV)*, 2012.

[18] H. Murase and S. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[19] Shree K Nayar, Sammeer A Nene, and Hiroshi Murase. Columbia object image library (coil 100). Technical report, Tech. Report No. CUCS-006-96. Department of Comp. Science, Columbia University, 1996.

[20] S. O'Hara and B.A. Draper. Scalable action recognition with a subspace forest. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1210–1217, 2012.

[21] S. O'Hara and B.A. Draper. Are you using the right nearest neighbor algorithm? In *Proceedings of Workshop on the Applications of Computer Vision (WACV)*, 2013.

[22] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[23] B.C. Russell, W.T. Freeman, A.A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1605–1614, 2006.

[24] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[25] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 193–199, 1999.

[26] T. Tuytelaars, C.H. Lampert, M.B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302, 2010.

[27] Jeffrey K. Uhlmann. Satisfying general proximity / similarity queries with metric trees. *Information Processing Letters*, 40(4):175–179, November 1991.

[28] L. Vendramin, R.J.G.B. Campello, and E.R. Hruschka. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4):209–235, 2010.

[29] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 2262–2269, 2009.

[30] C. Vondrick and D. Ramanan. Video annotation and tracking with active learning. In *Neural Information Processing Systems (NIPS)*, 2011.

[31] R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *Proceedings of the 9th International Conference on Computer Vision (ICCV)*, pages 516–523, 2003.

Appendix A

Nearest Neighbor Verification

This appendix provides experimental evidence that our modifications of Dunn Index and Silhouette Width, which includes using only the nearest neighbor of a cluster to calculate the measures, produces similar results to performing an exhaustive search across all clusters to find the minimum score. The experiments were done using the same subsets of the CalTech256 data set that were seen in Chapter 6. Tables A.1 and A.2 present the results when evaluating the nearest neighbor for Dunn and Silhouette, respectively.

Each table shows the percentage of times in which the nearest neighbor of a cluster produces the minimum score that would have been found using an exhaustive search. Since the nearest neighbor does not always return the minimum score, the tables also show where the nearest neighbor score ranks relative to the minimum score. Finally, the table shows the absolute difference between the minimum score and the values produced by the nearest neighbor, and the clusters producing the median and maximum scores during the exhaustive search. The reported ranking and absolute score differences are averaged over all clusters in the data set.

Table A.1: Results describing how often the nearest neighbor of a cluster produces the minimum Dunn index score, and how similar the nearest neighbor score is to the minimum.

			Absolute Score Differences to Minimum		
Subset	Exact Matches (%)	Ranking	Nearest Neighbor	Median	Maximum
20 Easy	26.0	6.726	0.049	0.444	1.358
1-20	23.4	6.496	0.041	0.286	0.876
21-40	24.8	7.771	0.055	0.296	1.366
41-60	25.7	6.858	0.050	0.309	0.915
61-80	30.0	5.955	0.048	0.332	0.926
81-100	23.3	6.287	0.048	0.346	1.080
101-120	23.4	7.839	0.056	0.320	0.980
121-140	19.7	7.258	0.054	0.310	1.058
141-160	21.7	7.034	0.055	0.400	1.277
161-180	30.8	4.558	0.043	0.303	0.914
181-200	18.9	6.523	0.055	0.321	0.911
201-220	16.3	6.096	0.055	0.290	0.822
221-240	23.0	7.135	0.058	0.322	1.287
241-256	23.7	5.333	0.051	0.436	1.090
Average	23.6	6.562	0.051	0.337	0.993

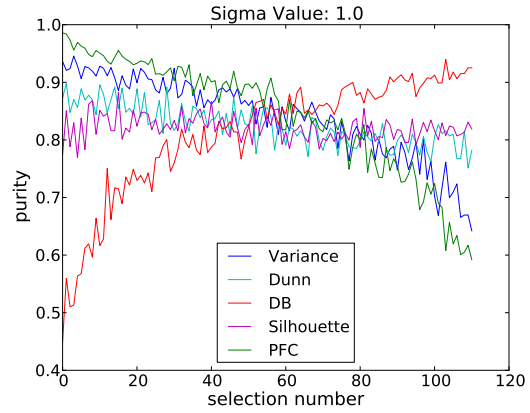
Table A.2: Results describing how often the nearest neighbor of a cluster produces the minimum Silhouette index score, and how similar the nearest neighbor score is to the minimum.

			Absolute Score Differences to Minimum		
Subset	Exact Matches (%)	Ranking	Nearest Neighbor	Median	Maximum
20 Easy	37.4	5.639	0.024	0.210	0.409
1-20	42.6	4.454	0.024	0.207	0.444
21-40	41.9	4.229	0.026	0.219	0.504
41-60	43.4	4.142	0.023	0.202	0.412
61-80	45.5	4.009	0.026	0.217	0.396
81-100	40.3	4.783	0.027	0.231	0.449
101-120	41.9	4.460	0.029	0.221	0.437
121-140	34.8	6.106	0.035	0.216	0.467
141-160	36.0	6.240	0.027	0.227	0.499
161-180	52.9	4.337	0.023	0.219	0.440
181-200	42.3	4.306	0.029	0.227	0.487
201-220	48.1	2.923	0.018	0.195	0.381
221-240	39.7	5.095	0.031	0.219	0.472
241-256	44.9	4.026	0.018	0.223	0.471
Average	42.3	4.625	0.026	0.217	0.448

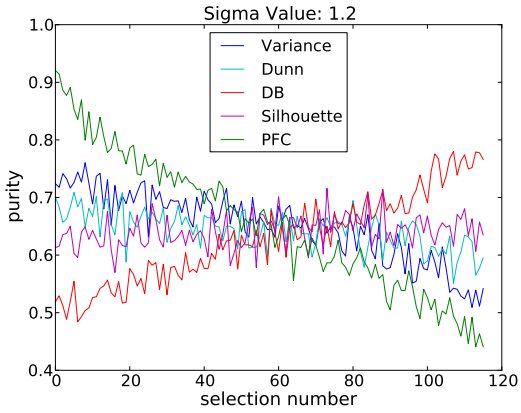
Appendix B

Purity Per Selection Plots

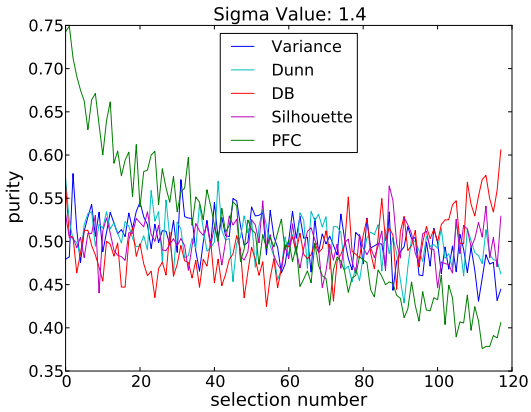
This appendix provides secondary plots of the results presented in this thesis without the use of a hard threshold to define what it means to be a pure cluster. This secondary plot creates a curve showing the true purity of clusters in the same order in which each measure suggests clusters should be labeled. The trends in the figures match the trends show in the precision-recall plots, although the figures in this appendix are much noisier.



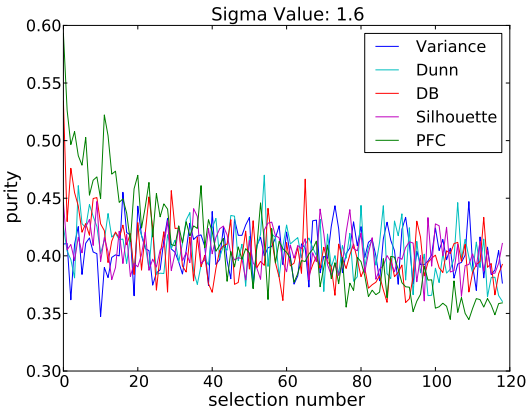
(a) $\sigma = 1.0$



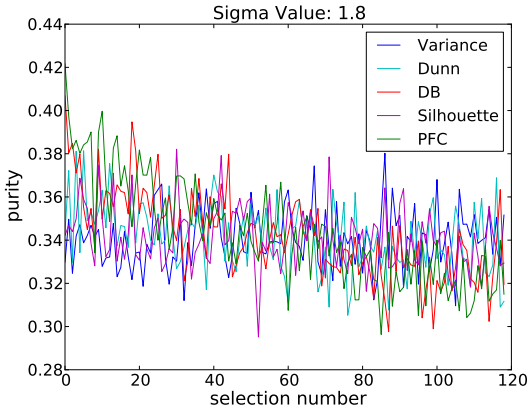
(b) $\sigma = 1.2$



(c) $\sigma = 1.4$

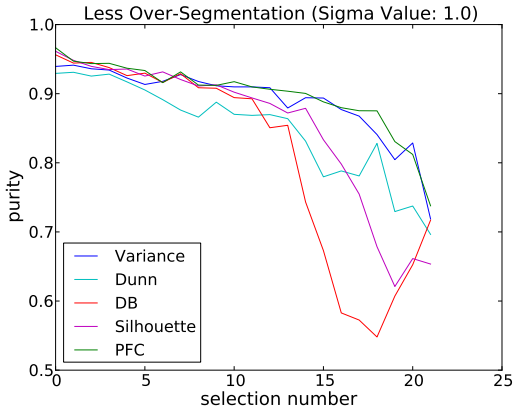


(d) $\sigma = 1.6$

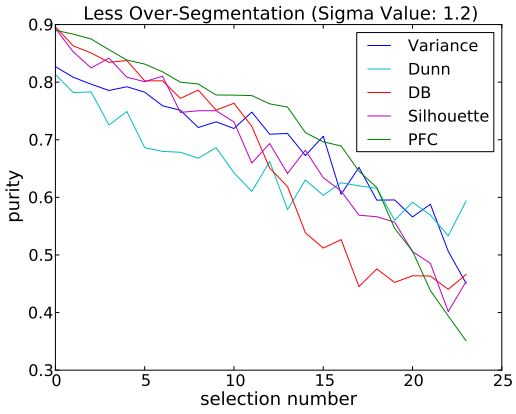


(e) $\sigma = 1.8$

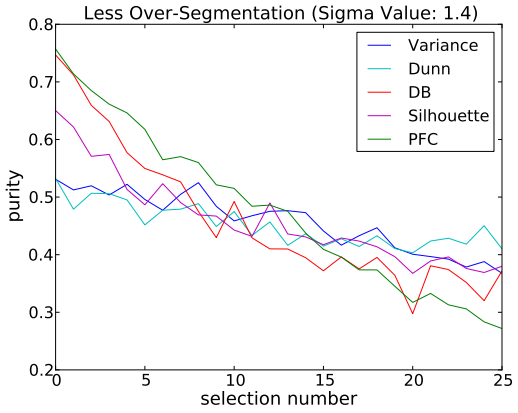
Figure B.1: True purity values for the ordered selection of clusters generated from 50-dimensional spherical Gaussian processes.



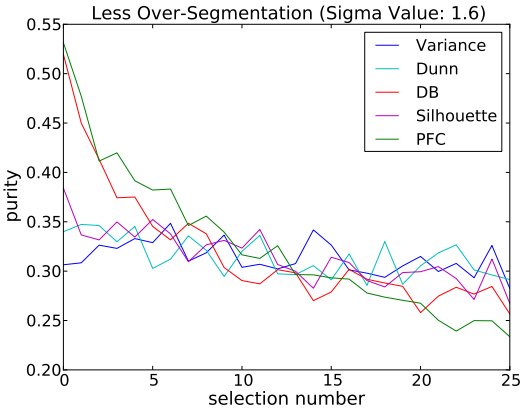
(a) $\sigma = 1.0$



(b) $\sigma = 1.2$

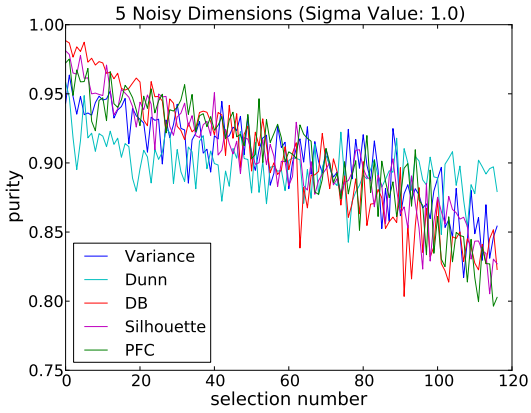


(c) $\sigma = 1.4$

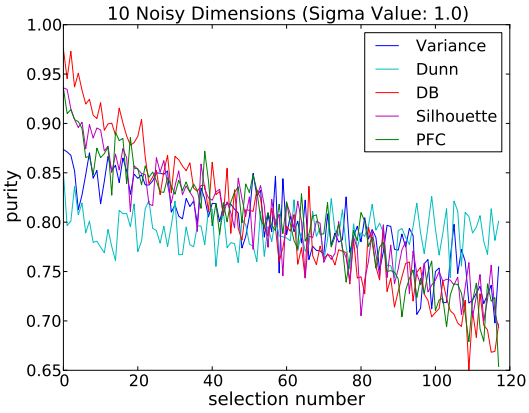


(d) $\sigma = 1.6$

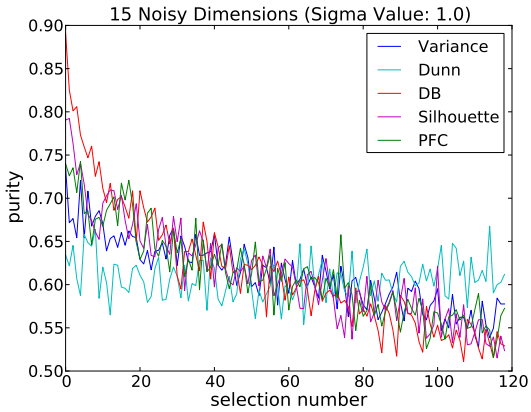
Figure B.2: True purity values for the ordered selection of clusters generated from 50-dimensional spherical Gaussian processes using less over-segmentation.



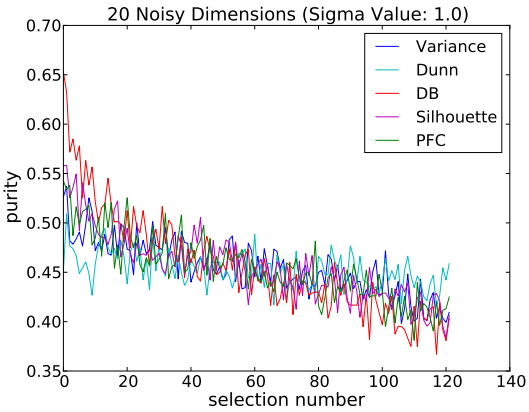
(a) $\sigma = 1.0$



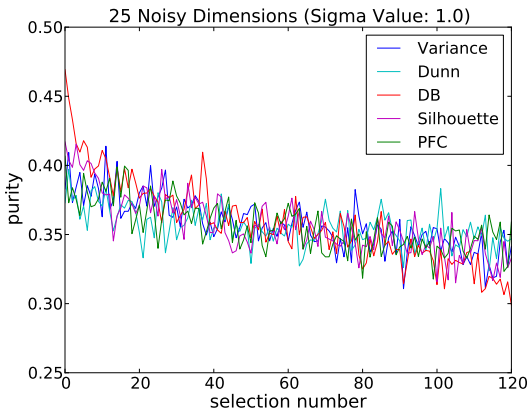
(b) $\sigma = 1.2$



(c) $\sigma = 1.4$



(d) $\sigma = 1.6$



(e) $\sigma = 1.8$

Figure B.3: True purity values for the ordered selection of clusters generated from 50-dimensional spherical Gaussian processes with noise dimensions.