Testing and Tuning SkinnyDip: Noise-Robust Clustering

Grant King
kinggra1@msu.edu

1. PROBLEM DESCRIPTION

It is not uncommon for some real-world data sets to feature an abundance of noise. Depending on the severity of this noise, classical clustering methods may fail due to due to dependence on clean data or confusion from increasingly excessive noise. SkinnyDip is an algorithm proposed by Maurus et al.[1] designed to handle clustering data in highly noisy environments. SkinnyDip is based on the statistical concept of the *dip* [2].

The dip views the structure of the Empirical Cumulative Distribution Function(ECDF) of a set of single dimensional data to determine whether it is unimodal or multimodal. This test is expanded to a multidimensional, recursive heuristic in order to isolate the various modes of each feature of a set of data. This results in a deterministic, parameter-free, unsupervised method of finding clusters that are based on the modes of multivariate distributions.

My goal is to augment the SkinnyDip algorithm with the addition of a Gaussian clustering model in an attempt to reduce the excessive inclusion of noise in hypercubic-bound clusters, and perform additional tests to demonstrate its usefulness on noisy, real-world data sets.

2. INTRODUCTION

Data clustering is a fundamental problem in Machine Learning, and has been approached in a variety of ways, resulting in a plethora of tools and methods [3], many of which are sensitive to noise and other outliers. Additionally, many common techniques, such as k-means clustering, operate as closed set clustering methods and are unable to reject noise at all.

[4] is a density-based technique for finding clusters in environments that may contain noise. However, it has the disadvantages of being a parameterized method, and still continues to find extraneous clusters in increasingly noisy data when compared to SkinnyDip.

A single other existing method for clustering using the statistical dip test was found, a technique called DipMeans[5]. This method takes a different approach to using the dip test by performing it on a collection of distance measurements as opposed to the raw data values themselves. SkinnyDip, however, requires no distance measurements and is both functionally and computationally distinct from the DipMeans technique.

While SkinnyDip does manage to find all values within a cluster with high accuracy and precision, there is still a notable risk of falsely matching noise to a particular class. This is inherent in the way that the algorithm segments the clusters into hypercubic regions. If the data does not fit a cubic model, then the values included in extreme regions of the cluster (e.g. the corners of a square cluster) are likely to be false positives and should not have been included. This discrepancy increases exponentially as the dimension of the data increases, an issue acknowledged in the original paper[1]. I plan to demonstrate that the addition of a second step to clustering can reduce this particular error rate, with minimal negative impact on the existing true positive matches.

3. DATA

A visual example of the purpose of SkinnyDip is demonstrated in the running example data from [1] (See Figure 1) which shows the extraction of distinct shapes from a two-dimensional data field that consists of 80% static noise. A variety of similar clustering methods were shown to perform poorly on this data set, both through the inclusion of the evenly-distributed, static noise in clusters, and through excessive segmentation of the actual classes.

Aside from the uniform background noise, the running example data consists of 2 distinct cluster models: two rectangular model clusters and four 2D Gaussian model clusters. In Figure 1 the square classes are on the left side of the image and are colored red and black, while the remaining four, smaller clusters are the 2D Gaussians. In the process of generating this data, 200 points are generated using their respective distributions (uniform or normal), and placed alongside uniform noise covering the entire field. Any noise that falls within the bounds of these distributions is recategorized as being part of that class.

Future synthetic data that I will generate will have clusters that are modeled by multi-dimensional normal distributions, to try to maximize these false positive classification errors in the original algorithm. Any synthetic or real data that will be analyzed using both the original and my augmented version of SkinnyDip must have uniform background noise, or there will be clusters falsely detected by variations in the distribution of the background.

4. PROGRESS

An updated project timeline can be seen in Figure 3

4.1 Completed

To confirm my suspicion of the high rate of false positive detection in the non-rectangular clusters, I developed some numerical and visual testing tools to

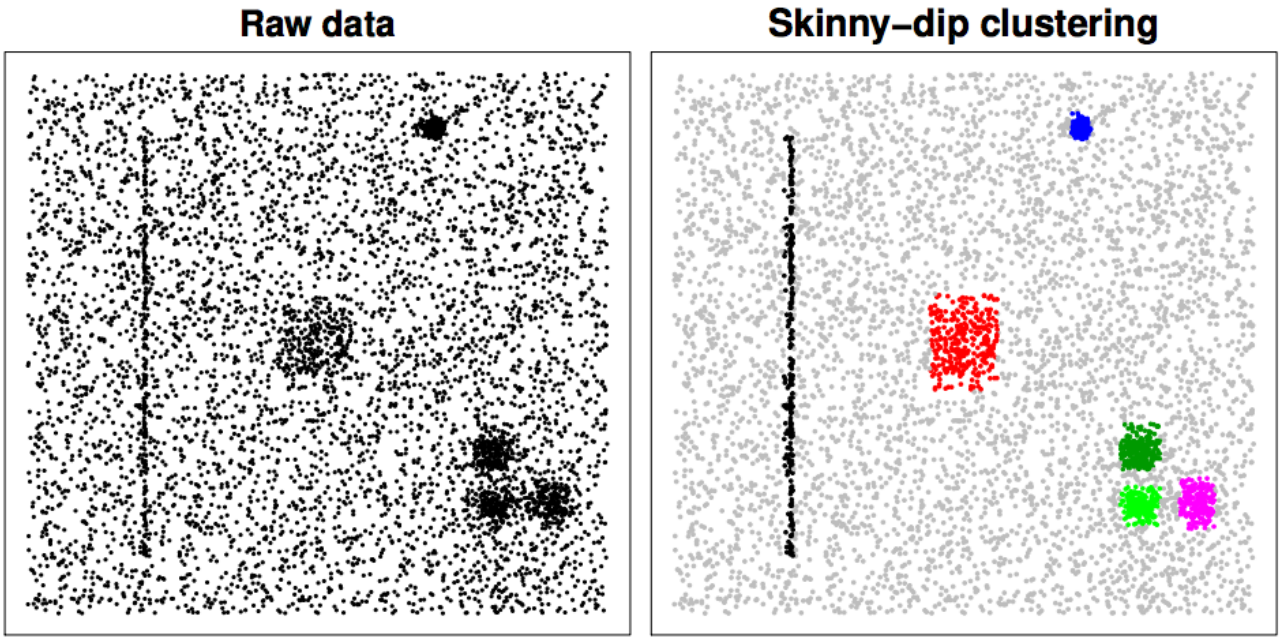

Figure 1: The running example used by Maurus, et al. throughout [1]

In my analysis of this synthetic data, it is clear that Gaussian models composed of a large number of false positives, even in just this 2D case. An image analyzing the the classification accuracy of this particular data set is seen in Figure 2. In this image, the locations of the different clusters are the same, but the coloration has changed slightly, the two leftmost shapes (black and red) are still the rectangularly modeled clusters, and the four, smaller shapes are the 2D Gaussian clusters.

The classes with rectangular models were the only classes that had a false positive rate below 10%, while the Gaussian modeled classes had false positive ranges that ranged from 10.2% up to 21.4%. This supports my hypothesis that the Gaussian modeled clusters would significantly over-sample from their respective classes.

4.2 Remaining

Now that I have a tool set to test the accuracy of each cluster's classifications, I plan to expand this testing by adding further dimensions to the existing synthetic data in order to visualize the change in the false positive rate relative to dimension for Gaussian model clusters. This will also be useful for showing the difference in both true positive and false positive changes as I implement a supporting clustering algorithm. For implementation of a second algorithm, I plan to take the subsets of data that are sectioned off by the hypercubic bounds of SkinnyDip and perform additional clustering. My initial approach will be a simple parameter estimation for the multi-dimensional Gaussian case to see what effect this has on the true and false positive detection as the dimension of the data increases. Beyond this, I want to apply this clustering to some of the additional data used in the original SkinnyDip experiments, such as clustering of road segments, to see if this addition will further decrease the noise captured in each of those clusters. As a stretch goal, I would be interested in seeing how this algorithm would apply to star mapping data, such as for the purpose of galaxy clustering.

5. REFERENCES

[1] Samuel Maurus and Claudia Plant. Skinny-dip: Clustering in a sea of noise. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1055–1064, 2016.

[2] J. A. Hartigan and P. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 1985.

[3] Rui Xu and D. Wunsch, II. Survey of clustering algorithms. *Trans. Neur. Netw.*, 16(3):645–678, May 2005.

[4] Martin Ester, Hans peter Kriegel, JÃűrg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231, 1996.

[5] A. Kalogeratos and A. Likas. Dip-means: an incremental clustering method for estimating the number of clusters. *In Advances in neural information processing systems, pages 2393âĂŞ2401*, 2012.

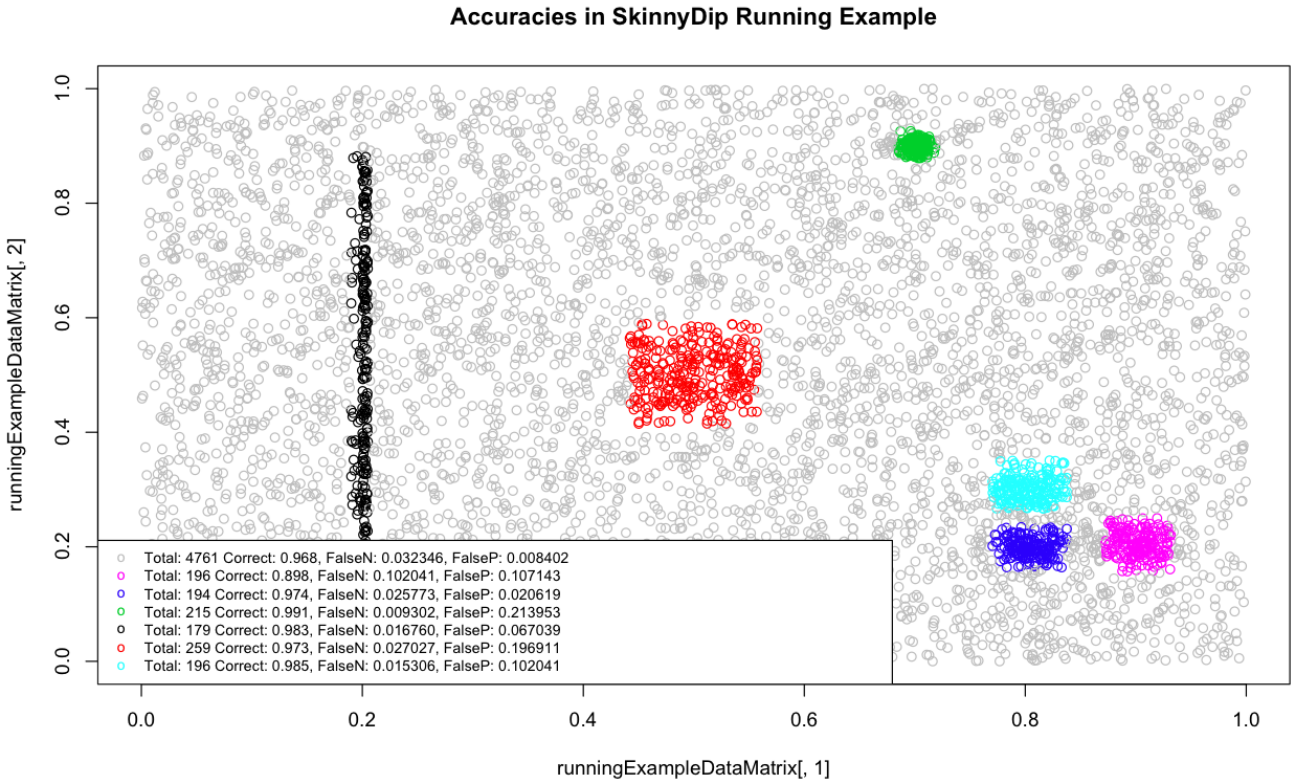

Figure 2: Accuracies of the given SkinnyDip example. NOTE: introduced a bug while updating code that caused legend labels to be shuffled incorrectly, reported data does not correctly match class colors. This will be corrected and updated to a different folder after report has been completed to maintain a "before midnight" commit timestamp.

2/24/2017	Finalize Project Goals
3/3/2017	Setup Computing Environment for R libraries
3/10/2017	Get SkinnyDip library Working and Begin Learning R
3/17/2017	Begin Expanding Library with Accuracy Testing
3/24/2017	Intermediate Report Due
3/31/2017	Try Fine Tuning Clustering Results (e.g. Density Based Methods)
4/7/2017	Test Higher Dimensionalities
4/14/2017	Look for/Test Real-World Noisy Data
4/21/2017	Finalize Report
4/28/2017	Final Report Due

Figure 3: Progress on Project Milestones