2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Core Support Extraction for Learning from Initially Labeled Nonstationary Environments using COMPOSE

Robert Capo, Anthony Sanchez, and Dr. Robi Polikar

***Abstract*—Learning in nonstationary environments, also called concept drift, requires an algorithm to track and learn from streaming data, drawn from a nonstationary (drifting) distribution. When data arrive continuously, a concept drift algorithm is required to maintain an up-to-date hypothesis that evolves with the changing environment. A more difficult problem that has received less attention, however, is learning from so-called initially labeled nonstationary environments, where the the environment provides only unlabeled data after initialization. Since the labels to such data never become available, learning in such a setting is also referred to as extreme verification latency, where the algorithm must only use unlabeled data to keep the hypothesis current. In this contribution, we analyze COMPOSE, a framework recently proposed for learning in such environments. One of the central processes of COMPOSE is core support extraction, where the algorithm predicts which data instances will be useful and relevant for classification in future time steps. We compare two different options, namely Gaussian mixture model based maximum a posteriori sampling and α-shape compaction, for core support extraction, and analyze their effects on both accuracy and computational complexity of the algorithm. Our findings point to—as is the case in most engineering problems—a trade-off: that α-shapes are more versatile in most situations, but they are far more computationally complex, especially as the dimensionality of the dataset increases. Our proposed GMM procedure allows COMPOSE to operate on datasets of substantially larger dimensionality without affecting its classification performance.**

I. Introduction

A nonstationary environment—in the context of machine learning—refers to an environment generating (typically) streaming data whose underlying distribution changes with time. This means that the joint probability distribution of the unlabeled data X and corresponding grouping variables (i.e., labels) Y changes at each time step, such that

$$p_t(X, Y) \neq p_{t+1}(X, Y)$$

where X is an N x d observation matrix with N observations in d dimensions, Y is the corresponding length N vector of labels, and t is the current time step. In this contribution, we focus on nonstationarity caused by drifting distributions [1] rather than a stationary feature distribution with only a change in the labeling function, $p(Y|X)$. Furthermore, we assume this drift to be limited or gradual by nature, as opposed to an abrupt concept change.

Robert Capo, Anthony Sanchez, and Dr. Robi Polikar are with the Department of Electrical and Computer Engineering at Rowan University, Glassboro, NJ, USA. (emails: robcapo@gmail.com sanche08@students.rowan.edu, polikar@rowan.edu)

This work is supported by the NSF under Grant No: ECCS-1310496.

The problem of learning concept drift becomes even more difficult if labeled data are only available initially (i.e., at $t = 0$), beyond which only unlabeled data are available at all future time steps. Obviously, the relevance of the initially received data and their labels degrades as the data distributions drift further from their initial positions. In our previous efforts [2], [3] we introduced a framework, called COMPOSE, for learning in precisely such an environment, whose data we refer to as "initially labeled nonstationary streaming data."

The COMPacted Object Sample Extraction (COMPOSE) algorithm is a wrapper approach to learning in initially labeled nonstationary environments. The framework itself makes only the gradual drift assumption about the nature of the change and makes no assumptions about the shape of the underlying distributions. COMPOSE receives a new batch of data at every time step, t, during which two distinct processes occur. The semi-supervised learning (SSL) process uses the currently available labeled data, $X_L(t)$, to assign labels to the currently unlabeled data, $X_U(t)$, and adds these observations to the labeled set. SSL is a mature topic in machine learning (see [4]–[6] for examples). We use the label propagation algorithm in our experiments, though COMPOSE is indepedent of the selection of the SSL algorithm, and can use any SSL algorithm that the user deems most appropriate for the problem. The second primary component, developed specifically for COMPOSE, is the core sample extraction (CSE) process, which prepares COMPOSE for future time steps by replacing X_L with $X_L^* \in X_L$, the set of labeled observations that will be most useful in classifying the future unlabeled data. An optional active learning process can be added as well as described in [7]. Doing so removes the constraint of limited drift for the ability to request the labels of a small number of carefully selected instances at any given time. Each of these processes is implemented modularly, such that different modular algorithms can be switched in and out.

The general process of COMPOSE can be seen in Figure 1, and our efforts for this contribution are focused around steps 4 and 5, specifically, on the analysis of the CSE process. The consequence of the gradual drift assumption on CSE is that the instances at the central core (typically high-density) regions of the currently available data will be the most useful ones in the future, as such instances are most likely to represent data generated during the next time step of a gradually drifting distribution. These instances will have the most overlap with future data regardless of the direction of drift, under the

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

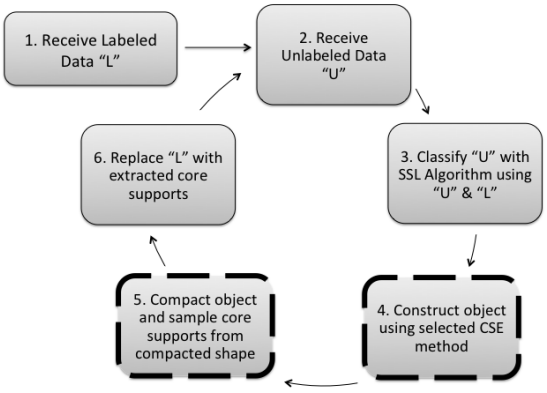

Fig. 1: Overview of COMPOSE. CSE steps outlined in bold dashed lines

gradual drift assumption. The goal of the CSE algorithm is to create an object or shape (preferably non-parametric) enveloping the data from each class, and extract (sample) new labeled data from compacted versions (i.e., areas of high density) of these objects. In this contribution we explore two methods to do so: i) α-shape compaction as used in the original formulation of COMPOSE [8], and ii) Gaussian mixture model (GMM) [9] maximum a posteriori sampling, which we propose in this effort.

This rest of this paper is organized as follows: Section II explains the theory behind two CSE methods and explores the strengths and weaknesses of each, Section III presents the results of our experiments, and Section IV explains the conclusions and path for future work based on our results.

II. Core Support Extraction Theory

A natural outcome of the gradual drift assumption is that class distributions overlap at subsequent time steps. In other words, as long as drift is limited, the center or core region of each class-conditional data distribution will have the most overlap with future data, regardless of drift direction or drift type. Figure 2 shows three different types of drift, rotational, translational, and volumetric, and shows that the compacted core region (outlined) has the most overlap with the drifted distribution (dashed line). The core support extraction (CSE) procedures we explore attempt to accurately identify which instances lie in the core region of the existing class distributions; these instances, which are previously labeled by the last SSL step, are then used as training data for the next iteration's SSL step in labeling the new unlabeled data. The input to the CSE process is p, the percentage of available observations to retain as training data for future time steps, and the output is a set of indices describing which instances are determined to be core supports. The two methods we compared for CSE are α-shape compaction and GMM maximum a posteriori sampling, which are described in Sections II-A and II-B respectively.

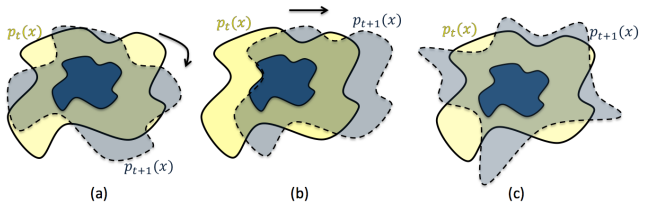

Fig. 2: Types of distribution drift: (a) rotational, (b) translational, (c) volumetric

A. Compacted α-shape Sampling

Compacted α-shape sampling is the original CSE approach that was developed for COMPOSE [2], [3]. An α-shape is a geometric representation of a dataset that can be described as a generlization of the dataset's convex hull. The convex hull of a dataset $X \in \mathbb{R}^d$ is the convex shape with minimum area that contains all of the observations in X, and can be described as the set of all possible convex combinations of the points in X, or

$$\left\{ \sum_{i=1}^{|X|} a_i x_i | (\forall i : a_i \geq 0) \wedge \sum_i a_i = 1 \right\}$$

for all possible a_i. The α-shape, however, contains a free parameter that allows for concavities in the shape. The free parameter, α, determines the level of detail of the shape. When $\alpha = \infty$, the α-shape is equivalent to the convex hull. As α decreases, concavities, holes, and disjoint shapes are created, depending on the data.

The α-shape creation first requires obtaining the Delaunay tessellation [10] of the data. The Delaunay tessellation is a set of adjacent d-simplices from the data forming a partitioned version of the convex hull. A d-simplex is the convex hull of $d+1$ observations; the observations are connected via faces, which are $d-1$-simplices. Therefore, a 2D simplex is the convex hull of 3 points (a triangle) connected via 1D simplices (lines), and a 3D simplex is the convex hull of 4 points, each connected via 2D simpexes (triangles), and so forth. The tessellation can be extended to any space of abritrary dimenison, but it is most easily explained in the 2D case (i.e., the Delaunay triangulation). There exist Delaunay conditions, which ensure that the triangulation is the unique solution that maximizes the minimum angle of any of these triangles, such that every observation is assigned to one or more triangles, and the entire set of triangles form the convex hull.

Each triangle has a circle that circumscribes it (i.e., all 3 vertices fall on the edge of the circle). If the Delaunay conditions are met, no observation will fall inside any circumcircle besides the three belonging to the triangle it circumscribes. A valid Delaunay tessellation exists for any dataset of abritrary dimension and arbitrary size as long as two non-limiting and non-restricting conditions are satisfied by the data: no observation in the dataset may be repeated exactly, and all of the data must not fall on the same $(d-1)$-hyperplane. The α-shape simply finds the Delaunay tesselation of the dataset and sets a threshold on the maximum radius

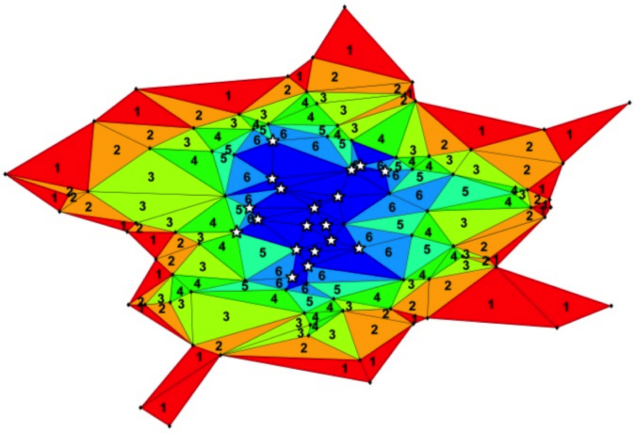

Fig. 3: Layers of an α-shape constructed from the Delaunay triangulation. Observations marked with stars indicate core supports.

of the circumsphere of any simplex belonging to the shape. Simplices whose circumsphere is too large are removed from the α-shape. The remaining simplices and their corresponding observations make up the final α-shape.

The algorithm used for the Delaunay tessellation is the Quickhull algorithm. This algorithm is of order $O(n^{(d+1)/2})$ where n is the number of observations and d is the dimensionality of the data. Hence, the algorithm is exponential in dimensionality, which makes it very expensive for large dimensional datasets.

In order to obtain the core support region, the α-shape is compacted (shrunk) by iteratively stripping away its outermost layer of simplices until the desired number of observations remain. The outer most layer is defined as the set of simplices that have one or more face not shared by any other simplex in the shape. After the first outer layer is found, those simplices are removed from the shape, and the new outer layer is found. The process continues until $p' \geq p$ percent of the original observations remain from each class. As many layers as possible are removed such that at least p percent of the observations remain. The CSE procedure then returns the indices of the remianing observations as core supports. An example of a layered α-shape and its Delaunay triangulation is shown in Figure 3 [2].

B. GMM Maximum A Posteriori Sampling

The Gaussian mixture model (GMM) is a probalistic model that describes the data as a mixture of unimodal Gaussian distributions. The GMM algorithm tries to fit K Gaussians to the data X, where K is a free parameter specified by the user. The probability density function of a GMM is the weighted sum of the K Gaussians as given by the equation,

$$p(\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k) \tag{1}$$

where

$$\theta = \{\theta_k \forall k\} = \{\mu_k, \Sigma_k, \pi_k \forall k\}$$

and μ_k, Σ_k and π_k are the mean, covariance and mixing coefficient (i.e., prior probability) of each Gaussian component, θ_k is the set of parameters describing the kth component, and θ is the set of parameters describing the entire model. The GMM algorithm randomly initializes K Gaussians in the feature space, and uses the expectation maximization (EM) algorithm to fit the Gaussians to the data with maximum likelihood. The EM procedure is an iterative two step procedure that runs until convergence (or a maximum number of iterations is reached). In the expectation step, the probability that each component, θ_k, can be explained by observation x_i is calculated. This is often called the membership weight and is defined as

$$p(\theta_k|x_i) = \frac{\pi_k p(x_i|\mu_k, \Sigma_k)}{\sum_j \pi_j p(x_i|\mu_j, \Sigma_j)} \tag{2}$$

for all k, i. The maximization step then calculates new parameters for each component to maximize the likelihood that the mixture model represents the data based on the new membership weights.

$$\mu_k = \frac{\sum_i p(\theta_k|x_i) x_i}{\sum_i p(\theta_k|x_i)} \tag{3}$$

$$\Sigma_k = \frac{\sum_i p(\theta_k|x_i)(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i p(\theta_k|x_i)} \tag{4}$$

$$\pi_k = \frac{\sum_i p(\theta_k|x_i)}{N} \tag{5}$$

The EM procedure is not guaranteed to find the model with global maximum likelihood (even if correct K is chosen), but local maxima found by the algorithm are often sufficient, as has been the case in our experiments. In addition, in real world scenarios, we rarely know the true value of K. The common solution to the optimal choice of K is to try a range of K values and choose the one that minimizes some penalty or cost function. In our implementation, we have used the Bayes Information Criterion (BIC) [11]. The BIC adds a penalty for large K to the negative log likelihood in order to prevent overfitting, and is given by

$$BIC = -2 \ln L + K \ln N \tag{6}$$

Once the best model is chosen, core supports are extracted by calculating the Mahalanobis distance [12] for each x_i to each component in the GMM. The minimum distance to any component is calculated as

$$d_{min}(x_i) = \min_k \sqrt{(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)} \tag{7}$$

The p observations with the smallest d_{min} are kept as core supports.

One important assumption GMMs make is that the underlying distribution is actually a mixture of Gaussians. Of course, in real world situations, this may not be the case, however, arbitrary shaped distributions can be approximated with GMMs provided that they are well represented by the data and K is chosen sufficently large.

C. Comparison of CSE Methods

Both α-shapes and GMMs have one free parameter, the resolution parameter α and the number of Gaussian components, K, respectively. One important consideration for choosing a CSE method is therefore the parameter that is easiest to estimate based on the data, and causes the least sensitivity to suboptimal choices. α-shapes require knowledge of the resolution of data. This parameter can be difficult to guess and may be mismatched among different features of the data. A potential solution is to normalize the features by a certain factor and experiment with different values of α on the training data. GMMs require correct K to be in the range of values that are searched. Our experiments have shown that it is better to pick K too large than too small. GMMs are resistant to overfitting as long as $K \ll N$, which is typically the case. One benefit of the K parameter is that it is a discrete number, whereas the range of possible α values is continuous with infinitely many possibilities.

GMMs are also significantly more computationally efficient than α-shapes, particularly when d is large. The computational complexity of the EM procedure for GMMs is difficult to quantify, because it is an iterative procedure, but it has been shown that the E-step and the M-step are order $O(NKd + NK)$ and $O(2NKd)$, respectively, for each iteration, where N is the number of observations, K is the number of mixture components and d is the dimensionality [13]. Our results in Section III confirm that that the GMM approach is indeed substantially faster than constructing α-shapes for any given dimensionality and data cardinality. The computational advantage allows for many more values of the parameter to be tested in a given time period.

One major advantage of the α-shape algorithm is that it relies only on the closeness assumption (i.e., highly related data reside close to each other in the feature space). GMMs, however, make implicit assumptions that the data represents a mixture of Gaussians. As we explained in Section II-B, GMMs can successfully extract meaningful core supports from arbitrary distributions as long as they are well represented and K is chosen sufficiently large. The relative computational efficiency of the EM procedure allows us to test many values of K and find the one that optimizes the BIC. Doing so allows us to relax the assumption that the data must strictly follow a mixture of Gaussians. Figure 4 shows a comparison of both CSE methods on arbitrarily shaped datasets. The α parameter was set to different values depending on the dataset, and the K parameter was the value that minimized the BIC in the range of 1 - 20. The figure clearly shows that α-shapes sample core supports that more accurately represent the overall distrubtion than GMMs. However, the accuracy of α-shapes is very sensitive to the proper selection of $alpha$, and the optimal values of α were chosen for each dataset. The GMMs, however, used the same range for all three datasets and chose the optimal K automatically. In order for α-shapes to properly extract core supports, the user must have extensive prior knowledge about the dataset.

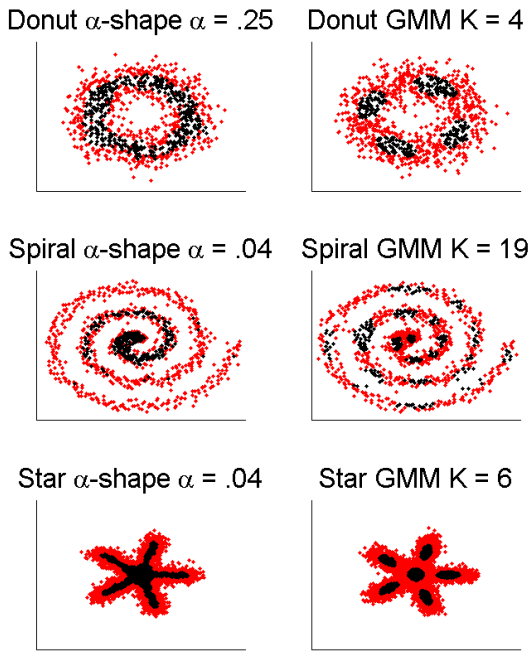

Fig. 4: Core supports (black) extracted by α-shape compaction and GMM based maximum a posteriori sampling on arbitrarily shaped datasets (original data in red)

III. Results

When extracting core supports, our general assumption is that the most informative data lie in the central core region or the densest region of the distribution. This is the typically the case in most situations, however it is not uncommon for a distribution to have two modes, one being less dense than the other, as shown in Figure 5. The information at the core region of both modes is important to keep for the next time step, as they both represent the underlying concepts generating data.

Our first experiment tested the ability of both α-shape's and GMM's in extracting core supports from both modes of an imbalanced distribution. Figure 5 shows the two distributions, the underrepresented D_1 (bottom) and D_2 (top), which was well represented throughout the experiment. The number of observations generated by D_2 remained constant at 10,000, while the number of observations generated by D_1 iterated from 100 to 10,000. The goal was to find the percentage of observations, p_1, extracted from D_1 as the cardinality, $|D_1|$, increased.

Figure 6 shows the ratio of the number of observations extracted from D_1 to the number of observations extracted from D_2, p_1/p_2, on the y-axis and the ratio of observations generated by D_1 to the number of observations generated by D_2, $|D_1|/|D_2|$ on the x-axis. Ideally, the CSE methods should extract the same percentage of observations from each mode, so the optimal value of p_1/p_2 is 1.

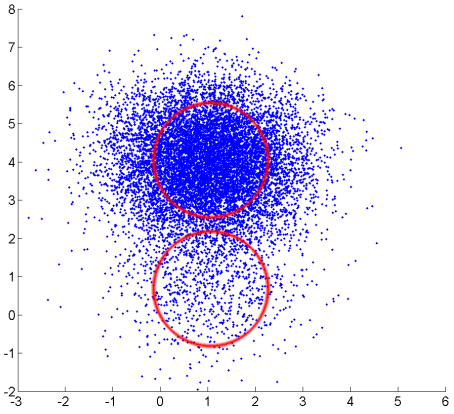

Fig. 5: Bimodal distribution with imbalanced densities

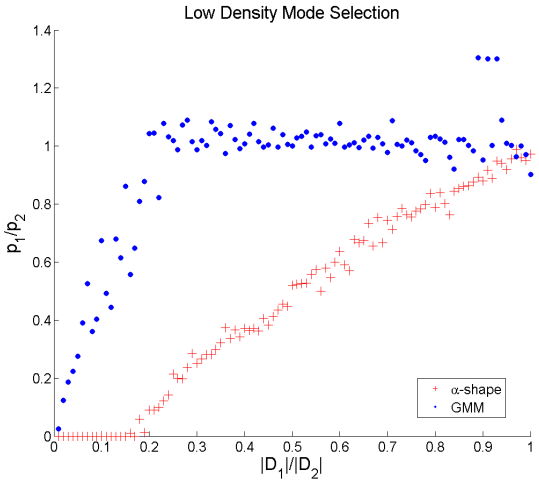

Fig. 6: Samples extracted from imbalanced bimodal distribution

Figure 6 shows that α-shapes generally fail to extract any core supports from D_1 until it is generating at least 25% of the data. Furthermore, p_1/p_2 doesn't approach the optimal value of 1 until D_1 is no longer underrepresented (i.e., $|D_1|/|D_2| = 1$). GMMs, however, extract core supports from D_1 almost immediately. The GMMs converge with minimal BIC when $K = 2$, and the observations that have small d_{min} (from Equation) to D_1 are selected as core supports. The outliers for GMMs are caused by improper convergence of the EM procedure, which is generally caused by poor initial conditions. These outliers indicate that GMMs are not as robust as α-shapes, and may cause unexpected results.

To determine the impact of the results of this experiment on the actual COMPOSE implementation, we developed a synthetic two class dataset. Each class contained a bimodal spiral distribution with imbalanced data as shown in Figure 7. We ran COMPOSE with a parameter sweep with values shown in Table I.

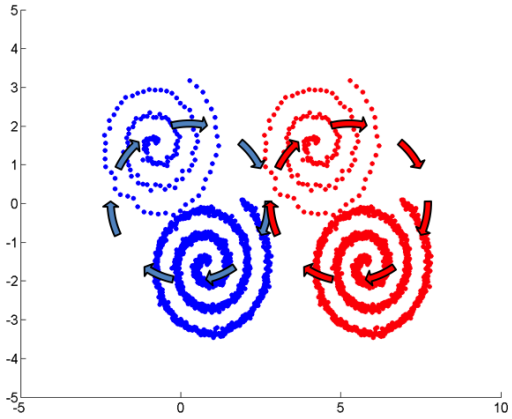

Fig. 7: Experimental setup of rotating spiral experiment

TABLE I: Parameter values for GMM and α-shapes

Method (Parameter)	Values
GMM (K)	{1, 2, 3, 5, 10, 15, 20}
α-shape (α)	{0.1, 0.2, 0.5, 1, 1.5, 2, 3}

During this experiment, we observed that the classification performance with α-shapes was very stable, while GMMs where less stable, as the performance relied heavily on the initial conditions at each time step. However, GMMs were able to track the underrepresented mode for longer and maintain higher performance as overlap increased (time steps 14, 43 and 73). Once the overlap was significant, both algorithms started misclassifying the underrepresented spiral, and their performances showed a significant drop, as seen in Figure 8. The shaded region in Figure 8 (and later in Figure 10) represent the 95% confidence interval obtained through 10 number of repetitions of random sampling from the given distributions..

In order to test each algorithm in a higher dimensional feature space, we developed a 4D experiment, shown—representatively—in Figure 9. The figure shows a 2D cross section of the experiment, as the data remained constant in the other two dimensions. This experiment contained three classes, two of which were unimodal, and one that was bimodal. Each mode of the three classes contained the same number of instances. The two unimodal classes followed the circular path indicated by the blue arrows. The two modes of the bimodal class continuously merged and separated as the other two classes rotated around them.

The results of the 4D experiment are shown in Figure 10, again where the shaded regions represent the 95% confidence intervals. Results are shown for α-shape, GMM, and the optimal Bayes classifier. The Bayes classifier was provided with the true means and covariances of each distribution at each time step, and used this information to output the statistically best classification. Figure 10 shows that COMPOSE performed remarkably close to the Bayes classifier using both

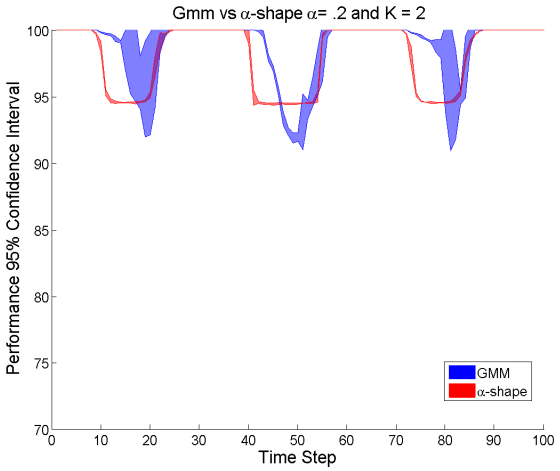

Fig. 8: Comparison of best performing α-shape (α = 1) and best performing GMM $K = 2$

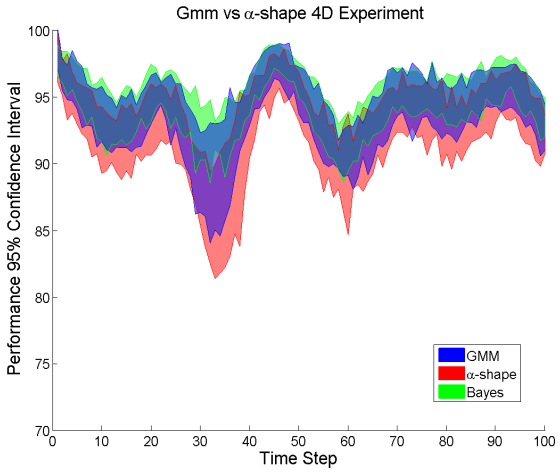

Fig. 10: Performance results of 4D experiment

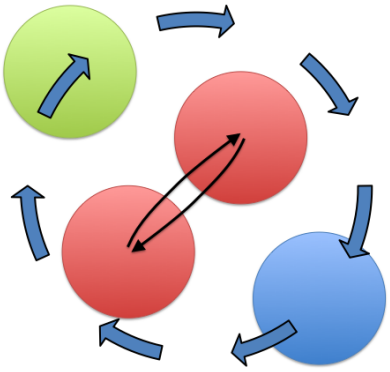

Fig. 9: 2D cross section of 4D three class experiment

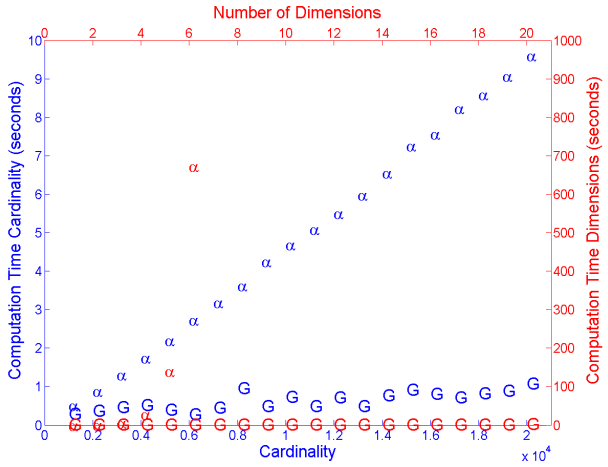

Fig. 11: Experimental time complexity for increasing dimensionality and cardinality of α-shapes α and GMMs (G)

CSE methods. GMMs appear to perform slightly better, but the confidence intervals indicate that there is no stastistically significant difference in performance. However, GMMs extracted core supports at an average of 0.172 seconds per time step, while α-shapes required an average of 13.013 seconds per time step. The difference in computational complexity is significant, especially if COMPOSE is running online.

We also looked at the empirical computational complexity of the two approaches. As stated previously, the time complexity of α-shapes is exponential with respect to dimensionality and linear with respect to number of instances. GMMs time complexity is less clear, because it depends on the initial conditions and is iterative by nature. Figure 11 compares the experimental time complexity of α-shapes to GMMs as N and d are increased. The computational complexity with respect to dimensionality was executed on a dataset with $1,250$ instances and is shown in red (plotted on upper horizontal and right vertical axes). The computational complexity with respect to cardinality was executed on a 2D dataset and is shown in blue (plotted on lower horizontal and left vertical axes). In this experiment, we found GMMs to extract core supports substantially faster than α-shapes with respect to both N and d.

IV. Conclusions

Our analysis shows that choosing one CSE algorithm over another is a tradeoff; each algorithm has its own advantages. More specifically, α-shapes are fairly robust and are non-parametric by nature. However, in high dimensional datasets even with relatively small cardinality, they become computationally infeasible to run. GMMs perform much faster than α-shapes, but they are dependent on initial conditions and may not always converge to the correct solution. Furthermore, GMMs assume that the data distribution is in fact a mixture of Gaussians, which is rarely the case for real world data. If sufficient size K is chosen, the assumption may be violated without consequence, but the statistical backing for the

algorithm becomes meaningless. GMMs are also more suited for mismatched distributions within a class. As long as the GMM fits a component to the underrepresented distribution, it will sample sufficient number of core supports from that distribution.

There are many other options for CSE algorithms. We would like to analyze other density estimators such as Parzen windows and k-nearest neighbors. CSE is essentially a density estimation technique, so existing non-parametric density estimators are of great interest. Furthermore, we would like to analyze the impact of different CSE algorithms on the performance of COMPOSE on real world data, beyond the evaluated synthetic datasets. Until now, the computational requirements of α-shape compaction restricted the types of real world data on which COMPOSE could operate. Our newly proposed core support extraction method opens up the possibility for COMPOSE to classify on datasets with substantially larger dimensionality.

REFERENCES

[1] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, 2013.

[2] K. Dyer, R. Capo, and R. Polikar, "Compose: A semisupervised learning framework for initially labeled nonstationary streaming data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 12–26, 2014.

[3] K. Dyer and R. Polikar, "Semi-supervised learning in initially labeled non-stationary environments with gradual drift," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, Brisbane, Australia, 2012, pp. 1–9.

[4] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.

[5] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Melon University, Pittsburgh, PA, Tech. Rep., 2002.

[6] T. Joachims, "Transductive inference for text classification using support vector machines," in *International Conference on Machine Learning (ICML)*, Bled, Slovenia, 1999.

[7] R. Capo, K. Dyer, and R. Polikar, "Active learning in nonstationary environments," in *International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, 2013.

[8] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions on Graphics (TOG)*, vol. 13, no. 1, pp. 43–72, Jan. 1994.

[9] D. A. Reynolds, "Gaussian mixture models." in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds. Springer US, 2009, pp. 659–663.

[10] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer-Verlag, 2008, ch. 9, pp. 191–214.

[11] G. Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[12] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proceedings National Institute of Science, India*, vol. 2, no. 1, Apr. 1936, pp. 49–55.

[13] S. Kung, M. Mak, and S. Lin, *Biometric Authentication: A Machine Learning Approach*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2004, ch. Expectation Maximization Theory, p. 64.