Online Gaussian Mixture Model for Concept Modeling and Discovery

Chunsheng Fang and Anca L. Ralescu

Machine Learning and Computational Intelligence Laboratory
Department of Computer Science, ML 0030
University of Cincinnati
Cincinnati, OH, 45237-0030, USA
fangcg@email.uc.edu, Anca.Ralescu@uc.edu

***Abstract*—How to model a concept, and how to discover a new concept, remain fundamental in machine learning research. Real world concepts are usually high-dimensional and have complicated distributions. Gaussian Mixture Model has strength in modeling complicated distributions. In this paper, we propose a data-driven concept modeling and discovery framework using GMM, with online updating mechanism for fast computation in real world application. Experiments show that our proposed algorithm can handle complicated concepts modeling and discovery with satisfactory performance in real time.**

I. INTRODUCTION

How to model a concept, and discover a new concept are fundamental in knowledge representation research, e.g. in machine learning, cognitive science, data mining. A concept is a combination of properties associated with salience weights. Detailed explanation will be given in Section II.

Real world concepts are usually not easy to describe, i.e. they can be having different combinations of properties, which can be modeled by clusters in conceptual space, i.e. can have complicated distribution in probabilistic interpretation. For example, the *apple* concept can be *red, smooth, round* or *green, smooth, round* or even *brown, wrinkled, round*, etc. Therefore, for this example there will be three distributions in conceptual space, which will be then modeled by a set of Gaussian mixture models.

A concept can be either defined by an expert manually, or learned from enough samples automatically using machine learning theory. In this paper, we propose a novel framework for modeling concepts automatically and adaptively, learning from enough observations of some certain concept in application.

Other than traditional *crisp concept*, *strict concept* and *loose concept* are required in some applications and also can be controlled in this framework by threshold settings. A *strict concept* is important in some critical environments such as "apple industry" if the manufacturing line want to pick certain type of "red" color or "green" color as for different pricing, it can be achieved by limit the concept range, i.e. *strict concept*.

Gaussian Mixture Modeling can approximate a complicated distribution using multi-Gaussian distributions. Recently GMM has successful applications in background modeling for computer vision [6], etc. It's obvious that complicated distribution can be modeled using a set of Gaussian distributions, which illustrated as Fig. 1. GMM is also widely used in content based image retrieval[4].

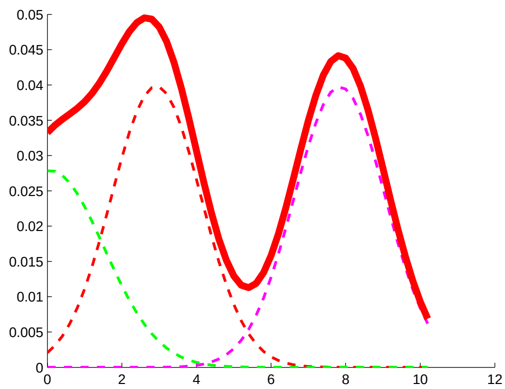

Fig. 1. GMM approximation

Expectation Maximization (EM) [1] is a common algorithm for estimating the GMM parameters. EM algorithm is used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. EM alternates between performing an expectation (E) step, which computes an expectation of the likelihood by including the latent variables as if they were observed, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step. The parameters found on the M step are then used to begin another E step, and the process is repeated. EM will finally converge to a local minimal parameter estimation of the GMM distribution.

However, EM is computation intensive if there are too many parameters to estimate. Besides, it cannot be applied to online approximation because EM requires a constant observation set for estimating the parameters in all iterations.

Therefore, online approximation for GMM is proposed in this paper to solve this concept modeling problem. A series of observations can be considered as random process from time

0 to time t. The advantages for this algorithm are that it does not require a constant set of observations, in other word, it can adapt to the new coming observations while memorizing its history (keeping record of the GMM parameters and updating it), and that it can be computed in real time for practical applications. It is also called incremental learning, which keeps both recent knowledge and historical knowledge in consistency, which is more desirable in machine learning applications.

Note that when human is learning a concept in early age, it requires enough observations to help "build up" the concept in mind. For instance, if a baby sees only one apple, it is not likely that it will learn the concept of *apple* perfectly. If the baby sees a certain amount of *apple* observations, the *apple* concept can be built and has some kind of "tolerance" for this *apple* concept, such as how red or green it can be. In the same sense, we argue that the "concept" can be modeled by a set of Gaussian mixtures given enough observations for that concept. Besides, the "tolerance" can be also modeled as *strict concept* or "loose concept" , which is achievable by threshold setting based on different applications.

It's also worthy to mention that to quantify "salience weight" for each property in conceptual space, we can implement it using different similarity measures, e.g. logistic function, or fuzzy hamming distance [3] in recent years.

This paper is organized as follows: Section II explains concept modeling in conceptual space and its theoretical fundamentals. Section III describes Gaussian Mixture Models and its online updating mechanism. Section IV proposes the algorithms for concept modeling and discovery. Finally in Section V will illustrate the experiment results and followed with Section VI concluding the contributions of this paper and possible future work.

II. CONCEPT MODELING

Conceptual space can provide an approach to knowledge representation that exploits the real world attributes while preserving their semantics.[5] The meaning of dimension, domain, property, and concept are explained as follows.

A. Dimension

A conceptual space can be modeled with a set of dimensions capable of describing the quality attributes of the information to be represented. These dimensions can be either psychophysical (e.g. natural language) or scientific (e.g. measuring the values associated with sensors, actuators, etc.) For a given application, there is generally no unique assignment of dimensions. Experts familiar with that application knowledge will specify the appropriate dimensions that capture its essential qualities. Dimensions generally possess geometric and/or topological structures that enable us to measure distances between two values.

B. Domain

Dimensions can be organized into multiple domains. Objects in a conceptual space are represented by points, one in each domain, that characterize their dimensional values. This constitutes the fundamental geometric character of conceptual spaces for knowledge representation. Note that only within the same domain can we measure the similarity between two objects as the similarity between their corresponding points. Using the *apple* concept example, we can measure the similarity between "red" and "green" in color domain, but we cannot naively measure the similarity between "red" in color domain and "sweet" in taste domain.

C. Property

A property is a convex region in some domain. The notion of convexity for property regions arises from the logical assumption that if two objects possess some property, then all objects located between them should likewise possess that property. Here, the "betweenness" can have different definition. In natural languages, properties often correspond to adjective-like descriptions (e.g., "red" , "tall" , or "round") in a particular domain. Properties can also capture more complex descriptions of objects, including shapes, actions and functional characteristics. They can also be defined in probabilistic or fuzzy terms, which provide a distinct advantage over representational schemes that require strict set membership. In other word, property in some domain can be obtained by applying "Voronoi tessellation" (from data clustering theory) of the domain, which divides the total volume into regions.

D. Concept

A concept is a combination of properties, typically across multiple domains, along with the salience weights associated with each property and the correlations (used in the sense of co-occurrences, as opposed to statistical correlations which can be positive or negative) between properties. The choice of properties is predicated upon the descriptive features of the application. The salience weights may be dependent upon the context.

For example, the concept *apple* may include the domains of color, taste, surface texture, shape, nutritional content and density, with multiple property regions within each domain to account for the various types of apples. In a visual context, color properties may be quite important, while in a cooking context, much less so. Furthermore, there will be distinct correlations between red, green or yellow color properties and a smooth surface texture property, and likewise between brown color and a wrinkled surface texture, and there will be anti-correlations between the converse pairings, e.g., brown with smooth, which do not occur together.

In this paper, we model the concept basically using a set of properties, which can be considered as the coordinate axis in the conceptual space, as well as salience weights associated with each property, which can be considered as the length on each property coordinate axis.

III. GAUSSIAN MIXTURE MODEL

Gaussian distribution is a basic probabilistic distribution, which is generally defined as (1), where X_t is the observation in time t, and μ , Σ are mean and variance for the Gaussian distribution respectively.

$$\eta(X_t,\mu,\Sigma)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\exp(-\frac{1}{2}(X_t,\mu)^T\Sigma^{-1}(X_t,\mu)) \quad (1)$$

In this paper, we consider every Gaussian distribution as one dimensional without lost of generality, because if a concept owns a combination of domains, e.g. K dimensional Gaussian distribution, we can assume the variance matrix as identity matrix, and then according to the independence of Gaussian distribution, this K dimensional Gaussian can be modeled as K 1-dimensional Gaussian distribution.

Gaussian Mixture Model is defined as the sum of a set of weighted Gaussian distribution. The recent history of observations $\{X_1, X_2, \ldots, X_t\}$ for one concept is modeled by a mixture of K Gaussian distributions. The probability of the current observation in GMM is defined as (2), where w_i,t is the weight for ith Gaussian in time t, K is the number of Gaussian used for approximate the distribution in conceptual space.

$$P(X_t)=\sum_{i=1}^{K} w_{i,t} * \eta(X_t,\mu,\Sigma) \quad (2)$$

Each time a new observation X_t comes in, the online GMM algorithm has to update the following three sets of parameters:

- The weight for each Gaussian (3), where $0 < \alpha < 1$, is the learning rate, and indicates the importance for the GMM concept of the incoming knowledge. If the concept drifts often, α can be set higher, and vice visa. Each time GMM update finishes, the weights are normalized to 1 so that the definition of probability is guaranteed.

$$w_{i,t} \;=\; (1-\alpha)w_{i,t}+\alpha M_{i,t} \quad (3)$$

where

$$M_{i,t} \;=\; \begin{cases} 1 & \text{if } X_t \text{ matches the } ith \text{ Gaussian} \\ 0 & \text{if no match} \end{cases}$$

- Mean for each Gaussian as (4). The meaning of this update is obvious: the matched Gaussian will shift closer to the new coming observation if match;

$$\mu_t=(1-\rho)*\mu_{t-1}+\rho * X_t \quad (4)$$

- Variance for each Gaussian as (5) and (6);

$$\sigma_t^2=(1-\rho)\sigma_{t-1}^2+\rho(X_t-\mu_t)^T(X_t-\mu_t) \quad (5)$$

$$\rho=\alpha\eta(X_t|\mu_k,\sigma_k) \quad (6)$$

After a certain period of learning, i.e. certain amount of coming observations, the concept modeled by this online GMM algorithm converges, which means that the concept is well-learned by the GMM models.

IV. ALGORITHM

In this section, two algorithms are proposed for concept modeling and concept discovery respectively.

A. GMM Concept Modeling

As explained in Section II, concept modeling in conceptual space can be described as Gaussian mixtures, given a series of observations X_t. Detailed algorithm is shown as Algorithm 1.

In Algorithm 1, we assume all properties are independent, so we can apply online GMM to each property, which is implemented as $X_t(Dimension)$.

For each property, we use K Gaussian to learn the real distribution online. If ith dimension of the new coming observation X_t match the existing jth Gaussian, then we update its corresponding weight, mean, variance according to (3) to (6).

Sort all the Gaussian mixtures in ith dimension property decreasingly by weight/variance, which is a measure of importance of each Gaussian distribution. If no match is found, we replace the least important Gaussian mean with the ith dimension value of new coming observation, and replace its variance and weight with initial values.

Therefore, in every round of iteration, the GMM is updated based on the new coming observation without destroying the previous history. After certain amount of observations, the concept C will converge to a set of GMM parameters, $\{mean(i,j)$, $var(i,j)$, $weight(i,j)\}$.

Strict or loose concepts can be also modeled by line 6 in Algorithm (1), because BW threshold actually determines the *radius* of the Gaussian distribution hyper-sphere in the conceptual space. If the application requires strict concept, the BW value can be set as low as possible, and maybe more Gaussian mixtures will be needed in approximate the actual concept distribution, vice visa.

B. GMM Concept Discovery

A new concept can be found if no match consistently occurs in the current concept. For example, if the *apple* concept is learned by our online GMM algorithm as red, smooth, round or green, smooth, round or brown, wrinkled, round. If after a period of time the coming observations are more like yellow, smooth, ellipsoid, then we can be more confident that the observations for *pear* concept are coming, which is a new concept in the conceptual space and also be learned in the same framework.

In our proposed algorithm in Algorithm 2, if the number of no match occurs is greater than a threshold T, then all the unmatched observations are redirect to another new concept learning process, which calls the same *GMMConceptModeling* function.

V. EXPERIMENT

In this section, experiments are performed on some synthesized observation dataset of the *apple* concept appeared as

Algorithm 1 Concept Modeling Function GMMConceptModeling(X_t)

Define properties in conceptual space for the concept C ;
Formulate the input stream of observations for some concept C, $Xt = t$th observation for C;
$Dimension \leftarrow Number\,of\,properties$;
$\alpha \leftarrow Learning\,rate$;
$K \leftarrow Number\,of\,Gaussians$;
$BW \leftarrow BandwidthofGaussian$;
$mean(Dimension, K) \leftarrow mean\,for\,each\,Gaussian$;
$var(Dimension, K) \leftarrow variancefor\,each\,Gaussian$;
$weight[Dimension, K] \leftarrow weightfor\,each\,Gaussian$;
for each X_t **do**
  for $i = 1$ to $Dimension$ **do**
    $match \leftarrow 0$;
    for $j = 1$ to K **do**
      if $|X_t(i) - mean(i, j)| < BWvar(i, j)$ **then**
        $match = j$;
        $break$;
      end if
    end for
    for $j = 1$ to K **do**
      if $match == j$ **then**
        $weight(i, j) = (1 - \alpha) \cdot weight(i, j) + \alpha$;
      else
        $weight(i, j) = (1 - \alpha) \cdot weight(i, j)$;
      end if
    end for
    for $j = 1$ to K **do**
      if $match == j$ **then**
        $pt = Gaussian(mean(i, j), var(i, j))$;
        $mean(i, j) = (1 - \alpha \cdot pt) \cdot mean(i, j) + \alpha \cdot pt \cdot Xt(i)$;
        $var(i, j) = (1-\alpha \cdot pt) \cdot var(i, j)+\alpha \cdot pt \cdot (Xt(i) - mean(i, j))^2$;
      end if
    end for
    Sort $weight(i, j)/var(i, j)$ decreasingly and rearrange $\{\ mean(i, j), var(i, j), weight(i, j)\ \}$ accordingly;
    Replace the least weight Gaussian if none is matched.
  end for
end for
$C \leftarrow \{mean(i, j), var(i, j), weight(i, j)\}$;
Return concept C;

Algorithm 2 Concept Discovery Function GMMConceptDiscovery(Xt)

$NMC \leftarrow$ Number of no matches obtained from $GMMConceptModeling$ function;
$T \leftarrow$ Threshold for new concept discovery;
if $(NMC > T)$ **then**
  $NewCon = GMMConceptModeling(X(t - NMC))$
end if
Return $NewCon$;

examples in this paper. All the experiments are finished in real time.

The learning rate is set to be 0.01 or 0.02, which is a good balance between current knowledge and history knowledge. The number of Gaussian K is 3. The more number of Gaussian is used, the better convergence will be achieved . Bandwidth is set to be 2.0, which affect the range of a Gaussian.

The first step is to specify all properties: we have color, shape, texture as the three properties for learning the *apple* concept. (Note that in real world application this specification can vary according to requirements.)

Without lost of generality, suppose the *apple* concept can be described as red, smooth, round and green, smooth, round. More complicated concepts can also be learned by this algorithm.

A collection of 200 observations of this *apple* concept are synthesized and the concept is learned by using *GMMConceptModeling* function.

Figure 2 to 4 illustrate the actual distribution for the concept, and the learned GMM results. After feeding 200 observations of the concept into the algorithm, the GMM converges to actual concept distribution. Blue histogram means the real distribution for the *apple* concept; Red asterisk curve means the learned GMM distribution; Red, magenta, green dashed curves mean the 1st, 2nd, 3rd Gaussian mixture component respectively. The peak of each Gaussian denotes its weight in the whole GMM distribution.

In Figure 3 and 4, because the actual distribution is more like single Gaussian distribution, we can see the GMM result is also quite convincing: Red asterisk curve is sufficient to approximate the actual distribution so the other two curves barely contribute to the final GMM distribution.

We can also see that in Figure 2 to 4 , there is still some gap between GMM distribution and actual distribution, which denotes the error between the GMM concept modeling and actual concept distribution. One of reason for this is that we need more observations to approximate a more accurate concept model. The other reason is for more complicated distribution, we need more Gaussian to approximate. Given enough observations for the concept, online GMM can converge to the actual distribution of a concept.

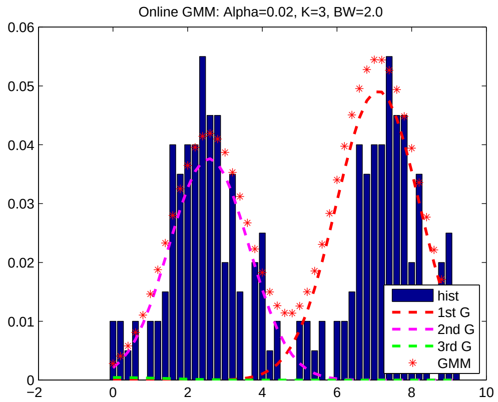

Fig. 2. GMM for *apple* concept color property

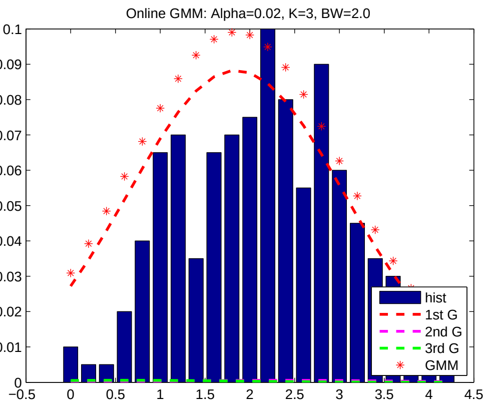

Fig. 3. GMM for *apple* concept texture property

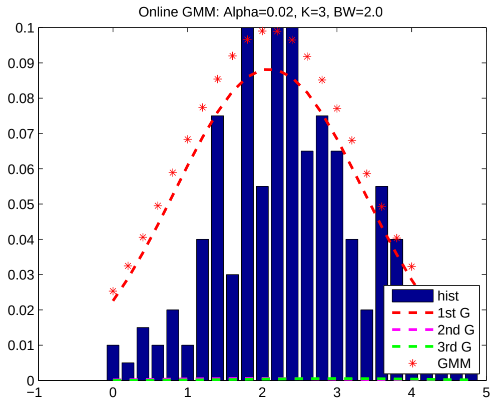

Fig. 4. GMM for *apple* concept shape property

VI. Conclusion and Future Work

We propose a novel probabilistic framework for concept modeling and discovery using online Gaussian Mixture Model, and in conclusion, the contributions in this paper are:

- Propose a probabilistic framework for concept modeling and discovery. Under this framework, complicated concepts can be modeled using data-driven mechanism, and automatic new concept discovery is possible.
- Online Gaussian Mixture Model is employed to solve this distribution approximation problem in real time, and adaptive to the emerging amount of observations belonging to some concept, i.e. incremental learning.
- Algorithms for concept modeling and discovery are provided which can be applied to practical applications, and real-time performance can be guaranteed.

Based on current progress proposed in this paper, how to model other types of concepts, e.g. , verbs, which are connections between nouns ? Note that in our algorithms, the properties have to be assigned beforehand, is that possible to incrementally increase additional properties when new instances containing such high-dimensional information of one concept come? These are remaining as future work.

Acknowledgment

This work was partially supported by the Department of the Navy, Grant ONR N000140710438. The valuable comments from anonymous reviewers are also appreciated.

References

[1] Dempster A., Laird N., and Rubin D. ”*Maximum likelihood from incomplete data via the EM algorithm*”, Journal of the Royal Statistical Society, 39 (Series B):1–38, 1977.

[2] Gärdenfors, P., Conceptual Spaces – The Geometry of Thought. The MIT Press, Cambridge, Massachusetts, 2000.

[3] Ionescu M., Ralescu A. ”*Fuzzy hamming distance in a content-based image retrieval system*”, Proceedings of the 2004 IEEE International Conference on Fuzzy Systems, Vol. 3 1721–1726.

[4] Permuter H., Francos J., and Jermyn I. H. ”*Gaussian mixture models of texture and colour for image database retrieval*,” Proceedings of ICASSP ’03, the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol.3, III-569–72.

[5] Rickard T. ”*A Concept Geometry for Conceptual Spaces with Applications to Levels 2 & 3 Fusion, Version 2.0*”, Lockheed Martin Proprietary Information, July 19, 2005.

[6] Stauffer, C., Grimson, W. E. L. ”*Adaptive background mixture models for real-time tracking*”, IEEE Int’l Conf. on Computer Vision & Pattern Recognition, 1998.