Weakly Semi Supervised learning based Mixture Model With Two-Level Constraints

Adama Nouboukpo, Mohand Saïd Allili
Université du Québec en Outaouais
283 Boulevard Alexandre-Taché, Gatineau (Québec) Canada J8X 3X7
{adama.nouboukpo, mohandsaid.allili}@uqo.ca

Abstract

We propose a new weakly supervised approach for classification and clustering based on mixture models. Our approach integrates multi-level pairwise group and class constraints between samples to learn the underlying group structure of the data and propagate (scarce) initial labels to unlabelled data. Our algorithm assumes the number of classes is known but does not assume any prior knowledge about the number of mixture components in each class. Therefore, our model : (1) allocates multiple mixture components to individual classes, (2) estimates automatically the number of components of each class, 3) propagates class labels to unlabelled data in a consistent way to predefined constraints. Experiments on several real-world and synthetic data datasets show the robustness and performance of our model over state-of-the-art methods.

Introduction

Recently, semi-supervised learning (*SSL*) has received a great interest in the fields of pattern recognition and machine learning. It has been applied to domains such as data mining, information retrieval, bioinformatics, image analysis & processing and text classification, where significant improvements have been obtained in comparison to fully supervised or unsupervised methods (Van Engelen and Hoos 2019).

SSL algorithms are broadly divided into two main categories: SSL for classification and SSL for clustering also known as *constrained clustering*. While the former are usually trained upon a small amount of labelled data and a very large amount of unlabelled data (Van Engelen and Hoos 2019), the latter try to group data by incorporating side information from domain or user knowledge (Boulmerka and Allili 2018). Side information usually comes in the form of pairwise constraints (must-links and cannot-links) between samples of data (Shental et al. 2004), which can be directly observed or inferred as background knowledge from user feedback (Nouboukpo and Allili 2019). The *must-link* establishes the samples which must be in the same cluster (or class) and the *cannot-link* refers to those samples that cannot be in the same cluster (or class).

Among the most popular SSL methods dealing simultaneously with clustering and classification, we can find graph-based and generative methods (Van Engelen and Hoos 2019). Graph-based methods have demonstrated a great performance to separate classes with a manifold structures thanks to their ability to efficiently encode relational information among samples (Filali, Allili, and Nadjia 2016). However, given their *transductive* nature, they are not generalizable to classifying new data. Generative methods, on the other hand, can predict the outcome of unseen data, but lack the relational aspect between data samples (Van Engelen and Hoos 2019).

In the past, Gaussian mixture models (GMMs) have been investigated for SSL (Van Engelen and Hoos 2019). These models seek to discover group structures from data by maximizing a likelihood function while using user/domain knowledge to avoid poor local minima (Zhao and Miller 2005). This knowledge can be either class labels on a small proportion of data (Van Engelen and Hoos 2019) or hard pairwise relationships indicating whether particular instances should be grouped together (Shental et al. 2004). Side information can also be available as group relations among samples known as *chunklets*. For example, in social networks, groups can be constructed by forming strong communities. Likewise, superpixel groups can be formed in image/video segmentation using spatially/temporally contiguous pixels (Filali, Allili, and Nadjia 2016).

Previous SSL algorithms using GMMs provide effective ways to make use of both labelled and unlabelled data. However, they can lose their efficiency notably when the labelled samples are scarce and classes with complex structure (Zhao and Miller 2005). Indeed, when labelled data are scarce, classification is mainly driven by unlabelled data (i.e. unsupervised learning) which tend to assign unlabeled data to the closest classes in the feature space by maximizing their likelihood. Although this can maximise model fitting, it can be sub-optimal for applications such as image segmentation where spatial contiguity of classes (e.g., objects, scenes, etc.) is more desirable than data fitting. Thus, biased models should be more encouraged to meet the desired segmentation output. Note also that GMM has been used for classification where each class is assumed to constitute one mixture component but the model can not deal with multi-component classes (Shental et al. 2004).

In this paper, we propose a mixture model which efficiently integrates weak supervision in classication/clustering

data problems. The supervision can come in the form of pairwise or group relationships as well as partially labelled data. The group constraints and the number of data within each group can be application-driven or generated automatically by initially clustering the data into a large number of groups (Boulmerka, Allili, and Ait-Aoudia 2014; Nouboukpo and Allili 2019). The group constraints are defined at two levels. The first level encodes hard *musk-link* constraints imposing all the data in a group to be assigned to the same mixture component. The groups defined by these constraints constitute atomic parts or building blocks constituting large clusters. The second level encodes soft inter-group class affinity when available. Our mixture model seamlessly integrates the two-level constraints where each class can be constituted of one or multiple mixture components and have very few labelled data initially. Our model can therefore achieve optimal fitting and labelling to data generated by manifold-structured classes.

The rest of this paper is organized as follows. Section II presents in details our proposed semi-supervised algorithms based on the EM method. The datasets, the experimental and results are presented in section III. Section IV draws the conclusion of this paper and discusses future work.

Proposed method

Before introducing our method, we first explain some terminology which will be used in this paper. Given a data set $X = \{x_n\}_{n=1}^{N}$, let us assume that $X = \bigcup_{i=1}^{M} S_i$ where S_i denotes a subset (chunklet or group) of samples x_n from the same unknown (Gaussian) distribution. We consider the assumption that chunklets are sampled i.i.d, with respect to the weight of their corresponding source (points within each chunklet are also sampled i.i.d).

Let us assume also that our data is composed of two subsets: $X_L = \{(S_i, t_i), ..., (S_{M^L}, t_{M^L})\}$ is the subset of labelled chunklets and $X_U = \{S_j, ..., S_{M^U}\}$ is the subset of unlabelled chunklets. Here, $t_i \in T$ is the class label from the label set $T = \{1, 2, ..., C\}$, M^L (M^U) denote the number of labelled chunklets (number of unlabelled chunklets) and $M = M^U + M^L$ denote the number of all chunklets (with $M^U >> M^L$). Moreover, we can separate X_L according to the labels in C disjoints sets $X_c = \{(S_i, t_i) | t_i = c, i = 1, \cdots, N_c\}$ one for each class where $X_L = \bigcup_{c=1}^{C} X_c$.

Let us suppose that K denotes the number of components and β_{ck} denotes the probability that component k is assigned to class c such that $\sum_{c=1}^{C} \beta_{ck} = 1$. For each component k, $k = 1, \cdots, K$, we assume the Gaussian PDF as $p(x_n|\theta_k)$ with $\theta_k = \{\mu_k, \Sigma_k\}$ where μ_k and Σ_k are the mean and covariance matrix. Each θ_k defines the parameter of the k^{th} component. We defined α_k as the mixing weights.

We first develop our learning objective assuming multiple component per class using only group-level constraints. The goal is to show the influence of using inter-groups in the model estimation and components classification. Without any class constraints, the complete log-likelihood function of our mixture model is given by:

$$\begin{aligned} Q(\Theta) = & \sum_{i=1}^{M^U} \sum_{k=1}^{K} \sum_{c=1}^{C} Z_{ik}^U V_{kc}^U \log \Big(\alpha_k \beta_{ck} \prod_{x_n \in S_i} p(x_n|\theta_k) \Big) \\ + & \sum_{c=1}^{C} \sum_{i=1}^{N_c} \sum_{k=1}^{K} Z_{ik}^L V_{kc}^L \log \Big(\alpha_k \beta_{ck} \prod_{x_n \in S_i} p(x_n|\theta_k) \Big) \end{aligned} \tag{1}$$

where:

- $\Theta = \{\alpha_k, \theta_k, \beta_{ck}, \forall k, c\}$ is the complete set of parameters needed to specify the model
- $Z_{ik} = Z_{ik}^U$ or Z_{ik}^L denotes the assignment matrix with $Z_{ik} = 1$ if chunklet S_i is assigned to component k else $Z_{ik} = 0$
- $V_{kc} = V_{kc}^U$ or V_{kc}^L denotes the cluster assignment with $V_{kc} = 1$ if component k is assigned to class c else $V_{kc} = 0$

By incorporating class constraints, Eq. 1 is modified by summing only over assignments which comply with the constraints (instead of summing over all possible assignments of data points to sources). The class constraints are derived from labelled samples which are used to specify what kind of structure is expected to be found. Thus, if two groups have different class labels, then this information indicates that they are known to be generated by different sources, which should of course have implications for their cluster assignments. This means that groups which do not belong to the same class should not belong to the same cluster/component. Our complete negative data log likelihood becomes:

$$\begin{aligned} J(\Theta) = & -Q(\Theta) \\ & + \frac{\gamma}{2} \sum_{i=1}^{M^L} \sum_{j=1}^{M^L} W_{ij} \sum_{c=1}^{C} \Big(\sum_{k=1}^{K} Z_{ik} V_{kc} \Big) \Big(\sum_{k'=1}^{K} Z_{ik'} V_{k'c} \Big) \end{aligned} \tag{2}$$

where γ is a positive number tuning the contribution of the penalty term. The weights W_{ij} ensure that chunklets containing data with different labels will not be assigned to the same class. In opposite, chunklets containing data with same labels are encouraged to be assigned to the same class. Their values are defined as follows:

$$W_{ij} = \begin{cases} 1 & \textit{cannot-link between } S_i \textit{ and } S_j \\ -1 & \textit{must-link between } S_i \textit{ and } S_j \\ 0 & \textit{otherwise} \end{cases}$$

The term associated with class constraints works as regularisation that force the model to select the appropriate group structure and choose the optimal number of components per class. To estimate the model parameters, the

objective is to maximize the log-likelihood function with respect to the parameters in Θ using the EM algorithm which consists of an E-Step and M-Step. Our resulting EM is carrying through the mean-field approximation as (Zhao and Miller 2005).

E-Step I (No class constraints assumed): The chunklet posterior probability derived from Eq. 1 is:

$$\tau^U_{ikc} = \frac{h_{nkc}}{\sum_{m=1}^{K} \alpha_m \prod_{x_n \in S_i} p(x_n|m,\theta_m)}$$
$$\tau^L_{ikc} = \frac{h_{nkc}}{\sum_{m=1}^{K} h_{nmc}} \quad (3)$$

where $\tau_{ikc} = p(Z_{ik}, V_{kc}|Si)$ is the probability that S_i is generated by component k and component k is generated by class c and $h_{nkc} = \alpha_k \beta_{ck} \prod_{x_n \in S_i} p(x_n|\theta_k)$

E-Step II (Class constraints assumed): Using the class constraints, the chunklet posterior probability becomes:

$$\tau^U_{ikc} = \frac{h_{nkc}}{\sum_{m=1}^{K}\sum_{c=1}^{C} h_{nmc}}$$
$$\tau^L_{ikc} = \frac{h_{nkc} \exp\left(-\frac{\gamma}{2}\sum_{j=1}^{M^L} W_{ij}\Big(\sum_{\substack{k'=1\\k'\neq k}}^{K} \tau^L_{jk'c}\Big)\right)}{\sum_{m=1}^{K}\sum_{c=1}^{C} h_{nmc} \exp\left(-\frac{\gamma}{2}\sum_{j=1}^{M^L} W_{ij}\Big(\sum_{\substack{m'=1\\m'\neq m}}^{K} \tau^L_{jm'c}\Big)\right)} \quad (4)$$

M-Step : The parameter updates of Eq. 1 or Eq. 2 take the form:

$$\alpha_k = \frac{\sum_{i=1}^{M^U} \tau^U_{ik} + \sum_{c=1}^{C}\sum_{i=1}^{N_c} \tau^L_{ikc}}{M} \quad (5)$$

$$\beta_{ck} = \frac{\sum_{i=1}^{M^U} \tau^U_{ikc} + \sum_{i=1}^{N_c} \tau^L_{ikc}}{M\alpha_k} \quad (6)$$

$$\mu_k = \frac{\sum_{i=1}^{M^U} \tau^U_{ik} \sum_{x_n \in S_i} x_n + \sum_{c=1}^{C}\sum_{i=1}^{N_c} \tau^L_{ikc} \sum_{x_n \in S_i} x_n}{\sum_{i=1}^{M^U} \tau^U_{ik}|S_i| + \sum_{c=1}^{C}\sum_{i=1}^{N_c} \tau^L_{ikc}|S_i|} \quad (7)$$

$$\Sigma_k = \frac{\sum_{i=1}^{M^U} \lambda_{ik}\tau^U_{ik} + \sum_{c=1}^{C}\sum_{i=1}^{N_c} \lambda_{ik}\tau^L_{ikc}}{\sum_{i=1}^{M^U} \tau^U_{ik}|S_i| + \sum_{c=1}^{C}\sum_{i=1}^{N_c} \tau^L_{ikc}|S_i|} \quad (8)$$

where $|S_i|$ denotes the number of points in chunklet S_i, $\lambda_{ik} = \sum_{x_n \in S_i} (x_n - \mu_k)(x_n - \mu_k)^T$ and $\tau^U_{ik} = \sum_{c=1}^{C} \tau^U_{ikc}$.

Model selection

In order to estimate automatically the number of components, we adopt the deterministic method. This method starts by obtaining a set of candidate models for a range of values of K (from K_{min} to K_{max}) which is assumed to contain the true/optimal K_{best} by minimizing using a cost function. The cost function is a penalized negative likelihood using Minimum Message Length (MML) (Figueiredo and Jain 2002) and has the form as follows:

$$\mathcal{F}(\hat{\theta}_k, K) = J(\Theta) + \frac{A}{2}\sum_{k=1}^{K} \log(\frac{M\alpha_k}{12}) + \frac{K}{2}\log\frac{A}{12} + \frac{K(A+1)}{2} \quad (9)$$

where A is the number of parameters in each component. In our work, we set $K_{min} \geq C$ to prevent that some classes will not be represented by the model. The steps composing our model estimation can be summarised in the following algorithm:

Algorithm 1: Our proposed algorithm

Input : N chunklets, K_{min}, K_{max}
Output : Mixture model in $\hat{\Theta}_{best}$ and K_{best}

$t \leftarrow 0$, $K_{best} \leftarrow K_{max}$, $\mathcal{F}_{min} \leftarrow +\infty$

Initialize the parameters
$\widehat{\Theta}(0) = \{\alpha_k, \mu_k, \Sigma_k, \beta_{ck}, \forall k\}$

Compute W_{ij}

while $K_{best} > K_{min}$ **do**

- $t \leftarrow t + 1$
- Evaluate the responsibilities using Eq. 4
- Re-estimate the mixing weights using Eq. 5
- Check if there are irrelevant components:

  if $\alpha_k < 10^{-6}$ **then**
  - ⋆ Discard the component k
  - ⋆ Set $K_{best} \leftarrow K_{best} - 1$
  - ⋆ Renormalizes the remaining mixing weights

  end
- Re-estimate the rest of the parameters using Eq. 6, 7 and 8
- Evaluate $\mathcal{F}_t$ using Eq. 9
- **if** $\mathcal{F}_t \leq \mathcal{F}_{min}$ **then**
  - ⋆ $\mathcal{F}_{min} \leftarrow \mathcal{F}_t$
  - ⋆ $\hat{\Theta}_{best} \leftarrow \hat{\Theta}(t)$

  end

end

Experiments

To evaluate the performance of our approach, we conducted experiments on five datasets such two synthetic datasets *Syn-*

data3G and *Twomoons* and three real-word datasets from Waveform (Dua and Graff 2017), MNIST (MNIST 2018), Banana (Team 2017) presented in Table 1. On synthetic and real-word datasets, we compare our proposed method to (Zhao and Miller 2005) denoted by *MCGMM* which surpassed other existing generative SSL methods. We use the combined measure of *Purity* and *Accuracy* scores named ρ_c to make the performance comparison. The combination is defined as follows:

$$\rho_c = \frac{2Purity * Accuracy}{Purity + Accuracy} \tag{10}$$

where Purity measures the homogeneity of estimated classes, i.e., how many of the estimated class points belong to a single true class and Accuracy measures how many of the true class points reside in a single estimated class (rather than being spread over several estimated classes).

Note that, the larger value of ρ_c indicates the best result. All the datasets were split into 70% and 30% ratio for training and test sets. We assume that the number of classes is known but the number of mixture components is unknown and must be inferred from the data. The performance curves were obtained by varying the labelled set size. The reported results is based on average over 10 executions.

Table 1: Summary of the employed datasets: N = Number of samples, D = Number of dimensions, C = Number of Classes

Dataset	N	D	C
Syndata3G	1350	2	2
Twomoons	1650	2	2
Banana	5300	2	2
Waveform	5000	40	3
MNIST : digit 1,2 and 3	3177	784	3
MNIST : digit 4,5, 6 and 7	3860	784	4

Employed dataset

Note that, we choose datasets that have high overlapping clusters and classes and have manifold structure in order to evaluate the robustness of our methods. *Syndata3G* is built with three components from two classes (with one containing two components). We randomly generate 30 labelled chunklets and 170 unlabelled chunklets for the three components. In each chunklet, we randomly generate sample between the range $[1, 10]$ to form the final dataset (see Fig. 1a). The number of class constraints is randomly chosen as 15% of the chunklets length for *Syndata3G*. The *Twomoons* dataset consists of 1650 samples and is manifold moons structure (see Fig. 2a). For a fair comparison on *Twomoons* dataset, the number of class constraints is chosen as 30% in each class. The labelled samples coupled the unlabelled samples for *Syndata3G* and *Twomoons* datasets are shown respectively in Fig. 1b and Fig. 2b.

For the MNIST, we divided the original dataset into two datasets (see Table 1) such that the first one named $mnist123$ contains the digits 1,2 and 3 and the second one named $mnist4567$ owns the digits four to seven. As preprocessing for *Waveform, mnist123, mnist4567*, standard principal component analysis was used to reduce the dimension for some dataset. We produce chunklets using KMeans algorithm. Afterwards, we assign every sample to its nearest initial chunklet. The constraints derived from classes are chosen randomly at each execution.

Evaluation results and Discussion

We used *Syndata3G* and *Twomoons* to verify the effectiveness and robustness of the clustering/classification results of our algorithm.

We denote the combined measure of Purity-Accuracy with respect to cluster by ρ. As we can see in Fig. 1 and 2, the different results of ρ_c shown below, demonstrate that our algorithm perform better than the $MCGMM$ method. Moreover, we clearly see that in Fig. 1c, 1d and 1e, our simplified and proposed method capture well the ground-truth clusters than *MCGMM*. Note that, we did not compare the clustering result on *Twomoons* data because the groundthruth clusters are not available. Our simplified and proposed method captures the true labels and clusters because as the constraints in second level are chosen randomly (which can be inconsistent and maybe hurt the performance), we decrease the sample spreading sensibility and the effect of inconsistent constraints by our first level constraint.

For the real-word datasets, as shown in figure 3a, 3c, 3d and 3b, the obtained average ρ_c results of our simplified method and our proposed method are better than $MCGMM$ method. These results are not a surprise because our methods avoid the spreading of data. In other word, we reduce initialisation sensibility by firstly grouping points who are most similar using. Our proposed method gives better performance than the simplification which ignore dependence on labelled samples. This suggests that it is important to account for the role played by labelled samples. The other advantage of our algorithms is that, as the number of constraints increases also our algorithm increases the classification performances. We argued that when the number of samples are too high the complexity of *MCGMM* increase heavily.

Foreground image segmentation

The proposed method is applied for foreground image segmentation where $C = 2$ (foreground/background) using natural color image dataset MSRA10K (Cheng et al. 2015). Our method is compared to existing algorithms such as (Glaister, Wong, and Clausi 2014), (Scharfenberger et al. 2013), (Martinez-Uso, Pla, and Sotoca 2010) and (Rother, Kolmogorov, and Blake 2004) named respectively TDLS, TD, SSLGMM and GC. TDLS is modified version of TD and both are based on statistical texture distinctiveness. SSLGMM is a semi-supervised apporaoch based on EM for model-based clustering. GC is based on graph-cut method using texture and edge information. Here, the chunklet is

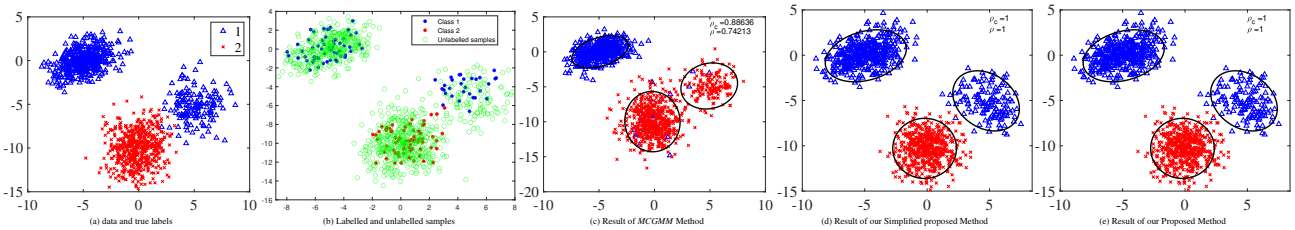


Figure 1: Average ρ and ρ_c scores comparison results for Syndata3G data

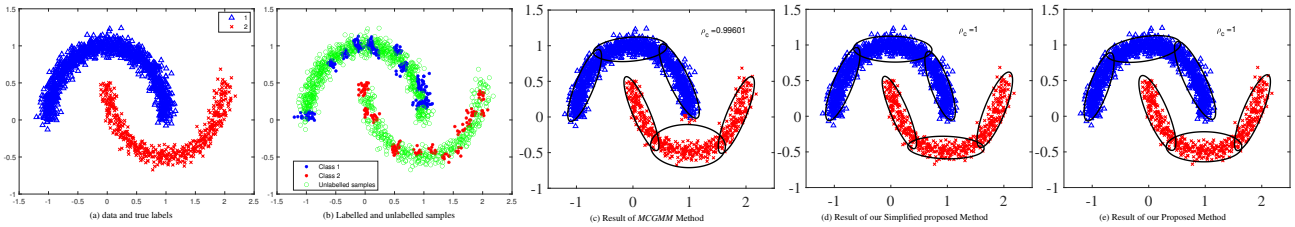
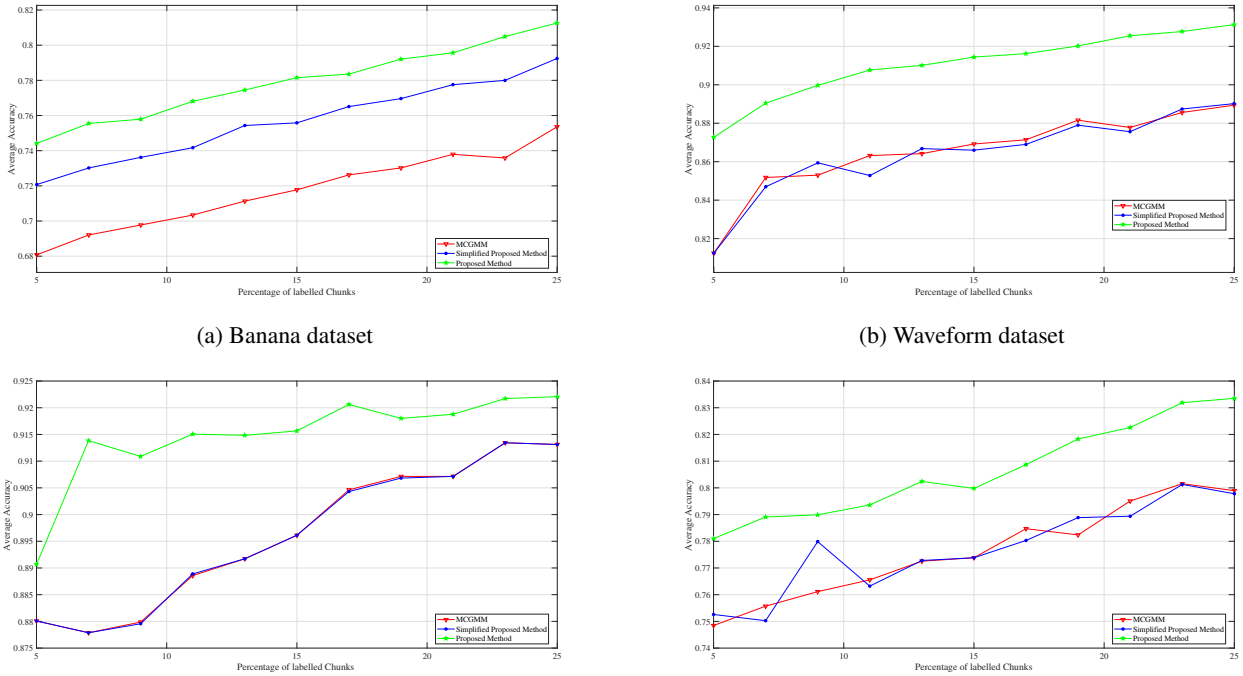


Figure 2: Average ρ_c score comparison results for twomoons data



(a) Banana dataset

(b) Waveform dataset

(c) mnist123 dataset

(d) mnist4567 dataset

Figure 3: Average ρ_c score results on real-word datasets

constructed by the concept of superpixel using the SLIC (Achanta et al. 2012) method. Wij is computed using the spatially adjacent of each superpixel and updated by manual labbelled superpixels (between 5 and 10). Therefore, we let our algorithms to form the clusters of the set of produced superpixels. After, each cluster is classified as foreground or background and propagate this information to unlabelled superpixels.

We evaluate the segmentation result with 500 images using ρ_c score that measure how close the predicted boundary of an object matches the groundthruth boundary. Figure 4 shows the visual comparison result provided by our proposed method compared to the groundtruth. By comparing the values listed in the Table 2, it is observed that, our algorithm achieved the highest value than other algorithms.

Table 2: Quantitative comparison results

Method	GC	SSLGMM	TD	TDLS	Ours
ρ_c	88.64	90.12	95.78	98.59	**99.7**

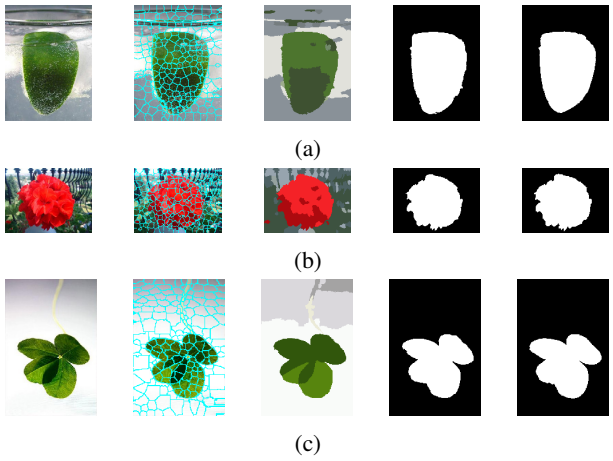


Figure 4: Qualitative result: Original image(first column), Estimated superpixels(second column) , Proposed method with optimal L (third column) Our segmentation map (forth column) Groundtruth(last column)

Conclusion

In this work, we proposed a generative model integrating weak supervision for semi-supervised classification and clustering. The supervision comes in the form of group constraints where the samples of each group (chunklet) are assumed to belong to one class label and be generated by the same mixture component. This supervision enables our model to seamlessly integrate spatial relationships between data, and for each class to be constituted of one or multiple mixture components. Our model can therefore achieve optimal fitting and labelling to data generated by classes with complex manifold structure. This work can be easily extended to other types of distributions such as Generalized Gaussian and Student distribution. It can also be readily adapted to discrete data.

Acknowledgments.

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support.

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S.; et al. 2012. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence* 34(11):2274–2282.

Boulmerka, A.; Allili, M. S.; and Ait-Aoudia, S. 2014. A generalized multiclass histogram thresholding approach based on mixture modelling. *Pattern recognition* 47(3):1330–1348.

Boulmerka, A., and Allili, M. S. 2018. Foreground segmentation in videos combining general gaussian mixture modeling and spatial information. *IEEE Trans. Circuits and Systems for Video Technology* 28(6):1330–1345.

Cheng, M.-M.; Mitra, N. J.; Huang, X.; Torr, P. H.; and Hu, S.-M. 2015. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3):569–582.

Dua, D., and Graff, C. 2017. UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

Figueiredo, M. A. T., and Jain, A. K. 2002. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence* 24(3):381–396.

Filali, I.; Allili, M. S.; and Nadjia, B. 2016. Multi-scale salient object detection using graph ranking and global-local saliency refinement. *Signal Processing: Image Communication* 47:380–401.

Glaister, J.; Wong, A.; and Clausi, D. A. 2014. Segmentation of skin lesions from digital images using joint statistical texture distinctiveness. *IEEE transactions on biomedical engineering* 61(4):1220–1230.

Martinez-Uso, A.; Pla, F.; and Sotoca, J. M. 2010. A semi-supervised gaussian mixture model for image segmentation. In *2010 20th International Conference on Pattern Recognition*, 2941–2944. IEEE.

MNIST. 2018. Mnist database. `http://yann.lecun.com/exdb/mnist/`. Accessed: 2020-01-12.

Nouboukpo, A., and Allili, M. S. 2019. Spatially-coherent segmentation using hierarchical gaussian mixture reduction based on cauchy-schwarz divergence. In *International Conference on Image Analysis and Recognition*, 388–396. Springer.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)* 23(3):309–314.

Scharfenberger, C.; Wong, A.; Fergani, K.; Zelek, J. S.; and Clausi, D. A. 2013. Statistical textural distinctiveness for salient region detection in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 979–986.

Shental, N.; Bar-Hillel, A.; Hertz, T.; and Weinshall, D. 2004. Computing gaussian mixture models with em using equivalence constraints. In *Advances in neural information processing systems*, 465–472.

Team, S. 2017. Scilab tutorials. `https://www.scilab.org/tutorials/machine-learning-%E2%80%93-classification-svm`. Accessed: 2020-01-12.

Van Engelen, J. E., and Hoos, H. H. 2019. A survey on semi-supervised learning. *Machine Learning* 1–68.

Zhao, Q., and Miller, D. J. 2005. Mixture modeling with pairwise, instance-level class constraints. *Neural computation* 17(11):2482–2507.