Pattern Recognition using Functions of Multiple Instances

Alina Zare and Paul Gader
Department of Computer and Information Science and Engineering, University of Florida
azare@cise.ufl.edu, pgader@cise.ufl.edu

Abstract

The Functions of Multiple Instances (FUMI) method for learning a target prototype from data points that are functions of target and non-target prototypes is introduced. In this paper, a specific case is considered where, given data points which are convex combinations of a target prototype and several non-target prototypes, the Convex-FUMI (C-FUMI) method learns the target and non-target patterns, the number of non-target patterns, and determines the weights (or proportions) of all the prototypes for each data point. For this method, training data need only binary labels indicating whether the data contains or does not contain some proportion of the target prototype; the specific target weights for the training data are not needed. After learning the target prototype using the binary-labeled training data, target detection is performed on test data. Results showing detection of the skin in hyperspectral imagery and sub-pixel target detection in simulated data are presented.

1. Introduction

The general notion of a FUMI algorithm learns target and non-target prototypes given a set of data points that are some unknown function of the target and non-target prototypes. Suppose there is a given data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ where each data point is some unknown function of prototypes, $\mathbf{x}_i = f(\mathbf{B}_i, \mathbf{P}_i)$ where $\mathbf{P}_i$ are the set of parameters for $\mathbf{x}_i$ and $\mathbf{B}_i$ is the "bag" of prototypes that contribute in a non-negligible way to the data points $\mathbf{x}_i$. In order to learn the target prototype, each training point $\mathbf{x}_i$ is given a binary label indicating whether the bag $\mathbf{B}_i$ contains or does not contain the target prototype, $\mathbf{b}_T$. Data points that are a function of prototype containing the target protoype are called positive training points, $\mathbf{x}_i^+ = f(\mathbf{B}_i^+, \mathbf{P}_i)$ where $\mathbf{b}_T \in \mathbf{B}_i^+$. After learning the target prototype using the binary-labeled training data, target detection can be performed on test data using methods outline below.

The FUMI algorithm can be related to Multiple Instance Learning (MIL) methods [2, 3]. In MIL, training data is divided into positive and negative "bags" in which each positive bag is a group of data points including at least one target point. In each positive bag, the exact number of data points belonging to the target class is unknown. Negative bags are composed entirely of non-target data points. The MIL methods are effective for learning target concepts and developing classifiers for cases where accurate sample-level labeled training data is unavailable. The FUMI method can be related to the MIL framework by recognizing each data point as being a function of a positive or negative bag. In this paper, the specific case that is considered is that each data point is assumed to be a convex combination of target and non-target prototypes, as shown in Equation 1 where the set of prototypes, $\mathbf{E}$, with non-zero weights for data points $\mathbf{x}_i$ define the bag $\mathbf{B}_i$.

$$\mathbf{x}_i = p_{iT}\mathbf{e}_T + \sum_{k=1}^{M} p_{ik}\mathbf{e}_k \tag{1}$$

where $\mathbf{x}_i$ is a data point, $\mathbf{e}_T$ is the target prototype, $\mathbf{e}_k$ is a non-target prototype for $k = 1, ..., M$ and p_{ik} is the weight (or proportion value) of the k^{th} prototype in data point i. The proportions are constrained to sum-to-one and be greater than zero.

$$p_{iT} + \sum_{k=1}^{M} p_{ik} = 1, \quad p_{iT} \geq 0, \quad p_{ik} \geq 0 \tag{2}$$

The positively labeled data points have some unknown positive weight on the target prototype, i.e., $\mathbf{x}_i^+ = p_{iT}\mathbf{e}_T + \sum_{k=1}^{M} p_{ik}\mathbf{e}_k$ with $p_{iT} > 0$. The negatively labeled points have zero weight on the target prototype, $\mathbf{x}_i^- = \sum_{k=1}^{M} p_{ik}\mathbf{e}_k$. The exact weight values for the training data are not needed. In addition to learning prototypes, the C-FUMI algorithm learns the number of non-target prototypes needed for a data set and determines the weights of prototypes for each data point.

$$G = (1-\mu)\sum_{i=1}^{N}\left|\left|\left(\mathbf{x}_i - \sum_{k=1}^{M} p_{ik}\mathbf{e}_k\right)\right|\right|_2^2 + \frac{\mu}{2}\sum_{k=1}^{M}\sum_{j=1}^{M}||(\mathbf{e}_k - \mathbf{e}_j)||_2^2 + \sum_{k=1}^{M}\gamma_k\sum_{i=1}^{N}p_{ik} = (1-\mu)R + \frac{\mu}{2}V + S \quad (3)$$

$$F = (1-\mu)\sum_{i=1}^{N}\left|\left|\left(\mathbf{x}_i - l(\mathbf{x}_i)p_{iT}\mathbf{e}_T - \sum_{k=1}^{M} p_{ik}\mathbf{e}_k\right)\right|\right|_2^2 + \frac{\mu}{2}R + \mu\sum_{k=1}^{M}||(\mathbf{e}_T - \mathbf{e}_k)||_2^2 + S + \sum_{\substack{i=1 \\ l(\mathbf{x}_i)=1}}^{N^+}\frac{1}{\sigma^2}(p_{iT}-1)^2 \quad (4)$$

2. The C-FUMI Algorithm

The C-FUMI Algorithm is an extension of the Sparsity Promoting Iterated Constrained Endmembers (SPICE) algorithm [4]. In SPICE, the prototypes and proportions are iteratively updated by minimizing the objective function in Equation 3 where $\gamma_k = \frac{\Gamma}{\sum_{i=1}^{N} p_{ik}}$ using the proportions from the previous iteration and Γ is parameter used to control the degree of sparsity. The first term of this objective computes the squared error between the input data and the estimate found using the current prototypes (or *endmembers*) and proportions. The second term produces prototypes that provide a tight fit around the data. The third term is a sparsity promoting term used to determine M, the number of endmembers needed to describe the input data. This objective is updated iteratively using alternating optimization on the endmembers and proportions.

The SPICE algorithm is an unsupervised algorithm to learn the proportions, $\mathbf{P}$, the prototypes, $\mathbf{E}$, and the number of endmembers, M, for a given unlabeled dataset. The C-FUMI algorithm extends the SPICE algorithm by using the positive and negative labeled training points to drive the C-FUMI algorithm to learn and distinguish the specific target prototype from the remaining non-target prototypes. The target prototype found can then be used for detection in test data.

Given the labeled training data, C-FUMI adds a Gaussian prior centered at 1 to the proportions associated with the target prototype for positively labeled data points. In contrast, for negatively labeled training data, the proportion value associated with the target prototype is constrained to be zero. Therefore, the objective function for C-FUMI can be written as shown in Equation 4 where $l(\mathbf{x}_i)$ is 1 when $\mathbf{x}_i$ is in the target class and 0 otherwise. The σ^2 parameter is Equation 4 can be adjusted based on the uncertainty of the labeling or prior knowledge of the range of proportions for the target endmember for the positively labeled data. For example, σ^2 can be set to a large value allowing for small proportions associated with the target endmember.

The C-FUMI algorithm updates the target and non-target endmembers, proportions, and number of endmembers by iteratively minimizing the objective function in Equation 4. In order to update both target and non-target endmembers, the proportions for all data points are held constant and Equation 4 is minimized by setting the derivative of the objective with respect to each endmember to zero and solving for the endmember value. When updating proportions, endmembers are held constant and Equation 4 is minimized subject to the constraints in Equation 2. Since this is a quadratic objective with linear constraints, a quadratic programming step is used to update the proportions. The sparsity promoting term (the 4^{th} term in the objective) is used to determine the number of non-target endmembers needed. This term drives the proportions associated with unneeded non-target endmembers to zero. Then, the unneeded endmembers can be removed with no effect on the squared error terms. The objective function is iteratively minimized until some stopping criterion is reached such as convergence or reaching a maximum number of iterations.

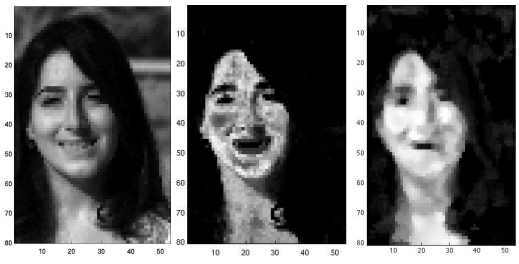
After learning the target and non-target prototypes, target detection on test data can be carried out. Two methods for target detection were considered. The first method, given the target and non-target prototypes and the test data, the proportions for all the prototypes for the given test data are computed by minimizing the residual sum of squared errors, $\left|\left|\left(\mathbf{x}_i - p_{iT}\mathbf{e}_T - \sum_{k=1}^{M} p_{ik}\mathbf{e}_k\right)\right|\right|_2^2$ subject to the constraints in Equation 2 using a quardratic programming step. The proportion value for the target endmember can be used as the detection statistic. An alternative method is the Hybrid Subpixel Detector method [1]. For the hybrid sub-pixel detector, the proportions are estimated for each data point twice. Once using both the learned target and non-target prototypes (to find $\mathbf{p}_i$ for each test data point) and once using only the non-target prototypes (to find $\mathbf{p}_i^*$ for each test data point). Then, the detection statistic is defined as the ratio of the resid-

ual errors from both sets of prototypes.

$$HSD = \frac{\left|\left|\left(\mathbf{x}_i - \sum_{k=1}^{M} p_{ik}^{*}\mathbf{e}_k\right)\right|\right|_2^2}{\left|\left|\left(\mathbf{x}_i - p_{iT}\mathbf{e}_T - \sum_{k=1}^{M} p_{ik}\mathbf{e}_k\right)\right|\right|_2^2} \quad (5)$$

3. Example Results

Two experiments were performed one with real hyperspectral images for skin detection and the other with synthetic data. The latter data set was used to investigate accuracy and produced some interesting results for further investigation. The first experiment was applied to visible/near-infrared hyperspectral imagery (with 120 spectral bands ranging for 410 to 845 nm in wavelength) for skin detection. The image (Figure 1(a)) contains 4240 total pixels. Note that there is considerable overlap between the skin pixels and other pixels in the image and that many of the skin pixels are in shadow. For training 100 target skin pixels and 100 non-target pixels were selected from the image. C-FUMI was applied to this training data with the following parameter settings: $\mu = 1 \times 10^{-3}$, $\Gamma = 0$, $M = 9$, and $\sigma^2 = 0.1$. After learning the prototypes, the proportions for every data point in the image was computed and the associated proportion maps for each of the 9 endmembers are shown in Figure 2. The first map shown in Figure 2 and in Figure 1(c) corresponds to the proportions found for the target endmember. The hybrid subpixel detector was also applied using the endmember found and results are shown in Figure 1(b).



(a) (b) (c)

Figure 1: (a) Original image data at wavelength 772 nm. (b) Hybrid Subpixel Detector Results for Skin Detection (c) Proportions of target endmember after spatial median filtering. The original image without filtering is shown in Figure 2.

In order to examine the capabilities of accurately detecting sub-pixel targets, C-FUMI was run on simulated data generated using random mixtures of three prototypes. The prototypes used were the first three prototypes found in the skin detection experiment. The prototypes correspond to the materials highlighted in the

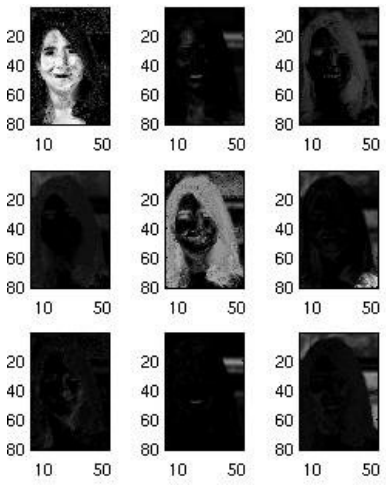


Figure 2: Proportion maps found using the C-FUMI algorithm for skin detection. The first image corresponds to the proportion map for the target skin endmember. Proportions range from 1 (white) to 0 (black).

top row of Figure 2. Experimentation with a range of true endmember prototypes will be conducted in future work. The target endmember was again the skin endmember. Training points containing 200 target and 200 non-target points were generated. The 200 target points were created by randomly generating mixtures of the three prototypes, including the target prototype. The 200 non-target points were generated using random mixtures of the two non-target prototypes. Fourhundred test points were also generated in the same fashion. The proportions for the target prototype for data points in the target class ranged from 4×10^{-4} to 0.88. Therefore, with the largest target proportion at 0.88, there were no pure target points in neither the test nor the training set. There were also no pure non-target data points in either set. The histograms of these proportions are shown in Figure 5. C-FUMI was applied to this training data with the following parameter settings: $\mu = 1 \times 10^{-4}$, $\Gamma = 2$, $M = 9$, and $\sigma^2 = 100$. The parameter values may seem ad-hoc, however, there is a logic to setting them. Future work can investigate joint parameter and prototype optimization. After initializing to 9 prototypes, the correct number of prototypes, 3, was estimated. The true and estimated prototypes are shown in Figure 3. The proportions associated with the target prototype for the test data estimated using the C-FUMI algorithm are shown in Figure 4. In order to perform detection, the proportions associated with the target prototype were thresholded. Points with proportions above the threshold were labeled as detected target points. Using a threshold value of 0.014 to ensure zero false alarms in the non-target points, 199 of 200 target

points (99.5%) were correctly detected. Therefore, even with very small proportions in the test data, accurate detection results were found.

In order to examine the accuracy of the estimated proportions for the target endmember, percentage error for the proportions were computed. Figure 6 plots the true proportions vs. percentage error. As shown in the figure, the percentage error is extremely flat (centered at 22% error) at a proportion value of 0.1 and greater. This indicates that the error is very stable for proportions exceeding 10%. The reasons for this stability is open for future investigation and may be a function of the quadratic programming solver being used.

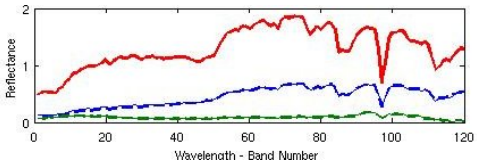


Figure 3: Comparison of true and estimated endmembers found using the C-FUMI algorithm. There are six plots in this figure. Estimated and true endmembers are extremely similar. The endmember in the center corresponds to the target.

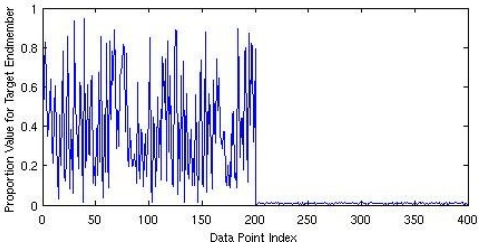


Figure 4: Plot of proportions estimated with the C-FUMI algorithm on simulated test data. The first 200 samples correspond to data containing some proportion of the target endmember with the rest corresponding to non-target data.

4. Discussion and Future Work

There are many interesting aspects of this work that require additional study. More thorough testing and training will be conducted. Also, incorporating piecewise convex representation and other alternative objective functions can be studied. Currently, the method learns a single target concept, future work can include extending this method into multiple target classes for classification. Additionally, the current method can be extended by modifying the objective function to encourage the target endmember to have a maximally unique spectral signature when compared to the non-target endmembers while maintaining a tight fit around the data

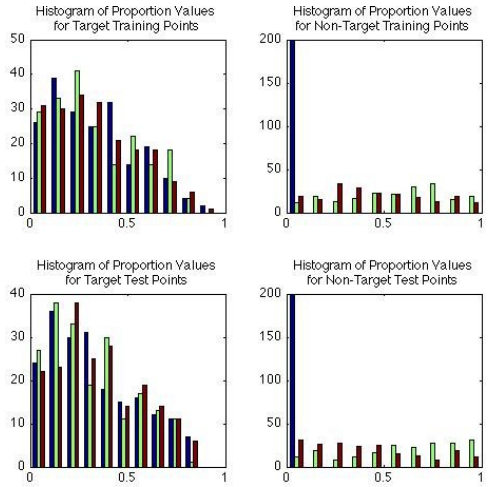


Figure 5: Histogram of proportions for training and test target and non-target points. The mean proportion value for target points is $\frac{1}{3}$.

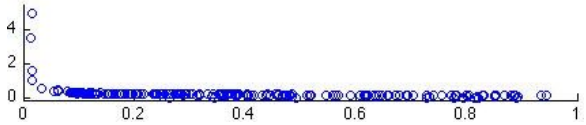


Figure 6: Percentage error of proportions estimated for target endmember on the simulated test data.

points. Currently, the proportions associated with each data point can be a string of small proportions. However, in many datasets, it is the case that data are associated with a small number of the endmembers. Therefore, sparsity promoting priors (such as Laplacian sparsity promotiong prior or priors encouraging minimum entropy) can be incorporated to encourage each data point to have non-zero proportions associated with a small number of endmembers.

References

[1] J. Broadwater and R. Chellappa. Hybrid detectors for subpixel targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1891–1903, Nov. 2007.

[2] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. *Aritificial Intelligence*, 89(1-2):31–17, 1997.

[3] O. Maron and T. Lozano-Perez. A framework for multiple-instance learning. *Neural Information Processing Systems*, 10, 1998.

[4] A. Zare and P. Gader. Sparsity promoting iterated constrained endmember detection for hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 4(3):446–450, July 2007.