Learning Topology of a Labeled Data Set with the Supervised Generative Gaussian Graph

Gaillard Pierre[1] , Aupetit Michaël[1] and Govaert Gérard[2]

1- French Atomic Energy Commission -
BP12 91680 Bruyères-le-Châtel - France
2- University of Technology of Compiegne - France

Abstract. Discovering the topology of a set of labeled data in a Euclidian space can help to design better decision systems. In this work, we propose a supervised generative model based on the Delaunay Graph of some prototypes representing the labeled data.

1 Introduction

Basic supervised learning problems involve a given set of M labeled training data $\{x_i, c_i | i = 1, ..., M\}$, where $x_i \in \mathbb{R}^D$ is a "feature" vector and $c_i \in \{1, ..., K\}$ is its associated class label. The ultimate goal of classification problems is to design a classifier which predicts the class of new feature vectors with a minimum error rate. However, prediction is only the last step of the learning process, which can be enriched through the exploratory analysis of the data, and specifically *the extraction of the topology of the classes*. Indeed, several topological characteristics of the classes could be useful among which: (1) *their connectedness* to evaluate the complexity of the classification problem [1]; (2) *their intrinsic dimension* to select relevant features [2]. In order to extract this topological information, we first assume that the data are drawn from some labeled *principal manifolds* [3] corrupted with some additive noise. One way to capture the structure of the data is to model their distribution in terms of *latent or hidden variables* [4]. The main generative models dealing with unsupervised manifold learning are *the Generative Topographic Mapping* [4] and *the Probabilistic Principal Component Analyzers* [5]. In the first approach, the intrinsic dimension is fixed a priori allowing visualization, while in the second approach, the intrinsic dimension is captured but the connectedness is lost. In order to overcome these limits, another generative model is proposed in [6] which is based on the Delaunay Graph (DG) of some prototypes representing the data. This model, called *Generative Gaussian Graph* (GGG), assumes no a priori about the topology and allows to learn the connectedness of sets of points. We propose to extend the GGG to the supervised case in order to extract the topology of the classes. We observe that the GGG can be viewed as a generalization of a Gaussian Mixture model (GM) and that the GM has been transposed to supervised learning [7]. Our approach uses the same path to extend the GGG to the supervised case.
Section 2 briefly reviews the GM and its supervised version as well as the GGG. In section 3, we introduce the new algorithm allowing to represent the topology of a labeled data set. Then we test it on artificial and real data in section 4, before the conclusion in section 5.

2 State of the art

2.1 The Gaussian Mixture

Mixture modeling can be regarded as a flexible way to represent a probability density function with a parametric model. A normal mixture density is defined by a finite weighted sum of Gaussian components having the following form : $p(x|\underline{\pi}, \underline{w}, \underline{\Sigma}) = \sum_{j=1}^{N} \pi_j g_j(x|w_j, \Sigma_j)$ where N is the number of components, g_j is a gaussian density with mean w_j and covariance matrix $\Sigma_j \in \underline{\Sigma}$. $\pi_j \in \underline{\pi}$ is the probability that an observation belongs to the j^{th} component such that $\pi_j \geq 0$ and $\sum_{j=1}^{M} \pi_j = 1$. This model can be viewed as a 2-step data-generating process: (step 1) drawing of the component j with a probability π_j; (step 2) drawing of the data following the density g_j of the component j. Therefore in this model, the principal manifold is assumed to be a set of points $\underline{w}$, having been corrupted with additive Gaussian noise g_j to lead to the observed data. In the context of supervised learning, Miller and Uyar [7] suggest to learn the allocation of the mixture components to the classes during the training. They introduce an additional parameter $\beta_{cj} \in \underline{\beta}$ to the GM, which represents the conditional probability of assigning the mixture component j to the class c. Moreover, since the components are common to the different classes, the model allows to represent easily a possible common structure for the different classes (e.g. high overlapping of the classes). The model, called Generalized Gaussian Mixture (GGM), takes the form: $p(x, c|\underline{\pi}, \underline{\beta}, \underline{w}, \underline{\Sigma}) = \sum_{j=1}^{N} \pi_j \beta_{cj} g(x|w_j, \Sigma_j)$ with the new constraints $\beta_{cj} \geq 0 \ \forall j, c$ and $\sum_{c=1}^{K} \beta_{cj} = 1 \ \forall j$.

2.2 The Generative Gaussian Graph

In this section, we use the same notations as provided in [6]. The GGG assumes that the data are generated by *some points and some segments* constituting the principal manifolds that have been corrupted with an additive spherical Gaussian noise with zero-mean and unknown variance σ^2. The underlying model is based on two Gaussian elements, namely the *Gaussian-points* and the *Gaussian-segments* which define a Gaussian mixture model. Given a set of N_0 prototypes $\underline{w}$ located over the data distribution using a vector quantization technique, the *Delaunay Graph* (DG) of the prototypes is constructed. With a weighted sum of N_0 vertices and the N_1 edges of the DG, convolved with an isotropic gaussian distribution with variance σ^2, the data generation process can be seen as a generalization of a usual gaussian mixture model and is defined by: $p(x_i|\underline{\pi}, \underline{w}, \sigma, DG) = \sum_{d=0}^{1} \sum_{j=1}^{N_d} \pi_j^d g^d(x_i|j, \sigma)$, where π_j^0 (resp. π_j^1) is the probability that a datum x_i was drawn from the Gaussian-point associated to w_j (resp. the Gaussian-segment associated to the j^{th} edge of DG). The density at point x_i involved by the j^{th} Gaussian-point and the j^{th} Gaussian-segment $[w_{a_j}, w_{b_j}]$ of length L_j are respectively defined as: $g^0(x_i|j, \sigma) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp(\frac{-(x_i - w_j)^2}{2\sigma^2})$ and $g^1(x_i|j, \sigma) = \frac{1}{L_j} \int_{w_{a_j}}^{w_{b_j}} g^0(x_i|w, \sigma) dw$.

3 The Supervised Generative Gaussian Graph

In this paper, the data are assumed to be generated by some points and some segments constituting the principal manifolds and then to be corrupted with an additive spherical Gaussian noise with zero-mean and unknown variance (*i.e.* $\underline{\Sigma} = \sigma$). Moreover, we assume that the j^{th} Gaussian element of dimension d (written (d, j)) can generate data from K different classes c with respective probabilities β_{cj}^d. Thus, we define the following model : $p(x, c|\underline{\pi}, \underline{\beta}, \underline{w}, \sigma, DG) = \sum_{d=0}^{1} \sum_{j=1}^{N_d} \pi_j^d \beta_{cj}^d g^d(x|j, \sigma)$, such that $\beta_{cj}^d \geq 0$, $\sum_{c=1}^{K} \beta_{cj}^d = 1 \; \forall j, \forall d$ and $\pi_j^d \geq 0$ and $\sum_{d=0}^{1} \sum_{j=1}^{M} \pi_j^d = 1$.

3.1 The four-step learning process

1. Initialization : Given a set of prototypes $\underline{w}$ located over the data distribution using a GGM with identical variances and the EM algorithm [7], the DG of the prototypes is constructed and defines the initial graph. Then each edge and each vertex of the graph is the basis of a generative model so that the graph generates a mixture of Gaussian density functions. The priors $\underline{\pi}$ are initialized to give equiprobability to each vertices and edges. The parameter $\underline{\beta}^0$ is initialized with the value obtained by the GGM while each component of $\underline{\beta}^1$ is set to $\frac{1}{K}$. Finally, we initialize σ with the noise value obtained by the GGM.

2. Learning of the parameters : The learning objective was chosen to be the joint likelihood over the observed labeled data. This measure of quality *wrt* the parameters of the model is defined as: $L = \prod_{i=1}^{M} p(x_i, c_i|\underline{\pi}, \underline{\beta}, \underline{w}, \sigma, DG)$. In order to maximize the likelihood we use the EM algorithm. The EM algorithm consists in t_{max} iterative steps updating $\underline{\pi}$, $\underline{\beta}$, σ which ensure the increase of the likelihood. The updating rules take into account the constraints about positivity or sum to unity of the parameters:

$$\begin{array}{ll} \pi_j^{d[\text{new}]} & = \frac{1}{M} \sum_{i=1}^{M} p(d, j|x_i, c_i) \\ \sigma^{2[\text{new}]} & = \frac{1}{DM} \sum_{i=1}^{M} [\sum_{j=1}^{N_0} p(0, j|x_i, c_i)(x_j - w_i)^2 \\ & + \sum_{j=1}^{N_1} p(1, j|x_i, c_i) \frac{(2\pi\sigma^2)^{-D/2} \exp(-\frac{(x_i - q_j^i)^2}{2\sigma^2})(I_1[(x_i - q_j^i)^2 + \sigma^2] + I_2)}{L_j \cdot g^1(x_i|j, \sigma)}] \\ \beta_{cj}^{d[\text{new}]} & = \sum_{i=1:c_i=c}^{M} p(d, j|x_i, c_i) / \sum_{i=1}^{M} p(d, j|x_i, c_i) \end{array} \quad (1)$$

with $I_2 = \sigma^2 \left((Q_j^i - L_j) \exp(-\frac{(Q_j^i - L_j)^2}{2\sigma^2}) - Q_j^i \exp(-\frac{(Q_j^i)^2}{2\sigma^2}) \right)$,
$I_1 = \sigma \sqrt{\frac{\pi}{2}} (\text{erf}(\frac{Q_j^i}{\sigma\sqrt{2}}) - \text{erf}(\frac{Q_j^i - L_j}{\sigma\sqrt{2}}))$, where $Q_j^i = \frac{\langle x_i - w_{a_j} | w_{b_j} - w_{a_j} \rangle}{L_j}$,
$q_j^i = w_{a_j} + (w_{b_j} - w_{a_j}) \frac{Q_j^i}{L_j}$ and $p(d, j|x_i, c_i) = \frac{\pi_j^d \beta_{c_i j} g^d(x_i|j, \sigma)}{p(x_i, c_i|\underline{\pi}, \underline{\beta}, \underline{w}, \sigma, DG)}$ is the posterior probability that the datum x_i was generated by the component (d, j).

3. Prunning : Finally, to get the supervised topology representing graph from the generative model, we prune from the initial DG the edges for which there is no chance they generated the data, *i.e.* edges having a null or an almost

null prior at the end of the learning process: $\pi_j^1 < \epsilon$. At that point, the edges represent the connectedness of the joint density of all the classes.

4. Model selection : In statistical inference from data, selecting a parsimonious model among a collection of models is an important but difficult task [8]. The complexity of the Supervised Generative Gaussian Graph (SGGG) is defined by its number of vertices and edges. Since the complexity of our model is closely related to the number of prototypes, we choose the best GGM in the sense of the Bayesian Information Criterion (BIC) [8] to build the initial DG. Thus we select the GGM $\mathcal{M}$ with N_0 components which maximises : $BIC(\mathcal{M}) = \prod_{i=1}^{M} p(x_i, c_i|\underline{\pi}_{\mathcal{M}}, \underline{\beta}_{\mathcal{M}}, \underline{w}_{\mathcal{M}}, \sigma_{\mathcal{M}}) - \frac{v_{\mathcal{M}}}{2} log(M)$ where $v_{\mathcal{M}}$ is the number of free parameters of $\mathcal{M}$: $v_{\mathcal{M}} = N_0.(K + D)$

4 Experiments

We drawn a 2-D data sample from a set of class-manifolds : two quarter-circles, one 'Y-shaped manifold' and one point with respective probabilities $\{0.2; 0.2; 0.5; 0.1\}$ and $\underline{\beta} = \{(1, 0), (0.5, 0.5), (0, 1), (1, 0)\}$. The observed data are obtained with an additive gaussian noise with mean 0 and variance σ^2 and are represented in the figure 1 (a). We use an artificial data-base from which we know the topology in order to verify the validity of the model.
For all the experiments we use the same parameter values : $t_{max} = 100$, $\epsilon = 0.01$. Figure 1 (b) shows that the SGGG allows recovering the topology of the four manifolds *wrt* the class-label while the GMM (figure 1 (d)) does not give us any insight about the connectedness of the classes. Moreover, the SGGG informs us about the class-manifold overlapping thanks to the paramater $\underline{\beta}$: a manifold overlapping of different classes is caracterized by $\max_c(\beta_{cj}^d) \neq 1$.
Figure 2 describes an experiment where we want to verify the ability of the SGGG to learn the topology of a labeled data set in various noise conditions. We drawn 30 different training sets for several variances of the noise. We use the learning process described in section 3 to build the SGGG and we extract the topological characteristics of the model (number of connected components, degree of the vertices). Then, we compare the topology of the model with the original topology of the set of manifolds : for example, we check that the part of the model representing the 'Y-shaped manifold' is a set of connected single-class edges (*i.e.* $\max_c(\beta_{cj}^1) = 1$) such that two vertices have a degree equal to 1 (the extreme points of the 'Y'), one has a degree equal to 3 (the crossing of the 'Y') and all the other vertices have a degree equal to 2. We observe (figure 2) that the model can recover the good topology *wrt* to the classes for the quarter-circles and the point even with high noise variance. However, the 'Y-shaped' is often modeled by a 'V-shaped' when the noise variance increases. With noisy data, the accuracy of the model decreases but it is still relatively robust.
We also tested the SGGG on real data. Figure 3 represents the natural and artificial seismic events in France in 2000 and the resulting SGGG.