ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

Gabriel Graph for Dataset Structure and Large Margin Classification: A Bayesian Approach

Luiz C.B Torres and Cristiano L. Castro and Antônio P. Braga *

Federal University of Minas Gerais Department of Electronics Engineering Av. Antonio Carlos, 6627, Pampulha 30161-970, Belo Horizonte, MG, Brazil

Abstract. This paper presents a geometrical approach for obtaining large margin classifiers. The method aims at exploring the geometrical properties of the dataset from the structure of a Gabriel graph, which represents pattern relations according to a given distance metric, such as the Euclidean distance. Once the graph is generated, geometric vectors, analogous to SVM's support vectors are obtained in order to yield the final large margin solution from a Gaussian mixture model approach. Preliminary experiments have shown that the solutions obtained with the proposed method are close to those obtained with SVMs.

1 Introduction

Little has been done in the literature to explore the concept of margin-based classifiers [1] from a geometrical perspective. The notion of a separation margin is in fact quite intuitive, however, the problem has been often formulated from a numerical perspective. In SVM's design the contribution of each support vector to the final classification outcome is given by the magnitude of the yielded Lagrange Multipliers [2]. The Quadratic Programming formulation of SVMs is optimal for preestablished values of kernel and regularization parameters so, by adopting approaches such as cross-validation a high performance solution may be achieved. Nevertheless, a geometrical formulation may uncover some inner properties of the dataset and provide insights into the problem that were not possible to be identified with a numerical approach.

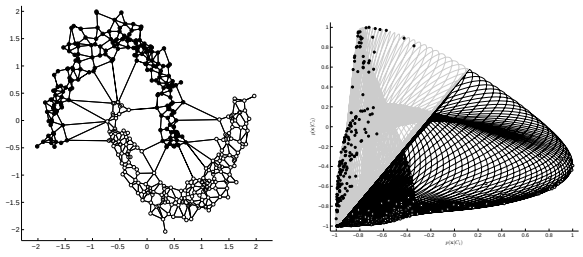
A geometrical formulation of a discriminative margin-based classifier is presented in this paper. The pairwise distances between input patterns are computed and used to build a graph model of the dataset (Gabriel Graph [3]) which, when associated with label information, allows the identification of the margin Geometrical Vectors ($\mathcal{GV}$). This set is similar to the edited set found in [4]. The computational cost to calculate the distance matrix to build the Gabriel Graph ($\ddot{G}$) has the same order of complexity of kernel construction. Only the $\mathcal{GV}$, which are analogous to SVM's Support Vectors ($\mathcal{SV}$), are combined as a mixture model of a Bayesian classifier to compute the final classification. Although there is no implicit mapping into a feature space like in SVM description, there is in fact an explicit mapping to the likelihoods space of each class (as can be seen in Fig. 1(b)), where the final classification decision is accomplished.

The structure of the present paper is as follows. The Gabriel Graph is presented in Section 2. Section 3 presents the geometrical Gaussian classifier

*This work has been supported by the Brazilian agencies CNPq and CAPES.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

method. An analogy between the method proposed and SVM classifiers is presented in Section 3.1. Finally results and conclusions are presented in Sections 4 and 5.



(a) (b)

Fig. 1: (a) Gabriel graph obtained from a binary classification problem (Two moons). (b) Mapping of Two-Moon dataset to the Likelihood space.

2 Gabriel Graph

Likewise SVM the proposed method classifies an input pattern according to a subset of the training set [5]. The selected points, hereafter called Geometric Vectors ($\mathcal{GV}$), are chosen from a convex graph called *Gabriel graph* [3], that will be described next.

2.1 Definition

Considering the dataset $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{C_1, C_2\}$, the Gabriel graph $\ddot{G}$ of $\mathcal{S}$ is defined as the graph with a set of vertices $\mathcal{V} = \{\mathbf{x}_i\}_{i=1}^{N}$ and edges $\mathcal{E}$ that meet the following definition:

> An edge connecting the vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ from $\mathcal{V}$ belongs to $\mathcal{E}$ only, and only if
>
> $$\delta^2(\mathbf{x}_i, \mathbf{x}_j) \leq \left[\delta^2(\mathbf{x}_i, \mathbf{x}_k) + \delta^2(\mathbf{x}_j, \mathbf{x}_k)\right], \quad (1)$$
>
> $\forall\, \mathbf{x}_k \in V$ and $i \neq j \neq k$, where $\delta(\cdot,\cdot)$ is the Euclidean distance between the vertices.

Fig. 1(a) shows an example of graph resulting from the previous definition.

3 Geometric Gaussian Classifier

Classification is accomplished in two distinct phases. The first one aims at finding the patterns located in the class separation region, which are called here Geometrical Vectors ($\mathcal{GV}$). This is carried on by identifying the edges of a Gabriel

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Graph that have patterns from different classes in their vertices. As mentioned earlier, the importance of $\mathcal{GV}$ for our method is similar to the support vectors of SVMs. In the second phase, $\mathcal{GV}$ samples are used to compute the parameters of Gaussian mixture models for each class and then a Bayesian classification rule is applied. Each one of the two phases will be described in details next.

- **Phase 1. Identifying the Geometrical Vectors**

  1. *Gabriel Graph.* Obtain the Gabriel Graph $\ddot{G} = \mathcal{V}, \mathcal{E}$ from the training set $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ as described in Section 2.1.
  2. *Eliminate overlapping.* For each $\mathbf{x}_i \in \mathcal{V}$, analyze the subgraph induced by the vertex $\mathbf{x}_i$, i.e, the subgraph formed by the edges that have $\mathbf{x}_i$ as one of the ends. If most of the neighbors (adjacent vertices of $\mathbf{x}_i$) belong to the opposite class, then $\mathbf{x}_i$ is considered as noise and should be eliminated of $\mathcal{V}$.
  3. *Geometrical vectors.* Select all edges that have vertices belonging to distinct classes, i.e., select $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}$, such that $y_i \neq y_j$. The Geometrical Vectors $\mathcal{GV}$ corresponds to the patterns at the extremes of these border edges.

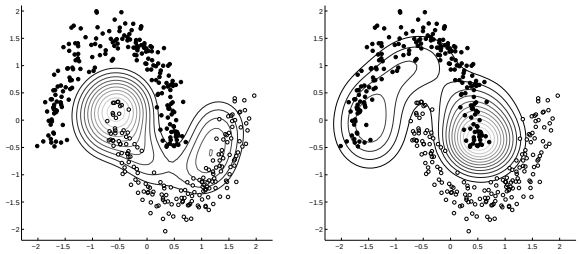
- **Phase 2. Computing Density Mixture**

  1. *Multivariate Gaussians.* For each pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{GV}$, such that $y_i \neq y_j$, obtain Gaussian densities $G_\tau(\mathbf{x}, \mu_\tau, \mathbf{\Sigma}_\tau)$, with mean vectors $\mu_\tau = \mathbf{x}_\tau$ and diagonal covariance matrices $\mathbf{\Sigma}_\tau$, for $\tau = i, j$. Each diagonal element of $\mathbf{\Sigma}_\tau$ is computed as $\sigma_\tau^2(l) = 2 \cdot \|\mathbf{x}_i(l) - \mathbf{mp}(l)\|$ (variance for the l-th dimension), where $\mathbf{mp}$ is the mid-point vector between the two vertices $\mathbf{x}_i$ and $\mathbf{x}_j$.
  2. *Density Mixtures per class.* Compute a density mixture model for each class C_k from the weighted sum of the Gaussian functions whose centers belonging to the same class, i.e.,

$$p(\mathbf{x}, \theta_k | C_k) = \sum_{j=1}^{N_k} w_j G_j(\mathbf{x}, \mu_j, \mathbf{\Sigma}_j), \text{ for } k = 1, 2 \qquad (2)$$

  where $\theta_k = [\{\mu_1, \mathbf{\Sigma}_1\}, \ldots, \{\mu_{N_k}, \mathbf{\Sigma}_{Nk}\}]$ is the parameter vector drawn from the N_k geometrical vectors of class C_k, w_j is the corresponding weight for the j-th density $G_j(\cdot)$, subject to $\sum_{j=1}^{N_k} w_j = 1$. Figs. 2(a) and 2(b) illustrates the mixture models $p(\mathbf{x}, \theta_1 | C_1)$ and $p(\mathbf{x}, \theta_2 | C_2)$ for the toy-problem *Two Moons*.
  3. *Decision Rule.* A Bayes decision rule is then formulated from the class mixture models, described by the parameter vectors θ_1 and θ_2. This rule is the one that minimizes the global error probability, such that the classification of an arbitrary pattern $\mathbf{x}$ is given by

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.



(a) (b)

Fig. 2: (a) density mixture for class C_1. (b) density mixture for class C_2.

$$f(\mathbf{x}) = \begin{cases} C_1 & \text{if } \frac{p(\mathbf{x},\theta_1|C_1)}{p(\mathbf{x},\theta_2|C_2)} \geq \frac{P(C_2)}{P(C_1)} \\ C_2 & \text{Otherwise.} \end{cases} \tag{3}$$

3.1 Analogy with SVMs

SVM's final classification of an input pattern $\mathbf{x}_i$ is obtained as $f(\mathbf{x}_i) = sign(\sum_j y_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j))$, which is in fact the sign of a weighted sum of the labels y_j. Although the sum is accomplished over all N training patterns, only the terms associated to the SVs, which have non-zero α_j (Lagrange Multiplier), are in fact computed. If the magnitude of the positive terms ($y_j = +1$) dominate the sum then the outcome is positive ($y_i = +1$); otherwise, if the negative terms dominate the sum ($y_j = -1$) then the outcome is negative ($y_i = -1$). Equation 4 shows SVM's classification rule with the positive and negative summation terms separated and label values assigned.

$$f(\mathbf{x}_i) = sign(\overbrace{\sum_{j=1}^{N_1} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)}^{\text{Positive Class}} - \overbrace{\sum_{l=1}^{N_2} \alpha_l K(\mathbf{x}_i, \mathbf{x}_l)}^{\text{Negative Class}}) \tag{4}$$

The method presented in this paper has an analogous classification rule, since the general Bayes classification rule of Equation 3 can be rewritten as $f(\mathbf{x}_i) = sign(p(\mathbf{x}_i, \theta_1|C_1) - \frac{P(C_2)}{P(C_1)} p(\mathbf{x}_i, \theta_2|C_2)$. Since the likelihoods $p(\mathbf{x}, \theta_1|C_1)$ and $p(\mathbf{x}, \theta_2|C_2)$ are described here as mixtures of Gaussian densities, or kernel functions centered in $\mathcal{GV}$, the general classification rule can be rewritten as follows

$$f(\mathbf{x}_i) = sign(\overbrace{\sum_{j=1}^{N_1} w_j K(\mathbf{x}_i, \mathbf{x}_j)}^{\text{Positive Class}} - \overbrace{\frac{N_1}{N_2} \sum_{l=1}^{N_2} w_l K(\mathbf{x}_i, \mathbf{x}_l)}^{\text{Negative Class}}) \tag{5}$$

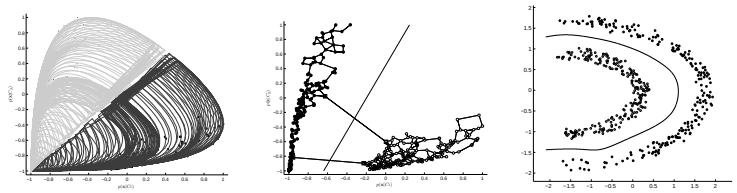
ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Since for each geometrical vector from one class there is a corresponding one for the other class, $N_1 = N_2$ and the two classification rules differ only on the way the mixing parameters α_j and w_j are computed. In the experiments presented in this paper $w_j = 1$ in all trials.

4 Preliminary Results

4.1 Toy problems

In order to illustrate the large margin solution yielded from the proposed method, experiments were conducted with a toy problem named Half Kernel. Some results are shown from Figs. 3(a) to 3(c). It is worth noting that our method is able to construct good solutions without the need of setting any prior parameter.



(a) (b) (c)

Fig. 3: Construction of the Large Margin Solution for *Half Kernel* dataset: (a) mapping to the Likelihood space; (b) solution obtained from Equation 3; (c) separation surface in the input space;

4.2 Real-World Data Sets

Experiments were also performed with 4 real-word datasets drawn from the UCI repository [6]: the Stalog Australian Credit (acr), the Wisconsin breast cancer (wbc), the Pima Indians diabetes (pid) and the Stalog heart disease (hea). All these datasets had their attributes normalized to mean 0 and standard deviation 1. Twenty different cases were generated for each dataset by shuffling the original indexes of its elements. Then, each case was split into training (2/3) and test (1/3) subsets in a stratified manner.

Table 1 shows the characteristics of each dataset along with the results obtained in terms of accuracy (mean) and its standard deviation. The symbols N_{tr} and N_{te} correspond to the number of patterns used for training and test, respectively and N_d is the total number of attributes. In order to provide some insight of the effectiveness of our method, the results were compared with the well-known benchmark of Least-Squares SVMs (LS-SVM) extracted from [7]. In that study, the regularization and kernel parameters of LS-SVMs were selected via 10-fold cross-validation. The kernel which reported best results was RBF kernel.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Table 1: Results from the UCI datasets

	acr	**wbc**	**pid**	**hea**
N_{tr}/N_{te}	462/165	456/159	512/157	182/60
N_d	14	9	8	13
LMG	86.6 ±2.3	96.6 ±1.0	77.6 ±2.6	82.8 ±5.0
LS-SVM*	87.0 ±2.1	96.4 ±1.0	76.8 ±1.7	84.7 ±4.8

5 Conclusion

In SVM's solution margin maximization and dataset fitting problems are an outcome of a quadratic programming setting. The method proposed in this paper considers explicitly the geometrical properties of the dataset and the margin definition, without aiming at the optimization problem itself. The dataset geometry in relation to the separation margin is obtained from the structure of a Gabriel graph, that represents within-class and between-class pattern relations. The geometric vectors obtained from such a structure are analogous to SVM's support vectors and yield a large margin solution, which is obtained directly with a mixture model of a Bayesian classifier and without the need of setting any parameter.

The preliminary results obtained via both methods were close, nevertheless, in contrast with SVMs, our method provides a way to represent an n-dimensional problem in a $\mathbb{R}^2$ space, which was named here as "likelihood space", so that the problem becomes more representative and less complex. The geometrical solution presented in this paper uncovers the inherent properties of a large margin classification problem, since margin patterns are obtained in a more intuitive way than the usual approach.

References

[1] Alexander J Smola. *Advances in large margin classifiers*. MIT press, 2000.

[2] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[3] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, second edition, 2000.

[4] Wan Zhang and Irwin King. A study of the relationship between support vector machine and gabriel graph. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 1, pages 239–244. IEEE, 2002.

[5] Kristin P Bennett and Erin J Bredensteiner. Duality and geometry in svm classifiers. In *ICML*, pages 57–64, 2000.

[6] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[7] Tony Van Gestel, Johan AK Suykens, Bart Baesens, Stijn Viaene, Jan Vanthienen, Guido Dedene, Bart De Moor, and Joos Vandewalle. Benchmarking least squares support vector machine classifiers. *Machine Learning*, 54(1):5–32, 2004.