Semi-Supervised Active Learning in Graphical Domains

Augusto Pucci, Marco Gori and Marco Maggini

Dipartimento di Ingegneria dell'Informazione
Via Roma 56, 53100 Siena (ITALY)
{augusto,marco,maggini}@dii.unisi.it

In a traditional machine learning task, the goal is training a classifier using only labeled data (data feature/label pairs) in order to be able to generalize on completely new data to be labeled by the classifier. Unluckily in many cases it is difficult, expensive or time consuming to obtain the labeled instances needed for training, also because we usually require a human supervisor to annotate lots of data to collect a significant training set. Moreover, in many cases we are not interested in generalization to any unseen example, but we just require to discover labels for a large quantity of unlabeled, but already available, data by using a small subset of labeled data. If the given scenario involves both these conditions, a semi-supervised learning algorithm can be exploited as a solution for the classification problem. Semi-supervised learning algorithms combine a large amount of unlabeled data and a available small set of labeled data, to build a reliable classifier. It is particularly interesting to focus on a sub-class of semi-supervised learning algorithms, that is graph-based semi-supervised learning. In this framework we represent data as a graph where the nodes represent the labeled and unlabeled examples in the dataset, and the edges are added according to a given similarity relationship between pairs of examples. A common feature of every graph-based method is the fact they are nonparametric, discriminative and transductive. However, a crucial issue is the very limited number of labeled (supervised) data points we have with respect to unlabeled points, so it is essential to have representative examples. In some cases the labeled data points are given, but there are also many scenarios where we only have a set of unlabeled data points and we can choose a limited number of them to built the labeled data set. In this paper we propose a graph-based semi-supervised active learning algorithm based on a reasonable choice for labeled data points, in oder to improve classification accuracy. In a typical semi-supervised learning problem we should consider a set of n data points $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ defined on a d dimensional feature space, such that each point $x_i \in I\!R^d$, and a set of labels $\mathcal{Y} = \{+1, -1\}$. We have an oracle function h which gives us the correct labelling for every data point we applied it to. So the oracle is a function defined on the data points as $h : X \to \mathcal{Y}$. Unluckily we cannot invoke the oracle on every point in X, but we have a limited number of points we can apply h to. This budget is l and we call $L \subset X$ the set of points we supervise, the remaining points are unlabeled, we refer to that set as $\bar{L} = X \setminus L$. The goal of a semi-supervised learning algorithm is to find the right labelling for the unlabeled points. The strength of the relationship among pairs of data points can be described by introducing a correlation function, as for example $w([i,j]) = e^{-\lambda \cdot \|x_i - x_j\|}$ where λ is a positive real number and $\|\cdot\|$ is any valid norm (for example the euclidean norm). Note that $w([i,j]) \in (0,1]$. Now it is possible to model the learning problem from a graph based perspective. In fact, we can build a graph $G = \{V, E\}$ where the vertex set V contains the objects in X and the set of the edges E is obtained by adding the edge $[i,j]$ connecting the two nodes corresponding to the objects x_i and x_j if and only if their correlation is above a fixed threshold ϵ, i. e. $w([i,j]) > \epsilon$. Moreover the edges in E are weighted and $w([i,j])$ will be the weight of the edge $[i,j]$. We consider G to be undirected, so if an edge $[i,j]$ connects the nodes i and j, the set E contains also the edge $[j,i]$. In the following, we use notation $i \sim j$ to refer the set of nodes in V that are adjacent to node j, we also introduce the "node grade" function computed as $g(v) = \sum_{u \sim v} w([u,v])$. In this graphical setting the oracle function has to be applied to nodes corresponding to data points, so we redefine it as $h : V \to \mathcal{Y}$ and L will be the set of labeled nodes we applied h to (thus $\bar{L}$ is the set of unlabeled nodes). We can model the fact that a subset L of nodes in V are labeled according to oracle function h by introducing the supervision function y_L defined as:

$$y_L(v) = \begin{cases} h(v) & \text{if } v \in L \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Thus, a graph-based semi-supervised learning algorithm essentially consist in spreading a "labelling function" $y_L(v)$ from the labeled nodes to the unlabeled nodes according to the correlation function and in a "smart" way, so that it is possible to find a classification function φ defined on both labeled and unlabeled nodes. In this paper we adopt the learning algorithm described in [2], but the proposed active learning technique can be applied to any graph-based semi-supervised learning algorithm which makes use of a spreading mechanism. In [2] the authors propose an effective algorithm to estimate the node labelling φ by a Laplacian regularization, they reduce the learning problem to an iterative equation:

$$\varphi_L^{t+1}(v) = \sum_{u \sim v} \left(\alpha \cdot \frac{w([u,v])}{\sqrt{g(u)g(v)}} \varphi_L^t(u) \right) + (1-\alpha) \cdot y_L(v)\ . \tag{2}$$

where $\alpha \in (0,1)$ is a bias parameter that can be used to balance the contribution of each term. This iterative formulation is also interesting because it provides an alternative view to interpret the regularization process. In fact if we look closer to the iterative equation we notice a diffusion process starting from labeled points in L and driven by the weight function w (after the normalization $w([u,v])/\sqrt{g(u)g(v)}$). This process is very close to the computation of biased PageRank ([1]), where y_L is the bias vector and $w([u,v])/\sqrt{g(u)g(v)}$ is entry u,v for the transfer matrix. In a matrix form 2 can be rewritten as:

$$\varphi_L = \alpha \cdot M \cdot \varphi_L + (1-\alpha) \cdot y_L \tag{3}$$

where entry (i, j) for matrix M is:

$$m_{uv} = \sqrt{\frac{w([u,v])}{g(u)} \cdot \frac{w([v,u])}{g(v)}} = \frac{w([u,v])}{\sqrt{g(u)g(v)}}$$

Note that M is symmetric, so $m_{uv} = m_{vu}$. After simple computation on 3 we can obtain $\varphi_L = \bar{M} \cdot y_L$ where $\bar{M} = \left(\sum_{t=0}^{T} \alpha^t \cdot M^t\right) \cdot (1-\alpha)$ with $T \to \infty$. So we can reduce the diffusion process to a simple linear projection by matrix $\bar{M}$ of supervision vector y_L. The resulting labelling function φ_L is just a linear combination of l supervised columns in $\bar{M}$ corresponding to nodes in L. All these columns give a positive contribution if they correspond to a positive node and a negative contribution if they correspond to a negative node. Matrix $\bar{M}$ defines an "influential profile" for every node in V, in fact element $\bar{m_{ij}}$ (that is exactly the same as $\bar{m_{ji}}$ due to the symmetry of matrix $\bar{M}$) measure the influence a node i has on node j and viceversa. We denote the i-th column (or equivalently row) of matrix $\bar{M}$ as $\bar{m_i}$ and it will be the influential profile for node i. So far we assumed L is given, on the other hand is quite obvious that the labelling function φ_L accuracy depends a lot on a informative set L. There are many applicative scenarios where L is not given and we only have a set of nodes unlabeled nodes V and a limited budget l to invoke oracle function h in order to build set L. In this section we introduce some criterions to choose a good labeled node set L in order to make it as informative as possible. So we suppose our set of labeled points is initially empty, that is $L_0 = \emptyset$ and we need to choose one after another which data point we want to apply the oracle function to. At every decision step we build L_e from $L_{(e-1)}$ by adding a node k, so $L_e = \{L_{(e-1)} \cup k\}$. A good choice for k should maximise three parameters: "influence degree". "absolute innovation degree" and "relative innovation degree".

Influence degree for a node k measures the impact of this node on the other nodes, if it is low it means $\varphi_{L_{(e-1)}}$ would not be changed a lot in case we add k to $L_{(e-1)}$, on the opposite if influence degree is high, the presence of k in L can makes a huge difference for the classification. We measure it with respect to influence profile $\bar{m_k}$ norm as $\|\bar{m_k}\|^2 =< \bar{m_k}, \bar{m_k} >$.

Absolute innovation degree for a node k measures the variety of information in k. While influence degree measures the strength of k influence, absolute innovation degree will be high if affects many other nodes and with a very variable intensity for every node. We measure it as $(1 - \cos\gamma_k)$, where we compute $\cos\gamma_k$ as $\cos\gamma_k = \frac{<\bar{m_k}, \mathbf{1}>}{\|\bar{m_k}\| \cdot \|\mathbf{1}\|}$ here $\mathbf{1}$ is a vector whose components are all 1.

Relative innovation degree for a node k measures the added information we obtain if we add k to $L_{(e-1)}$ with respect to other data points already present in $L_{(e-1)}$. The general idea is we wish node k profile vector were as independent as possible from other data points, otherwise it would be just a linear combination of them, but we also need to consider the kind of contribution (negative or positive). We measure it as $\sum_{i \in L_{(e-1)}} (1 - \cos\gamma_{ki})$ where we compute $\cos\gamma_{ki}$ as $\cos\gamma_{ki} = \frac{<+/-\bar{m_k}, h(i) \cdot \bar{m_i}>}{\|\bar{m_k}\| \cdot \|\bar{m_i}\|}$. The problem here is the fact we do not know the sign contribution coming from node k, so we need to consider both cases, so we have for the positive case $\sum_{i \in L_{(e-1)}} (1 - \cos \gamma_{k+i})$ where $\cos \gamma_{k+i} = \frac{<+\bar{m_k}, h(i) \cdot \bar{m_i}>}{\|\bar{m_k}\| \cdot \|\bar{m_i}\|}$ and also, for the negative $\sum_{i \in L_{(e-1)}} (1 - \cos \gamma_{k-i})$ where $\cos \gamma_{k-i} = \frac{<-\bar{m_k}, h(i) \cdot \bar{m_i}>}{\|\bar{m_k}\| \cdot \|\bar{m_i}\|}$. Now we can combine the three criterions we introduced in order to define the "information function" associated with a node $k \in \bar{L}$ given the current labeled node set L:

$$\delta_L(k) = \|\bar{m_k}\|^2 + \mu^0(p_k^0) \cdot (1 - \cos \gamma_k) + \mu^+(p_k^+) \cdot \sum_{i \in L} (1 - \cos \gamma_{k+i}) + \mu^-(p_k^-) \cdot \sum_{i \in L} (1 - \cos \gamma_{k-i}) \tag{4}$$

where μ functions (μ^0, μ^+ and μ^-) are monotonic growing functions defined on $\mu : V \to \infty$, while we use some pseudo-probabilities to consider the certainly we have for the sign of k node label given the information we have so far in the current L. In particular we use $g_{\bar{M}}(i) = \sum_{j=1}^n \bar{m_{ij}}$, $g_{\bar{M}}^+(i) = \sum_{j=1}^n s(y_L(j)) \cdot \bar{m_{ij}}$, $g_{\bar{M}}^-(i) = \sum_{j=1}^n s(-y_L(j)) \cdot \bar{m_{ij}}$ and $g_{\bar{M}}^0(i) = g_{\bar{M}}(i) - \left(g_{\bar{M}}^+(i) + g_{\bar{M}}^-(i)\right)$ where $s(\cdot)$ is the step function $s(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$. Now we can compute pseudo-probabilities as $p_k^0 = \frac{g_{\bar{M}}^0(i)}{g_{\bar{M}}(i)}$, $p_k^+ = \frac{g_{\bar{M}}^+(i)}{g_{\bar{M}}(i)}$ and $p_k^- = \frac{g_{\bar{M}}^-(i)}{g_{\bar{M}}(i)}$. Note that pseudo-probabilities p_k^0, p_k^+ and p_k^- depends on the current choice for L ($L_{(e-1)}$ for example), so they need to be re-computed after every update of labeled node set. Finally we can use information function 4 to built the labeled node set. At every step e we choose a new node k to be added to set $L_{(e-1)}$ taken from set $\bar{L}_{(e-1)}$ in order to maximise function $\delta_{L_{(e-1)}}(k)$, so we find $k = \arg\min_{i \in \bar{L}_{(e-1)}} \delta_{L_{(e-1)}}(i)$, then we update L as $L_{(e-1)} = \{L_e \cup k\}$, we need to repeat this procedure until $e = l$ and we finished our budget for oracle function invocation.

References

[1] L. Page, S. Brin, R. Motwani, and T. Winograd, *The pagerank citation ranking: Bringing order to the web*, Technical report, Stanford University, 1998.

[2] D. Zhou and B. Schölkopf, *Regularization on Discrete Spaces*, Pattern Recognition, Proceedings of the 27th DAGM Symposium, 361-368, Springer, Berlin, Germany, 2005.