LAGO on the Unit Sphere

Alexandra Laflamme-Sanders and Mu Zhu
University of Waterloo
200 University Ave. W.
Waterloo, Ontario, Canada N2L 3G1
{alaflamme,m3zhu}@uwaterloo.ca
http://www.math.uwaterloo.ca/~m3zhu/

We study the rare target detection problem, that is, a two-class classification problem in which the class of interest (C_1) is very rare; most observations belong to a majority, background class (C_0). Given a set of unlabelled observations, the goal is to *rank* those belonging to C_1 ahead of the rest. Clearly, one can use any classifier to do this as long as the classifier is capable of producing an estimated posterior probability $P(y \in C_1|\mathbf{x})$ or a classification score, e.g., the support vector machine (SVM). Since its emergence, the SVM has spawned a wave of new research in kernel-based methods. If radial-basis kernel functions are used, the final decision function constructed by the SVM can be written as

$$f(\mathbf{x}) = \sum_{i \in SV} \alpha_i y_i \phi(\mathbf{x}; \mathbf{x}_i, r\mathbf{I}) + \beta_0, \tag{1}$$

where $\phi(\mathbf{x}; \mathbf{x}_i, r\mathbf{I})$ is a radial-basis kernel function centered at $\mathbf{x}_i$ with radius r, and SV denotes the set of "support vectors." For ranking purposes, the constant term β_0 can be dropped.

LAGO [4] is an efficient kernel method designed specifically for the rare target detection problem. The decision function constructed by LAGO for ranking unlabelled observations can be written as

$$f(\mathbf{x}) = \sum_{\mathbf{x}_i \in C_1} |\mathbf{R}_i| \phi(\mathbf{x}; \mathbf{x}_i, \alpha\mathbf{R}_i), \quad \mathbf{R}_i = r_i\mathbf{I}, \tag{2}$$

where r_i is the average distance between the kernel center, $\mathbf{x}_i \in C_1$, and its K-nearest neighbors from C_0, i.e.,

$$r_i = \frac{1}{K} \sum_{\mathbf{w} \in N_0(\mathbf{x}_i, K)} d(\mathbf{x}_i, \mathbf{w}). \tag{3}$$

The notation "$N_0(\mathbf{x}_i, K)$" denotes the K-nearest neighbors of $\mathbf{x}_i$ from C_0; and $d(\mathbf{u}, \mathbf{v})$ is a distance function, e.g., $d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|$. The parameters α and K are global tuning parameters.

Hence, (2) has exactly the same form as (1), but it is constructed in an efficient manner that fully exploits the special nature of the rare class detection problem. Instead of using an iterative optimization procedure to identify support vectors and calculate the coefficients, α_i ($i = 1, 2, ..., n$), LAGO simply uses all training observations from the rare class, C_1, as its "support vectors" and sets the coefficient in front of each kernel function to be $|\mathbf{R}_i|$, the volume of the kernel. The only calculation required is the computation of r_i — equation (3) — for every $\mathbf{x}_i \in C_1$. This is very efficient since the size of C_1 is typically very small for rare target problems.

Zhu *et al.* [4] gave a few theoretical arguments for why all these shortcuts are justified. Suppose $p_1(\mathbf{x})$ and $p_0(\mathbf{x})$ are density functions of C_1 and C_0. The main argument is that (2) can be viewed as a kernel density estimate of p_1 adjusted locally by a factor that is approximately inversely proportional to p_0, i.e., $|\mathbf{R}_i|$. The resulting ranking function $f(\mathbf{x})$ is thus approximately a monotonic transformation of the posterior probability that item $\mathbf{x}$ belongs to the rare class. Intuitively, the "LAGO principle" can be summarized as follows: To evaluate a new observation $\mathbf{x}$, each training observation $\mathbf{x}_i \in C_1$ will cast a vote, and its vote will be weighted according to how close $\mathbf{x}_i$ is to nearby observations from C_0.

An important advantage of kernel methods such as the SVM lies in their modularity: to solve a different problem, just use a different kernel function. A wide variety of kernel functions are available for solving various domain-specific problems [e.g., 1, 2, 3]. Many of these domain-specific kernels, such as the latent-semantic kernel [1] and the mismatch string kernel [2], are defined explicitly as inner products in the feature space. That is, explicit feature vectors are first defined using domain-specific knowledge, and a simple inner-product kernel, $\phi(\mathbf{u}; \mathbf{v}) = \mathbf{u}^T\mathbf{v}$, is used. Unfortunately, one cannot simply use an inner-product kernel in LAGO. With inner-product kernels, one can no longer interpret (2) as a locally adjusted kernel density

estimate of p_1. More importantly, the volume of the kernel $|\mathbf{R}_i|$, a very important ingredient of LAGO, is missing for the inner-product kernel. In other words, it is not clear how to compute r_i — equation (3) — and construct the decision function (2).

We propose a solution to this problem and make LAGO applicable to a much wider variety of practical problems. The gist of our solution is to apply the "LAGO principle" on the unit sphere, instead of in the Euclidean space. This particular solution is based upon three critical insights:

(I1) Most kernel functions used in kernel density estimation have a common structure. Suppose $\mathbf{x} \in \mathbb{R}^d$, then these kernel functions can often be written as

$$\phi(\mathbf{x}; \mathbf{x}_i, r_i\mathbf{I}) = \frac{C^d}{|r_i\mathbf{I}|}\phi_c\left(\frac{d(\mathbf{x}, \mathbf{x}_i)}{r_i}\right), \tag{4}$$

where C is the normalizing constant such that $C\int \phi_c(z)dz = 1$. There are two key ingredients, a basic (positive) kernel function $\phi_c(\cdot)$, and a distance metric $d(\cdot, \cdot)$. For example, the radial-basis kernel has this structure. Simply take $d(\mathbf{x}, \mathbf{x}_i) = \|\mathbf{x} - \mathbf{x}_i\|$ to be the Euclidean distance and $\phi_c(z) = e^{-z^2/2}$.

(I2) Using the kernel function (4), the "LAGO principle" is extremely easy to describe. First, pick a distance metric $d(\cdot, \cdot)$. Using the chosen distance metric, define r_i according to (3). Multiply each kernel by $|r_i\mathbf{I}|$. Finally, add all the pieces together according to (2). Notice that the "LAGO principle" does not depend on the distance metric $d(\cdot, \cdot)$ or the basic kernel function $\phi_c(\cdot)$.

(I3) If $\mathbf{u}, \mathbf{v}$ are unit vectors, we can decompose any inner product and write it as $\mathbf{u}^T\mathbf{v} = \cos\big(\arccos(\mathbf{u}^T\mathbf{v})\big)$. Then, we can view $\arccos(\mathbf{u}^T\mathbf{v})$ as a distance metric — it measures the *angular distance* between two points lying on the unit sphere, and $\cos(\cdot)$ as the basic kernel function $\phi_c(\cdot)$ — if we truncate the cosine function to zero beyond $\pm\pi/2$ to ensure that it is positive.

Based on (I1)-(I3), our solution is as follows: Given explicit feature vectors $\mathbf{x}, \mathbf{x}_i \in \mathbb{R}^d$, first normalize all the feature vectors to lie on the unit sphere, i.e., $\|\mathbf{x}\| = \|\mathbf{x}_i\| = 1$, and then apply the "LAGO principle" using the angular distance metric,

$$d(\mathbf{x}, \mathbf{x}_i) = \theta(\mathbf{x}, \mathbf{x}_i) = \arccos(\mathbf{x}_i^T\mathbf{x}), \tag{5}$$

and the truncated cosine kernel function,

$$\phi_c(z) = \cos(z)I\left(|z| < \frac{\pi}{2}\right). \tag{6}$$

Hence, the "LAGO principle" stays exactly the same as before. The only change lies in the type of geometry: Rather than Euclidean geometry, we now work with unit-sphere geometry instead. So we measure distances differently and use a different kernel function. Empirical experiments using the latent-semantic kernel [1] and text data from `http://www.cs.cmu.edu/~webkb/` show that our solution is successful, often performing better and taking much less time to train than the SVM.

References

[1] Cristianini, N., Shawe-Taylor, J., and Lodhi, H. (2001). Latent semantic kernels. In *Proceedings of the 18-th International Conference on Machine Learning*, pages 66–73. Morgan-Kaufmann.

[2] Leslie, C., Eskin, E., Cohen, A., Weston, J., and Noble, W. (2004). Mismatch string kernels for discriminative protein classification. *Bioinformatics*, **20**(4), 467–476.

[3] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

[4] Zhu, M., Su, W., and Chipman, H. A. (2006). LAGO: A computationally efficient approach for statistical detection. *Technometrics*, **48**, 193–205.

Topic: learning algorithms
Preference: oral