Neural Networks 21 (2008) 1220–1223

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Neural networks letter

LAGO on the unit sphere

Alexandra Laflamme-Sanders, Mu Zhu *

University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 21 March 2008
Accepted 1 August 2008

Keywords:
Geometry
Inner product
Kernel method
Rare target detection
SVM

ABSTRACT

LAGO is an efficient kernel algorithm designed specifically for the rare target detection problem. However, unlike other kernel algorithms, LAGO cannot be easily used with many domain-specific kernels. We solve this problem by first providing a unified framework for LAGO and clarifying its basic principle, and then applying that principle on the unit sphere instead of in the Euclidean space.

1. Introduction

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. Refer to Bolton and Hand (2002) for various interesting applications.

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, e.g., Cristianini and Shawe-Taylor (2000)). 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 (using quadratic programming) 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.

First developed in the statistics research community, LAGO (Zhu, Su, & Chipman, 2006) is an extremely efficient kernel method designed specifically for the rare target detection problem. It constructs a decision function much like Eq. (1) but does not use any iterative optimization procedure to do so. The main purposes of this short article are: (i) to introduce LAGO to the broader computational intelligence research community, and (ii) to resolve an existing difficulty faced by LAGO, which prevents LAGO from being used with many interesting domain-specific kernel functions.

2. LAGO

The decision function constructed by LAGO for ranking unlabelled observations can be written as (Zhu, 2008; Section 4.2)

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

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, $\alpha_i (i = 1, 2, \ldots, n)$, 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 – Eq. (3) – for every $\mathbf{x}_i \in C_1$. This is extremely efficient since the size of C_1 is typically very small for rare target problems.

* Corresponding address: University of Waterloo, Statistics and Actuarial Science, 200 University Ave. W., Waterloo, Ontario, Canada N2L 3G1. Tel.: +1 519 888 4567; fax: +1 519 746 1875.
E-mail address: m3zhu@math.uwaterloo.ca (M. Zhu).

0893-6080/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2008.08.002

Zhu et al. (2006) 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 or, equivalently, according to the local density of C_0 nearby.

3. Difficulty

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, and the underlying learning algorithm can stay the same (Shawe-Taylor & Cristianini, 2004). A wide variety of kernel functions are available for solving various domain-specific problems (e.g., Cristianini, Shawe-Taylor, and Lodhi (2001), Leslie, Eskin, Cohen, Weston, and Noble (2004) and Shawe-Taylor and Cristianini (2004)). Many of these domain-specific kernels, such as the latent-semantic kernel (Cristianini et al., 2001) and the mismatch string kernel (Leslie et al., 2004), 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, $\phi(\mathbf{u}; \mathbf{v}) = \mathbf{u}^{\mathrm{T}}\mathbf{v}$, is used.

Strange as it may sound, 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 – Eq. (3) – and construct the decision function (2).

4. Solution

In this section, we propose a solution to the aforementioned difficulty 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 (Silverman, 1986) have a common structure. Suppose $\mathbf{x} \in \mathbb{R}^q$, then these kernel functions can often be written as

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

where C is the normalizing constant such that $C\int \phi_c(z)dz = 1$. There are two key ingredients, a (positive) basic kernel function $\phi_c(\cdot)$, and a distance metric $d(\cdot, \cdot)$. For example, the radial-basis kernel has this structure. Simply take

$$d(\mathbf{x}, \mathbf{x}_i) = \|\mathbf{x} - \mathbf{x}_i\| \tag{5}$$

to be the Euclidean distance and

$$\phi_c(z) = \mathrm{e}^{-z^{2/2}}. \tag{6}$$

The normalizing constant is $C = 1/\sqrt{2\pi}$.

(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 $\phi_c(\cdot)$.

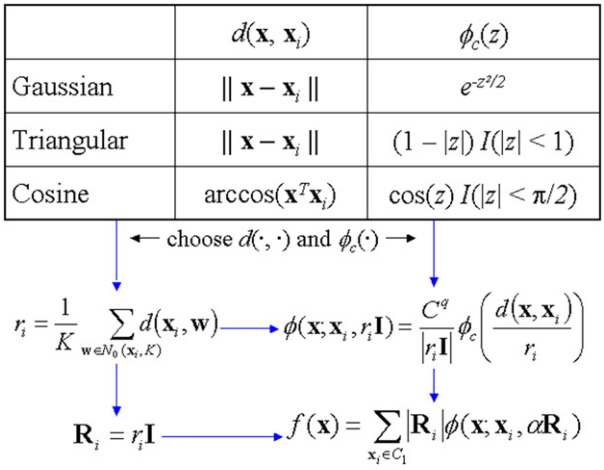

Fig. 1. A unified framework for LAGO. One is free to choose the distance metric $d(\mathbf{x}, \mathbf{x}_i)$ and the basic kernel function $\phi_c(\cdot)$; the fundamental "LAGO principle" is independent of these choices. The three possible choices given explicitly in the figure, "Gaussian", "Triangular" and "Cosine", are not exhaustive; many other choices are possible.

(I3) If $\mathbf{u}$, $\mathbf{v}$ are unit vectors, we can decompose any inner product and write it as

$$\mathbf{u}^{\mathrm{T}}\mathbf{v} = \cos\left(\arccos(\mathbf{u}^{\mathrm{T}}\mathbf{v})\right). \tag{7}$$

Then, we can view $\arccos(\mathbf{u}^{\mathrm{T}}\mathbf{v})$ 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 $\mathbf{x}, \mathbf{x}_i \in \mathbb{R}^q$, first remove the overall mean and 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,

$$d(\mathbf{x}, \mathbf{x}_i) = \theta(\mathbf{x}, \mathbf{x}_i) = \arccos(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}), \tag{8}$$

and the truncated cosine kernel function,

$$\phi_c(z) = \cos(z)I\left(|z| < \frac{\pi}{2}\right). \tag{9}$$

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 (using angular distances rather than Euclidean distances), and use a different kernel function (the truncated cosine kernel rather than other kernels). Fig. 1 presents a unified framework for LAGO, and shows how various kernels fit into the general form of (4).

Putting (2)–(4) and (8)–(9) together, we obtain the decision function:

$$f(\mathbf{x}) = \sum_{\mathbf{x}_i \in C_1} \cos\left(\frac{\arccos(\mathbf{x}_i^{\mathrm{T}}\mathbf{x})}{\alpha r_i}\right) I\left(\left|\frac{\arccos(\mathbf{x}_i^{\mathrm{T}}\mathbf{x})}{\alpha r_i}\right| < \frac{\pi}{2}\right), \tag{10}$$

where

$$r_i = \frac{1}{K}\sum_{\mathbf{w}\in N_0(\mathbf{x}_i, K)} \arccos(\mathbf{x}_i^{\mathrm{T}}\mathbf{w}). \tag{11}$$

Again, α and K are global tuning parameters. Our empirical experiences suggest that LAGO is not very sensitive to K and much more sensitive to α. In practice, it often suffices to fix $K = 5$, leaving us with just one tuning parameter, α.