University of Waterloo

Mu Zhu

LAGO: A Computationally Efficient Method for Statistical Detection

Copyright © 2003–2005 by Mu Zhu
Acknowledgment

Co-authors: Wanhua Su; Hugh A. Chipman.

Research support: NSERC; MITACS; CFI; Acadia Centre for Mathematical Modelling and Computation.

Others: William J. Welch, R. Wayne Oldford, Jerry F. Lawless, Mary Thompson, S. Young.
1. The statistical detection problem.
2. Average precision.
3. Drug discovery and high throughput screening.
4. LAGO.
5. Radial basis function (RBF) networks.
6. Support vector machines (SVMs).
7. Results.

Agenda
The Detection Problem

A fraction $t$ from $C$, out of which $h(t)$ is relevant.

Figure 1: Illustration of a typical detection operation. A small fraction of the entire collection $C$ is of interest (relevant). An algorithm detects $\pi$ relevant instances out of which $h(t)$ is detected.
Figure 2: Illustration of the typical modelling and prediction process.
Figure 3: Illustration of some hit curves. Note that $h_A(t)$ and $h_B(t)$ cross each other; $h_P(t)$ is an ideal curve produced by a perfect algorithm; $h_R(t)$ corresponds to the case of random detection.
The Average Precision

Let \( h(t) \) be the hit curve; let \( r(t) = h(t) \pi \) and \( p(t) = h(t)/t \).

Then,

\[
\text{Average Precision} = \int p(t) \, dr(t).
\]

In practice, \( h(t) \) takes values only at a finite number of points \( t_i = i/n \), \( i = 1, 2, \ldots, n \). Hence, the integral (1) is replaced with a finite sum

\[
\int p(t) \, dr(t) = \sum_{i=1}^{n} p(t_i) \Delta r(t_i).
\]

where \( \Delta r(t_i) = r(t_i) - r(t_{i-1}) \).
\[ A \text{Simple Example}\]
Figure 4: Illustration of the high throughput screening process.

High Throughput Screening (HTS)
Drug Discovery Data

Original data from National Cancer Institute (NCI) with predictors calculated by GlaxoSmithKline, Inc.

1. \( n = 29,812 \) chemical compounds, of which only 608 are active against the HIV virus.

2. \( p = 6 \) chemometric descriptors of the molecular structure, known as BCUT numbers.

3. Using stratified sampling, randomly split the data to produce a training set and a test set (each with \( n = 14,906 \) and 304 active compounds).

4. Tuning parameters selected using 5-fold cross-validation on the training set, and compare performance on the test set.

LAGO

Copyright © 2003-2005 by Min Zhu
1. Given a vector of predictors $x$, the posterior probability $g(x) = P(y=1|x) = \pi_1 p_1(x) + \pi_0 p_0(x)$ is arguably a good ranking function, i.e., items with a high probability of being relevant should be ranked first.

2. As far as ranking is concerned, all monotonic transformations of $g$ are clearly equivalent, so it suffices to focus on the ratio function $f(x) = \frac{p_1(x)}{p_0(x)}$ since the function $g$ is of the form $g(x) = af(x) + 1$ for some constant $a$ not depending on $x$, which is a monotonic transformation of $f$.

\[ f(x) = \frac{p_1(x)}{p_0(x)} = (x) f \]

\[ \frac{(x)^0 d}{(x)^1 d} = (x) f \]

I. Given a vector of predictors $x$, the posterior probability

\[ \text{Ranking Functions} \]
Two Assumptions

A1. For all practical purposes the density function $p_1(x)$ can be assumed to be relatively flat in comparison with $p_0(x)$. In that neighborhood can be moreover, the density function $p_0(x)$ in its immediate local neighborhood; observations, say $m$, from $C_0$ in its immediate local neighborhood; there are at least a certain number of observations $x_i \in C_1$. For every observation $x_i \in C_1$, the support of $p_1$ can be written as

$$\bigcap_{i=1}^{m} S_{p_1}^c \subset \mathbb{R}^d \cap \bigcap_{i=1}^{m} S_{p_0} = S$$

Two Assumptions
In order to build a predictive model for statistical detection problems, it suffices to estimate $p_1(x)$ alone and adjust $p_0(x)$ locally depending on nearby $p_1(x)$. 
Assume $x \in \mathbb{R}$ is a scalar.

Generalize to $x \in \mathbb{R}^p$ for $p < 1$. 

$\uparrow$

Assume $x \in \mathbb{R}$ is a scalar.
Step 1: Estimating $p$

1. Use an adaptive bandwidth kernel estimator:

$$
\hat{p}(x) = \frac{1}{n} \sum_{y_i \in C_0} K(x; x_i, r_i)
$$

(5)

2. For each $x_i \in C_1$, choose $r_i$ adaptively to be the average distance between $x_i$ and its $K$-nearest neighbors from $C_0$, i.e.,

$$
\forall m \in \mathbb{Z}, \quad \sum_{y \in N(x_i, K)} \frac{|y - x_i|}{I} = r_i
$$

(6)

3. The notation $N(x_i, K)$ is used to refer to the set that contains the $K$-nearest class-0 neighbors of $x_i$.

Then the parameter to be selected empirically, e.g., with cross-validation.

The number $K$ is a tuning parameter to be selected empirically, e.g., with cross-validation.
Figure 5: The ancient Chinese game of Go is a game in which each player tries to claim as many territories as possible on the board. Image taken from http://go.arad.ro/Introducere.html.
Step 2: Local Adjustment of $p_1$.
An Idealized Situation

Instead of saying $p_0(x_i; x) \approx c_i$, we shall explicitly assume that, for every $i \in \mathbb{C}$, there exist i.i.d. observations $w_1, w_2, \ldots, w_m$ from $C_0$ that can be uniformly distributed on the interval $[x_i - 1/2c_i, x_i + 1/2c_i]$.

If $r_0$ is the average distance between $x_0$ and its $K$ nearest neighbors from class $C_0$ ($K < m$), then we have

$$\frac{c_0(1 + m)}{1 + K} = (0, \ell) F$$

Theorem 1

Let $x_0$ be a fixed observation from class $C_1$. Suppose

$\text{w}1, \text{w}2, \ldots, \text{w}m$ are i.i.d. observations from class 0 that are uniformly distributed around $x_0$, say on the interval $[x_0 - 1/2c_0, x_0 + 1/2c_0]$. If $m > K$, then we have
Implications of the Theorem

\[ \hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{r_i} \]

Since \( r_i \) is already computed, there is no need to estimate \( c_i \). We simply use (8) for \( K > m \), can conclude \( r_i \) is approximately proportional to \( 1/c_i \).

For \( K \leq m \), can conclude \( r_i \) is approximately uniformly around every \( x \in C_1 \). Can assume there are at least \( m \) observations from \( C_0 \) distributed.
1. Estimation of $p_1$:

\[
\hat{p}_1(x) = \frac{1}{n} \sum_{i=1}^{n} K(x; x_i, r_i)
\]

2. Adjustment of $p_1$ according to $p_0$ nearby:

\[
\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K(x; x_i, r_i) c_i \Rightarrow \hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} r_i K(x; x_i, r_i)
\]

A Short Summary
1. For every training observation in class 1, $x^i \in C_1$, compute a specific bandwidth vector $r^i = (r^i_1, r^i_2, ..., r^i_d)^T$, where $r_{ij}$ is the average distance between $x^i$ and its $K$-nearest class-0 neighbors in the $j$th dimension.

2. For every new observation $x$ where a prediction is required, score and rank $x$ according to:

$$f(x) = \prod_{j=1}^{d} \prod_{p=1}^{K} \frac{1}{1 + r_{ij}} = (x)^f$$

which uses the Naive Bayes principle (Hastie et al. 2001, Section 6.6.3).

Extension to $\mathbb{R}^p$
Some Kernel Functions

\[ I \geq |n| \]
\[ I = (n)f \]
\[ |n| - I = (n)f \]
\[ \left( \frac{z}{\varepsilon^2 n} \right) \exp \propto (n)f \]
Radial Basis Function Networks

A radial basis function (RBF) network has the form:

\[
 f(x) = \sum_{i=1}^{n} \beta_i K(x; \mu_i, r_i),
\]

where \( K(x; \mu, r) \) is a kernel function centered at location \( \mu \) with radius \( r \). Clearly, in order to construct an RBF network we must specify the centers and the radii of the RBFs. For example, \( \mu \) is a kernel function centered at location \( \mu \) with radius \( r \):

\[
(p_1, \ldots, p_d, \cdot, \cdot, \cdot) = \mu
\]

A radial basis function (RBF) network has the form:

Radial Basis Function Networks
General Parameterization

Figure 6: Illustration. Left: Density functions $p_0$ and $p_1$. Right: The ratio function $f(x)$. α-effect: radius  β-effect: height
Parameterizing the $\alpha$- and $\beta$-Effects

- Take a kernel function belonging to a location-scale family:
  \[ \mathcal{K}_{\epsilon^2}(x-x_i) \]

In constructing the LAGO model, we have in effect argued that $\beta = 0$ (or $\beta' = 1$).

\[ \left( \frac{\epsilon_x}{x-x_i} \right) \mathcal{K}_{\epsilon^2} \equiv \left( \frac{\epsilon_x}{x-x_i} \right) \mathcal{K}_{1-\epsilon^2} \propto \left( \frac{\epsilon_x}{x-x_i} \right) \mathcal{K}_{1} \left( \frac{\epsilon_x}{x-x_i} \right) \]

Can explicitly parameterize the $\alpha$- and $\beta$-effects as follows:

\[ \left( \frac{x}{x-x_i} \right) \mathcal{K}_{\epsilon^2} \]

Take a kernel function belonging to a location-scale family:

Parameterizing the $\alpha$- and $\beta$-Effects
The LAGO Model

With two tuning parameters, $K$ and $\alpha$, the final form of the LAGO model is:

$$f(x) = \sum_{\ell=1}^{K} \prod_{j=1}^{r_{ij}} \prod_{p} \chi_{\ell}^{\ell} \prod_{u=1}^{r_{ij}} \ell_{u}^{u} = (x)f$$

The final form of the LAGO model is:

- For relatively large $K$, can usually obtain a model with very similar performance by setting $\alpha > 1$ and using a much smaller $K$.
- Hence by keeping $\alpha$, can restrict ourselves to a much narrower range.
- When selecting $K$ by cross-validation, can usually obtain a model with very similar performance by setting $\alpha < 1$.
- Hence by keeping $\alpha$, can restrict ourselves to a much narrower range.

The LAGO Model
Separating Hyperplanes

A separating hyperplane is characterized by

\[ f(x) = \beta^T x + \beta_0 = 0 \]

where \( x_i \in \mathbb{R}^d \).

Given \( y_i \in \{-1, +1\} \) (two classes), a hyperplane is a separating hyperplane if there exists \( c > 0 \) such that

\[ y_i (\beta^T x_i + \beta_0) \geq c \quad \forall i. \]

A hyperplane can be reparameterized by scaling, e.g.,

\[ 0 = (0\beta' + \mathbf{x}_L_\beta') s \quad \text{is the same as} \quad 0 = 0\beta' + \mathbf{x}_L_\beta' \]

A separating hyperplane satisfies

\[ a \in \mathbb{R}^d \quad \text{such that} \quad c_0 \in \mathbb{R}^d \]

\[ 0 = 0\beta' + \mathbf{x}_L_\beta' = (x)f \]

Given \( \mathbf{x} \in \mathbb{R}^d \), a hyperplane in \( \mathbb{R}^d \) is characterized by

Separating Hyperplanes
Figure 7: Two separating hyperplanes, one with a larger margin than the other.
It can be calculated that a canonical separating hyperplane has maximal margin equal to 

$$\frac{\|\mathring{\mathcal{E}}\|}{1}$$

It is found a „best“ (maximal) margin (canonically) canonical separating hyperplane to separate the two classes labeled +1 and -1 by solving

$$\begin{align*}
\text{min}_{\omega} & \sum_{i=1}^{n} \xi_i + \frac{1}{2} \|\mathring{\mathcal{E}}\|_2^2 \\
\text{s.t.} \quad & 0 \leq \xi_i \\
\text{and} \quad & y_i (\omega^T \mathbf{x}_i + \omega_0) \geq 1 - \xi_i \quad \forall i.
\end{align*}$$

The Support Vector Machine (SVM) finds a „best“ (maximal) margin (canonically) canonical separating hyperplane has
Let $w_0$ and $w_1$ be class weights; extend the optimization problem to be:

$$\min \quad \frac{1}{2} \|\beta\|^2 + \gamma_1 \sum_{y_i=1} \xi_i + \gamma_0 \sum_{y_i=0} \xi_i$$

s.t. $\xi_i \geq 0$ and $y_i(\beta^T x_i + \beta_0) \geq 1 - \xi_i$ for all $i$.

where $\gamma_0 = \gamma w_0$ and $\gamma_1 = \gamma w_1$. 

**ASVM for Unbalanced Classes**
SVM: Characterizing the Solution

The solution for \( \beta \) is characterized by

\[
\hat{\beta} = \sum_{i \in SV} \hat{\alpha}_i y_i x_i,
\]

where \( \hat{\alpha}_i \geq 0 \) (for \( i = 1, 2, \ldots, n \)) are solutions to the dual optimization problem and \( SV \) is the set of “support vectors” with \( \hat{\alpha}_i \) strictly positive.

This means the resulting hyperplane can be written as

\[
0 = \vec{c}^T + \sum_{i \in SV} \hat{\alpha}_i y_i x_i^T \vec{c} = \vec{c}^T + \vec{x}^T \vec{c} = (\vec{x})_f
\]

The solution for \( \vec{c} \) is characterized by

\[
\sum_{i \in SV} \vec{x}^T y_i \vec{c} = \vec{c}
\]
RBF network (Schölkopf et al. 1997).

Hence SVM can be viewed as an automatic way of constructing an RBF network (Schölkopf et al. 1997).

The boundary is linear in the space of 
\( h(x) \) where 
\( h(\cdot) \) is such that 
\[ K(u; v) = \langle h(u), h(v) \rangle \]

The boundary is linear in the space of \( h(\cdot) \) where \( h(\cdot) \) is such that 
\[ \langle (\lambda) h_1, (\lambda) h_2 \rangle = (\lambda \cdot \lambda) \]

Can replace the inner product \( \langle x; x \rangle \) with a kernel function \( K(x; \cdot) \).
Figure 8: The average precision of all algorithms evaluated on the test data.

**Performance Results: Drug Discovery Data**
Performance Results: ANOVA Set-up

Let $\mu_K$, $\mu_S$, $\mu_A$, $\mu_U$, and $\mu_T$ be the average result of K-NN, SVM, ASVM, and LAGO, respectively.

<table>
<thead>
<tr>
<th>Contrast</th>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cntr1</td>
<td>$\mu_T - \mu_G$</td>
<td>0.0027</td>
</tr>
<tr>
<td>Cntr2</td>
<td>$\mu_G - \mu_A$</td>
<td>0.0339</td>
</tr>
<tr>
<td>Cntr3</td>
<td>$\mu_A - \mu_S$</td>
<td>0.0230</td>
</tr>
<tr>
<td>Cntr4</td>
<td>$\frac{\mu_S - \mu_K}{2}$</td>
<td>0.0157</td>
</tr>
<tr>
<td>Cntr5</td>
<td>$\mu_U - \mu_K$</td>
<td>0.0014</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contrast</th>
<th>Estimantae</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cntr1</td>
<td>$\mu_T - \mu_G$</td>
<td>0.0027</td>
</tr>
<tr>
<td>Cntr2</td>
<td>$\mu_G - \mu_A$</td>
<td>0.0339</td>
</tr>
<tr>
<td>Cntr3</td>
<td>$\mu_A - \mu_S$</td>
<td>0.0230</td>
</tr>
<tr>
<td>Cntr4</td>
<td>$\frac{\mu_S - \mu_K}{2}$</td>
<td>0.0157</td>
</tr>
<tr>
<td>Cntr5</td>
<td>$\mu_U - \mu_K$</td>
<td>0.0014</td>
</tr>
</tbody>
</table>

the Gaussian kernel, respectively.

Let $\mu_K$, $\mu_S$, $\mu_A$, $\mu_U$, and $\mu_T$ be the average result of K-NN, SVM, ASVM, and LAGO, respectively.
Performance Results: ANOVA Summary

<table>
<thead>
<tr>
<th>Source</th>
<th>SS (×10⁻⁴)</th>
<th>df</th>
<th>MS (×10⁻⁴)</th>
<th>P₀</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
<td>233.504</td>
<td>15</td>
<td>18.893</td>
<td>0.0014</td>
<td></td>
</tr>
<tr>
<td>Cntr1</td>
<td>0.140</td>
<td>3</td>
<td>0.046</td>
<td>0.8258</td>
<td></td>
</tr>
<tr>
<td>Cntr2</td>
<td>22.916</td>
<td>1</td>
<td>22.916</td>
<td>0.0017</td>
<td></td>
</tr>
<tr>
<td>Cntr3</td>
<td>10.534</td>
<td>1</td>
<td>10.534</td>
<td>0.0090</td>
<td></td>
</tr>
<tr>
<td>Cntr4</td>
<td>6.531</td>
<td>1</td>
<td>6.531</td>
<td>0.0014</td>
<td></td>
</tr>
<tr>
<td>Cntr5</td>
<td>0.036</td>
<td>1</td>
<td>0.036</td>
<td>0.8258</td>
<td></td>
</tr>
<tr>
<td>Splits</td>
<td>18.827</td>
<td>1</td>
<td>18.827</td>
<td>0.0001</td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td>10.893</td>
<td>23</td>
<td>0.469</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>263.274</td>
<td>23</td>
<td>10.893</td>
<td>0.0001</td>
<td></td>
</tr>
</tbody>
</table>
Figure 9: Only the initial part of the curves (up to \( n = 500 \)) are shown.
Main Conclusions

(Triangle LAGO $\sim$ Gaussian LAGO) $\succ$ (Uniform LAGO $\sim$ KNN).

Computationally, ASVM is extremely expensive.
<table>
<thead>
<tr>
<th>Split</th>
<th>SVM</th>
<th>ASVM</th>
<th>Total Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14602</td>
<td>304</td>
<td>14602</td>
</tr>
<tr>
<td>2</td>
<td>6755</td>
<td>293</td>
<td>1863</td>
</tr>
<tr>
<td>3</td>
<td>11706</td>
<td>293</td>
<td>3156</td>
</tr>
<tr>
<td>4</td>
<td>3472</td>
<td>303</td>
<td>11419</td>
</tr>
<tr>
<td>5</td>
<td>5927</td>
<td>294</td>
<td>11531</td>
</tr>
<tr>
<td></td>
<td>291</td>
<td></td>
<td>11706</td>
</tr>
</tbody>
</table>

The Number of SYS
Figure 10: Choosing $\alpha$ and $\beta$ (while fixing $K = 5$) using 5-fold CV.

Empirical Evidence: $\beta = 0$
References


