Sparse Bayesian Learning Based on an Efficient Subset Selection

Liefeng Bo, Ling Wang, and Licheng Jiao

Institute of Intelligent Information Processing and National Key Laboratory
for Radar Signal Processing, Xidian University, Xi'an 710071, China
blf0218@163.com

Abstract. Based on rank-1 update, Sparse Bayesian Learning Algorithm (SBLA) is proposed. SBLA has the advantages of low complexity and high sparseness, being very suitable for large scale problems. Experiments on synthetic and benchmark data sets confirm the feasibility and validity of the proposed algorithm.

1 Introduction

Regression problem is one of the fundamental problems in the field of supervised learning. It can be thought of as estimating the real valued function from a samples set of noise observation. A very successful approach for regression is Support Vector Machines (SVMs) [1-2]. However, they also have some disadvantages [3]:

- To derive analytically error bars for SVMs is very difficult.
- The solution is usually not very sparse.
- Kernel function must satisfy Mercer's condition.

In order to overcome the above problems, Relevance Vector Machine (RVM) [3] is proposed, which is very elegant and obtains a high sparse solution. However, RVM needs to solve linear equations, whose cost is very expensive, and therefore not feasible for large scale problems.

Based on rank-1 update, we propose Sparse Bayesian Learning Algorithm (SBLA), which has low complexity and high sparseness, thus being very suitable for large scale problems. Experiments on synthetic and benchmark data sets confirm the feasibility and validity of SBLA.

2 Model Specification

Let $z = \{(\mathbf{x}_1, y_1), \cdots (\mathbf{x}_l, y_l)\}$ be empirical samples set drawn from

$$y_i = f(\mathbf{x}_i, \mathbf{w}) + \varepsilon_i, \quad i = 1, 2, \cdots l, \tag{1}$$

where ε_i is independent samples from some noise process which is further assumed to be mean-zero Gaussian with variance σ^2. We further assume

$$f(\mathbf{x},\mathbf{w})=\sum_{i=1}^{l} w_i k(\mathbf{x},\mathbf{x}_i)\,. \tag{2}$$

According to Bayesian inference, the posterior probability of $\mathbf{w}$ can be expressed as

$$P(\mathbf{w}\,|\,z)=\frac{P(z\,|\,\mathbf{w})P(\mathbf{w})}{P(z)}\,. \tag{3}$$

Due to the assumption of independence of z, Likelihood $P(z\,|\,\mathbf{w})$ can be written as

$$P(z\,|\,\mathbf{w})=(2\pi\sigma^2)^{l/2}\exp\left(\frac{1}{2\sigma^2}(\mathbf{Kw}-\mathbf{y})^T(\mathbf{Kw}-\mathbf{y})\right). \tag{4}$$

If Gaussian prior $P(\mathbf{w})=(2\pi\gamma^2)^{l/2}\exp\left(\frac{\mathbf{w}^T\mathbf{w}}{2\gamma^2}\right)$ is chosen, maximizing the log-posterior is equivalent to minimizing the following likelihood function

$$\hat{\mathbf{w}}=\arg\min\left(L(\mathbf{w},\lambda)=\left(\mathbf{w}^T\left(\mathbf{K}^T\mathbf{K}+\lambda\mathbf{I}\right)\mathbf{w}-2\mathbf{w}\mathbf{K}^T\mathbf{y}\right)\right). \tag{5}$$

where $\lambda=\sigma^2/\gamma^2$, $\mathbf{I}$ is unit matrix.

3 Sparse Bayesian Learning Algorithm

For large datasets, the classic methods for quadratic programming such as conjugate gradient methods [4] are not feasible, due to the requisite time and memory costs. The following greedy approximation scheme can be used. Starting with an empty set $P=\varnothing$ and set $Q=\{1,2,\cdots l\}$, we select at each iteration a new basis function s from Q, and resolve the problem (**5**) containing the new basis function and all previously picked basis functions. The basis function is deleted when the removing criterion is satisfied and the algorithm is terminated when certain criterion is satisfied.

3.1 Adding One Observation

Let $\mathbf{H}=(\mathbf{K}^T\mathbf{K}+\lambda\mathbf{I})$ and $\mathbf{b}=\mathbf{K}^T\mathbf{y}$, then (**5**) can be rewritten as

$$\hat{\mathbf{w}}=\arg\min\left(L(\mathbf{w},\lambda)=\left(\mathbf{w}^T\mathbf{H}\mathbf{w}-2\mathbf{w}\mathbf{b}\right)\right). \tag{6}$$

Assume that $P=\{p_1,\cdots,p_n\}$, $\mathbf{R}^{t-1}=(\mathbf{H}_{PP})^{-1}$, and $\mathbf{h}_s=[H_{p_1 s},\cdots H_{p_n s}]^T$ at the $(t-1)^{th}$ iteration. If the s^{th} basis function is added in t^{th} iteration, in terms of a rank-1 update [5], we have

$$\mathbf{R}^t=\begin{bmatrix}\mathbf{R}^{t-1} & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix}+\alpha\begin{bmatrix}\boldsymbol{\beta}\\ -1\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}^T & -1\end{bmatrix}. \tag{7}$$

where $\boldsymbol{\beta} = \mathbf{R}^{t-1}\mathbf{h}_s$, $\alpha = (H_{ss} - \mathbf{h}_s^T\boldsymbol{\beta})^{-1}$. Thus the weights can be updated by the following equations

$$\begin{bmatrix} \mathbf{w}_P^t \\ w_s^t \end{bmatrix} = \mathbf{R}^t \begin{bmatrix} \mathbf{b}_P \\ b_s \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{t-1}\mathbf{b}_P \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} \boldsymbol{\beta} \\ -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}^T\mathbf{b}_P & -b_s \end{bmatrix}. \tag{8}$$

Together with $\mathbf{w}_P^{t-1} = \mathbf{R}^{t-1}\mathbf{b}_P$, we have

$$\begin{bmatrix} \mathbf{w}_P^t \\ w_s^t \end{bmatrix} = \begin{bmatrix} \mathbf{w}_P^{t-1} \\ 0 \end{bmatrix} + \alpha(\boldsymbol{\beta}^T\mathbf{b}_P - b_s)\begin{bmatrix} \boldsymbol{\beta} \\ -1 \end{bmatrix}. \tag{9}$$

The rest major problem is how to pick the appropriate basis function at each iteration. A natural idea is to choose s^{th} basis function for which we have the biggest decrease in the objective function, which is called pre-fitting [6]. However its cost is too expensive. Here we adopt a cheap selection criterion that is also used in numerical algebra [7]

$$s = \max_{k \in Q}\left(abs\left(g_k^t\right)\right). \tag{10}$$

where $g_k^t = \sum_{j=1}^{n} \mathbf{K}_{kj} w_j^t - y_k$.

3.2 Removing One Observation

Let $\mathbf{R}^{(t,s)}$ represent the matrix with the $(P_s)^{th}$ observation deleted at the t^{th} iteration. In terms of a rank-1 update, we have

$$\left(\mathbf{R}^{(t,s)}\right)_{ij} = \left(\mathbf{R}^t\right)_{ij} - \frac{(\mathbf{R}^t)_{is}(\mathbf{R}^t)_{sj}}{(\mathbf{R}^t)_{ss}} \qquad i,j \le n; i,j \ne s. \tag{11}$$

$$\left(\mathbf{w}^{(t,s)}\right)_i = \sum_{j=1, j\ne k}^{n} \left(\left(\mathbf{R}^t\right)_{ij} - \frac{(\mathbf{R}^t)_{is}(\mathbf{R}^t)_{sj}}{(\mathbf{R}^t)_{ss}}\right) b_{p_j} \qquad j \le n; j \ne s. \tag{12}$$

Together with $\mathbf{w}_P^t = \mathbf{R}^t\mathbf{b}_P$, (**12**) is simplified as

$$\left(\mathbf{w}^{(t,s)}\right)_i = \mathbf{w}_i^t - \mathbf{w}_s^t \frac{(\mathbf{R}^t)_{is}}{(\mathbf{R}^t)_{ss}} \qquad i \le n; i \ne s. \tag{13}$$

Then the problem is how to pick an appropriate basis function at each iteration. Let $f^{(t,k)}$ be the objective value with the $(P_k)^{th}$ observation deleted, we have

$$f^{(t,k)} = -\sum_{i,j=1, i,j\ne k}^{n} b_{p_i}\left(\mathbf{R}^{(t,k)}\right)_{ij} b_{p_j}. \tag{14}$$

Substituting (**11**) into (**14**), we obtain

$$f^{(t,k)} = -\sum_{i,j=1}^{n} b_{p_i}\left(\mathbf{R}^t\right)_{ij} b_{p_j} + \sum_{i,j=1}^{n} b_{p_i}\frac{(\mathbf{R}^t)_{ik}(\mathbf{R}^t)_{kj}}{(\mathbf{R}^t)_{kk}} b_{p_j} = f^t + \frac{\left(\sum_{j=1}^{n}(\mathbf{R}^t)_{kj} b_{p_j}\right)^2}{(\mathbf{R}^t)_{kk}} \tag{15}$$

By the virtue of $\sum_{i=1}^{n} b_{p_i} (\mathbf{R}^t)_{ik} = w_k^t$, (**15**) is translated into

$$\Delta f^{(t,k)} = f^{(t,k)} - f^t = \frac{\left(w_k^t\right)^2}{(\mathbf{R}^t)_{kk}} . \tag{16}$$

Thus we can obtain s by

$$s = \arg\min_{k \in P} \left(\Delta f^{(t,k)} \right) . \tag{17}$$

If $\Delta f^{(t,s)}$ is smaller than some threshold ε, the corresponding basis function will be removed. The algorithm is terminated if the number of removed observations is larger than some threshold M.

Accordingly, Sparse Bayesian Learning Algorithm can be described as the following

```
Sparse Bayesian Learning Algorithm
1.  Let $\mathbf{w}^0 = \mathbf{0}^T$, $\mathbf{g}^0 = -\mathbf{y}$, $Q = \{1,2,3,\cdots,l\}$, $P = \{\}$, $t = 1$;
2.  $s = \arg\max_{k \in Q} \left( abs(\mathbf{g}_Q^t) \right)$;
3.  Add the $s^{th}$ observation and update $\mathbf{R}^t$ and $\mathbf{w}^t$ according to (7) and (9);
4.  $Q = Q - \{s\}$, $P = P + \{s\}$;
5.  $s = \arg\min_{k \in P} \left( \Delta f^{(t,k)} \right)$;
6.  If $\Delta f^{(t,s)} \le \varepsilon$, remove the observation $s$ and update $\mathbf{R}^t$ and $\mathbf{w}^t$ according to (11)
        and (13), $P = P - \{s\}$;
7.  If the stop criterion is satisfied, stop.
8.  $\mathbf{g}_Q^t = \mathbf{K}_{QP} \mathbf{w}_P^t$, $t = t + 1$, goto 2;
```

Fig. 1. Sparse Bayesian Learning Algorithm.

3.3 Parameters Selection

Let $L_{\min}$ be the minimal value of likelihood function, we have

$$\left(L_{\min} + \mathbf{y}^T \mathbf{y} \right) / l \approx 0 . \tag{18}$$

Then, the average contribution of one observation for $L_{\min}$ is about $\mathbf{y}^T \mathbf{y} / l$. Let $\varepsilon = \mathbf{y}^T \mathbf{y} / l / T$. If the decrease of the cost function is smaller than $\mathbf{y}^T \mathbf{y} / l / T$ after the s^{th} observation be deleted, we will remove it. If the number of removed observations is larger than M, the algorithm is terminated. The common choice of T and M is 10^4 and $l/10$. λ can be fixed to 10^{-3}.

3.4 Complexity of Algorithms

Calculating $\mathbf{h}_s$ and updating $\mathbf{R}$ and $\mathbf{g}_Q$ are operations of cost $O(nl)$, $O(n^2)$ and $O(n(l-n))$, respectively. Therefore, the single step computational complexity of SBLA is only $O(nl)$ and successive $(n+M)$ iterations incur a computational cost of $O(n^2 l + nMl)$. Note that SBLA has low complexity since $n, M \ll l$. Besides that, the memory footprint of the algorithms is also only $O(nl)$.

4 Simulation

In order to evaluate the performance of the proposed algorithm, we performed three experiments on synthetic and benchmark data sets. Data sets for regression come from STATLOG COLLECTION. For the sake of comparison, different algorithms used the same input sequence. The elements of Gram matrix $\mathbf{K}$ were constructed using the Gaussian kernel function of the form $k(\mathbf{x},\mathbf{y}) = \exp(\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma^2})$. In all experiments, λ, T and M was fixed to 10^{-3}, 10^4 and $l/10$, respectively. Kernel width σ was chosen by 10-fold cross validation procedure.

The Sinc function $y = \sin(x)/x + \sigma N(0,1)$ is a popular choice of illustrating support vector machines regression. Training samples were generated from Sinc function at 100 equally-spaced x-value in [-10,10] with added Gaussian noise of standard deviation 0.1. Results were averaged over 100 random instantiations of the noise, with the error being measured over 1000 noise-free test samples in [-10,10]. The decision functions and support vectors obtained by our algorithm and SVMs are shown in Fig. 4.1.

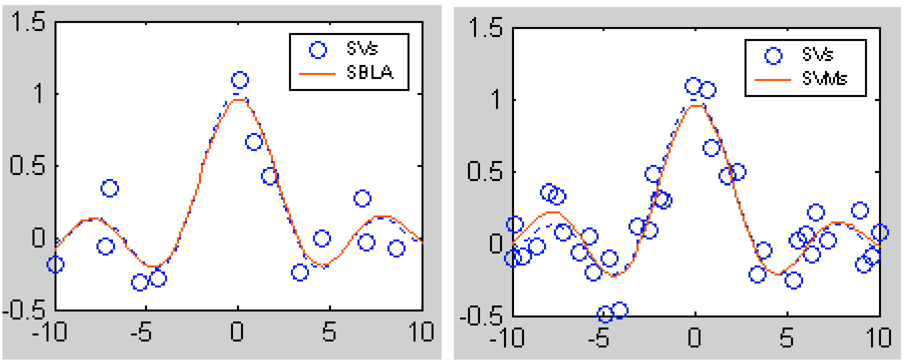

Fig. 2. Decision functions and support vectors by SBL and SVMs. Real lines denote the decision functions and circles denote support vectors.

For the Boston Housing data set, we averaged our results over 100 random split of the full dataset into 481 training samples and 25 testing samples. For the Abalone data set, we averaged our results over 10 random splits of the mother dataset into 3000 training samples and 1177 testing samples. Before experiments, we scaled all the training data in [-1,1] and then adjusted test data using the same linear transformation. The results are summarized in Table 1.

Table 1. Results obtained by SBLA and SVMs. NSV denotes the number of support vector. MSE denotes the mean squared error. For Sinc problem, unit of error is 10^{-3}.

	SBLA			SVMs		
Problem	(λ,σ)	NSV	MSE	(C,σ)	NSV	MSE
Sinc	$(10^{-3},2^{1.5})$	13.99 $\pm$ 1.93	0.89 $\pm$ 0.41	$(2^{1},2^{1.5})$	35.03 $\pm$ 4.87	1.24 $\pm$ 0.53
Housing	$(10^{-3},2^{0.5})$	62.35 $\pm$ 4.80	10.58 $\pm$ 6.91	$(2^{3},2^{0.5})$	162.40 $\pm$ 3.34	10.54 $\pm$ 8.17
Abalone	$(10^{-3},2^{-0.5})$	30.9 $\pm$ 4.12	4.48 $\pm$ 0.25	$(2^{1},2^{-0.5})$	1188 $\pm$ 47.12	4.48 $\pm$ 0.25

SBLA and SVMs obtained similar generalization ability; however, the solution of SBLA is much sparser than that of SVMs.

5 Conclusion

SBLA offers a simple and computationally efficient scheme for supervised learning. Its application is by no mean limited to regression problem. Work on classification problem is in progress.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under grant 60372050 and 60133010 and National "863" Project grant 2002AA135080.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
2. Vapnik, V.: Statistical Learning Theory. Wiley-Interscience Publication, New York (1998)
3. Tipping, M.E.: Sparse Bayesian Learning and the Relevance Vector Machines. Journal of Machine Learning Research. 1 (2001) 211-244
4. Xing, Z.D., Cao J.R.: Matrix Numerical Analysis. Science and Technology Press, Shannxi (1999)
5. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis. Springer-Verlag, New York (1993)
6. Smola, A. J., Scholkopf, B.: Sparse Greedy Matrix Approximation for Machine Learning. Proceedings of the Seventeenth International Conference on Machine Learning. Morgan-Kaufmann, San Mateo (2000) 911-918
7. Chow, E., Saad, Y.: Approximate Inverse Preconditioners via Sparse-Sparse Iterations. SIAM Journal of Scientific and Statistical Computing. 19 (1998) 955-1023