Fast Laplace Approximation for Sparse Bayesian Spike and Slab Models

Syed Abbas Zilqurnain Naqvi[1], Shandian Zhe[1], Yuan Qi[1] and Jieping Ye[2]
[1]Purdue University [2]University of Michigan, Ann Arbor

Introduction

- **Goal of the paper:** We develop approximate inference technique for the Bayesian spike-and-slab models for very high dimensional feature selection problems.
- **Challenge:** For very high dimensional problems, existing MCMC methods converge slowly; and the variational Bayes (VB) and expectation propagation (EP) approaches, unless they enforce structural constraints on the posterior, are impractical for large data.
- **Solution:** To address the computational issue, we develop the (FLAS) model. The features of our approach include:
  - FLAS is a hybrid of frequentist and Bayesian treatment, enjoying the benefits of both worlds. It is computationally as efficient as the frequentist methods.
  - It is free of any factorization assumptions on the joint posterior, but still enjoys a linear cost $O(np)$.
- **Evaluation:** Our new method FLAS performs feature selection better than or comparable to the alternative approximate methods with less running time, and provides higher prediction accuracy than various alternative sparse methods.

Model

The hierarchical Bayesian model for regression is:

$$p(\mathbf{t}|\mathbf{X},\mathbf{w},\tau) = \prod\nolimits_{i=1}^{n} \mathcal{N}(t_i|\mathbf{x}_i^\top \mathbf{w}, \tau^{-1}) \quad (1)$$

$$p(\mathbf{w}|\mathbf{z}) = \prod\nolimits_{j=1}^{p} \mathcal{N}(w_j|0,r_0)^{(1-z_j)} \mathcal{N}(w_j|0,r_1)^{z_j}, \quad (2)$$

$$p(z_j = 1|s_j) = s_j \quad (1 \le j \le p) \quad (3)$$

where $\mathbf{w}$ are regression weights, τ is the precision parameter, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$. z_j is a binary selection indicators for the j-th feature, and s_j is a selection probability with uninformative prior $p(s_j) = \mathrm{Beta}(a_0, b_0)$, with $a_0 = b_0 = 1$. For classification, $p(\mathbf{t}|\mathbf{X},\mathbf{w}) = \prod_{i=1}^{n} \sigma(\mathbf{x}_i^\top \mathbf{w})^{t_i}[1-\sigma(\mathbf{x}_i^\top \mathbf{w})]^{1-t_i}$ where $t_i \in \{0,1\}$, $\mathbf{w}$ are classifier weights, $\sigma(a) = 1/(1+\exp(-a))$, and $\mathbf{t} = [t_1, \ldots, t_n]^\top$.

MAP estimation for Laplace approximation

We use two scalable nonconvex optimization methods, L-BFGS and GIST . For L-BFGS(FLAS), we marginalize out both $\mathbf{z}$ and $\mathbf{s}$ and do the following optimization:

$$\min_{\mathbf{w},\mathbf{s}} \mathcal{F}(\mathbf{w}) = \min_{\mathbf{w}} L(\mathbf{w}) - \sum_{j=1}^{p} \log\left(\frac{1}{2}\mathcal{N}(w_j|0,r_1) + \frac{1}{2}\mathcal{N}(w_j|0,r_0)\right),$$

for GIST (FLAS*), we only marginalize out $\mathbf{z}$ and jointly optimize $\mathbf{w}$ and $\mathbf{s}$:

$$\min_{\mathbf{w},\mathbf{s}} \mathcal{F}(\mathbf{w},\mathbf{s}) = \min_{\mathbf{w}} L(\mathbf{w}) + \min_{\mathbf{s}} R(\mathbf{w},\mathbf{s}) \quad (4)$$

where $R(\mathbf{w},\mathbf{s}) = \sum_{j=1}^{p} R_j(w_j, s_j)$ and $R_j(w_j,s_j) = -\log\left(s_j\mathcal{N}(w_j|0,r_1) + (1-s_j)\mathcal{N}(w_j|0,r_0)\right)$.

Marginal Posterior variance estimation using Ensemble Nystrom

The inverse of Hessian for regression is approximated using Nyström method as:

$$\mathbf{H}^{-1} \approx \tilde{\mathbf{H}}^{-1}, \quad \tilde{\mathbf{H}} = \tau\mathbf{X}^\top\mathbf{X}_k(\mathbf{X}_k^\top\mathbf{X}_k)^\dagger\mathbf{X}_k^\top\mathbf{X} + \mathrm{diag}(\mathbf{v}). \quad (5)$$

$\mathbf{X}_k = [\mathbf{f}_{i_1}, \ldots, \mathbf{f}_{i_k}]$, where $\mathbf{f}_{i_t}$ is the i_t-th column of $\mathbf{X}$, and $v_j = -\frac{d^2\log(p(w_j))}{dw_j^2}\Big|_{w_j=\tilde{w}_j}$ for LBFGS and $v_j = -\frac{d^2\log(p(w_j,\bar{s}_j))}{dw_j^2}\Big|_{w_j=\tilde{w}_j}$ for GIST. Applying Woodbury matrix identity we can estimate the diagonal entries in $O(nkp)$ time:

$$\tilde{\mathbf{H}}^{-1} = \mathrm{diag}(\mathbf{v})^{-1} - \mathrm{diag}(\mathbf{v})^{-1}\mathbf{X}^\top\mathbf{X}_k(\tau^{-1}\mathbf{X}_k^\top\mathbf{X}_k + \mathbf{X}_k^\top\mathbf{X}\mathrm{diag}(\mathbf{v})^{-1}\mathbf{X}^\top\mathbf{X}_k)^{-1}\mathbf{X}_k^\top\mathbf{X}\mathrm{diag}(\mathbf{v})^{-1}.$$

Since we can choose $k \ll p$, the inversion cost will still be linear in p. For classification, $\mathbf{H} = \tilde{\mathbf{X}}^\top\tilde{\mathbf{X}} + \mathrm{diag}(\mathbf{v})$. Where $\tilde{\mathbf{X}} = \mathbf{X}\mathrm{diag}(\sqrt{\mathbf{b}})$, and $b_i = \sigma(\mathbf{x}_i^\top\tilde{\mathbf{w}})(1-\sigma(\mathbf{x}_i^\top\tilde{\mathbf{w}}))$. Rest of the procedure remains the same. To improve the accuracy, a simple ensemble approach is proposed. We sample d disjoint sets of columns of $\mathbf{X}$, each set is of the same size k. The estimation of the j-th diagonal entry of inverse Hessian is obtained by $\mathbf{H}^{-1}(j,j) \approx \frac{1}{d}\sum_{r=1}^{d}\tilde{\mathbf{H}}_r^{-1}(j,j)$, where $\tilde{\mathbf{H}}_r^{-1}$ is an approximation for the set r; time complexity is $O(npkr)$, $k, r << n, p$.

Theoretical analysis for Ensemble Nystrom

Theorem 1. Define $\Omega = \{\mathbf{A} \in \mathbb{R}^{p\times p}|\mathbf{A} \succ \mathbf{0}, \lambda_{min}(\mathbf{A}) \ge c, \lambda_{max}(\mathbf{A}) < \infty\}$. Assume Hessian $\mathbf{H}$ and approximate Hessian $\tilde{\mathbf{H}}$ both belong to $\mathbf{\Omega}$. Consider a function $f(\mathbf{A}) = \mathbf{e}_j^\top\mathbf{A}^{-1}\mathbf{e}_j, \mathbf{A} \in \mathbf{\Omega}$. Then, $\|\nabla f(\mathbf{A})\|_F \le L$, $(1-\eta)\mathbf{H} + \eta\tilde{\mathbf{H}} \in \mathbf{\Omega}\ \forall\ \eta \in [0,1]$, and with high probability,

$$|\mathbf{H}^{-1}(j,j) - \tilde{\mathbf{H}}^{-1}(j,j)| \le L \cdot D_0$$

where c is a small positive constant, and $L = p/c^2$. $\mathbf{e}_j$ is a standard basis vector with 1 in j-th coordinate and 0's elsewhere, and D_0 is the standard Nyström error bound based on Frobenius norm.

Theorem 2. Define $\Omega = \{\mathbf{A} \in \mathbb{R}^{p\times p}|\mathbf{A} \succ \mathbf{0}, \lambda_{min}(\mathbf{A}) \ge c, \lambda_{max}(\mathbf{A}) < \infty\}$. Assume Hessian $\mathbf{H}$ and a set of approximate Hessians $\{\tilde{\mathbf{H}}_1, \ldots, \tilde{\mathbf{H}}_d\}$ all belong to $\mathbf{\Omega}$, then with high probability,

$$|\mathbf{H}^{-1}(j,j) - \frac{1}{d}\sum_{r=1}^{d}\tilde{\mathbf{H}}_r^{-1}(j,j)| \le L \cdot D_1$$

where D_1 the error bound for ensemble Nyström based on Frobenius norm. Because $D_1 < D_0$, the ensemble approach has a smaller error bound.

Proposition 1. Assume that $\lambda_{max}(\mathbf{X}^\top\mathbf{X}) < \infty$, and $\forall j\ b \le v_j < \infty$, where b is a small positive constant. Then both Hessian $\mathbf{H}$ and any approximate Hessian $\tilde{\mathbf{H}}$ based on Nyström method belong to $\mathbf{\Omega}_0 = \{\mathbf{A} \in \mathbb{R}^{p\times p}|\mathbf{A} \succ \mathbf{0}, \lambda_{min}(\mathbf{A}) \ge b, \lambda_{max}(\mathbf{A}) < \infty\}$, and hence theorems 1 and 2 are satisfied with $L = p/b^2$

Bayesian inference of s_j, z_j

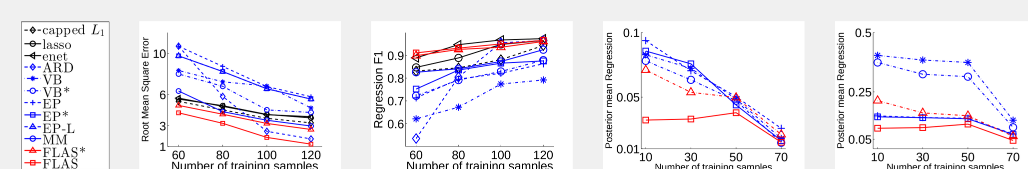
Posterior moments calculated, in $O(1)$ time, by Gauss-Hermite quadrature

$$\mathrm{E}[s_j] = \int \frac{2\mathcal{N}_1(w_j) + \mathcal{N}_0(w_j)}{3(\mathcal{N}_1(w_j) + \mathcal{N}_0(w_j))} q(w_j)\mathrm{d}w_j, \quad \mathrm{Var}[s_j] = \int \frac{3\mathcal{N}_1(w_j) + \mathcal{N}_0(w_j)}{6(\mathcal{N}_1(w_j) + \mathcal{N}_0(w_j))} q(w_j)\mathrm{d}w_j - \mathrm{E}^2[s_j]$$

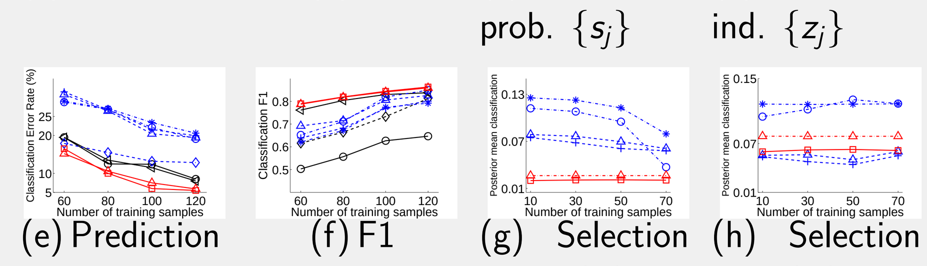
$$\mathrm{E}[z_j] = \int \frac{\mathcal{N}_1(w_j)}{\mathcal{N}_1(w_j) + \mathcal{N}_0(w_j)} q(w_j)\mathrm{d}w_j, \quad \mathrm{Var}[z_j] = \int \frac{\mathcal{N}_1(w_j)}{\mathcal{N}_1(w_j) + \mathcal{N}_0(w_j)} q(w_j)\mathrm{d}w_j - \mathrm{E}^2[z_j].$$

where $\mathcal{N}_g(w_j) = \mathcal{N}(w_j|0, r_g)$ (for $g = 0, 1$) and $q(w_j) = \mathcal{N}(w_j|m_j, \sigma_j^2)$.

Experimental results on simulation and real data



(a) Prediction (b) F1 (c) Selection prob. $\{s_j\}$ (d) Selection ind. $\{z_j\}$



(e) Prediction (f) F1 (g) Selection prob. $\{s_j\}$ (h) Selection ind. $\{z_j\}$

Figure: Simulation results, with $p = 1000$ and only 20 relevant features, for the prediction accuracy, the F1 score of feature selection. Results for the root mean squared error for the posterior mean estimation of $\{s_j\}$ and $\{z_j\}$ were obtained for $p = 100$, used Gibbs sampling as a gold standard. Results are averaged over 50 runs.

Table: The training time (seconds) on simulated data ($p = 1000$).

(a) Regression

method	60	80	100	120
capped L_1	**0.0054**	0.0705	**0.0103**	**0.0108**
lasso	0.0312	0.0313	0.0321	0.0329
elastic net	0.0360	0.0346	0.0352	0.0349
ARD	0.03	0.17	0.20	0.67
VB	2.4161	2.3999	2.4794	2.4404
VB*	2.7544	3.0118	2.6012	2.7795
EP	0.9345	1.0478	1.1160	1.1468
EP*	0.505	0.681	1.047	1.936
MM	2.5230	1.1047	0.4314	0.5282
FLAS*	0.0664	0.0642	0.0704	**0.0855**
FLAS	**0.0140**	**0.0154**	**0.0216**	1.4526

(b) Classification

method	60	80	100	120
capped L_1	0.0180	0.0499	0.0427	0.0559
lasso	0.1033	0.1289	0.1555	0.1821
elastic net	0.08690	0.1009	0.1163	0.1356
ARD	0.06	0.07	0.15	0.45
VB	10.3312	11.2570	12.3317	13.3366
VB*	0.0812	0.1570	2.8915	3.0194
EP	1.1165	1.1695	1.2400	1.3090
EP-L	0.0132	0.0581	0.0598	0.1631
FLAS*	0.0344	0.0736	0.0794	0.1929
FLAS	**0.0097**	**0.0111**	**0.0139**	**0.0152**

Table: Root mean square error on regression datasets (the first 6 rows) and classification error rates (%) on large binary classification datasets (the last 8 rows). Note that EP-L is designed for classification task only and thus does not have results on the regression datasets. The results are averaged over 10 runs.

dataset	lasso	elast net	capped L_1	ARD	EP-L	FLAS*	FLAS
gse5680	0.107 ± 0.003	0.107 ± 0.003	0.107± 0.003	0.136 ± 0.005	NA	0.122 ± 0.002	**0.089± 0.002**
10k corpus	0.382 ± 0.002	0.382 ± 0.002	0.382 ± 0.002	0.382 ± 0.384	NA	0.383 ± 0.003	**0.372 ± 0.003**
tied	0.656 ± 0.013	0.627 ± 0.014	0.656 ± 0.013	**0.532 ± 0.017**	NA	0.719 ± 0.012	0.656 ± 0.013
House	1.576 ± 0.011	1.578 ± 0.017	1.587 ± 0.012	0.435 ±0.0006	NA	0.561 ± 0.015	**0.425 ±0.002**
Year	0.296 ± 0.009	0.293 ± 0.007	0.307 ± 0.004	0.306 ± 0.006	NA	0.248 ± 0.0005	**0.234 ± 0.0001**
dlbcl	1.76 ± 0.026	1.75 ± 0.027	1.75 ± 0.028	2.38 ± 0.063	NA	**1.60 ± 0.047**	**1.60± 0.047**
classic	6.69 ± 0.002	5.94 ± 0.002	**4.14 ± 0.002**	18.2 ± 0.002	8.94 ± 0.002	5.76 ± 0.002	4.20 ± 0.001
hitech	23.2 ± 0.005	21.4 ± 0.004	21.3 ± 0.003	28.5 ± 0.019	25.2 ± 0.001	**19.4 ± 0.003**	19.9 ± 0.003
k1b	5.44 ± 0.005	4.91 ± 0.004	**4.42 ± 0.004**	23.0 ± 0.013	7.94 ± 0.004	5.03 ± 0.005	4.74 ± 0.005
reviews	7.68 ± 0.003	6.47 ± 0.002	6.09 ± 0.001	35.4 ± 0.05	8.28 ± 0.002	5.93 ± 0.002	**5.54 ± 0.001**
sports	3.72 ± 0.001	3.15 ± 0.0008	3.25 ± 0.0009	24.1 ± 0.032	10.9 ± 0.008	2.78 ±0.001	**2.77 ± 0.007**
ng3sim	19.3 ± 0.005	16.2 ± 0.003	15.4 ± 0.003	21.3 ± 0.006	14.5 ± 0.002	13.7 ± 0.003	**13.6 ± 0.002**
ohscal	13.8 ± 0.001	13.7 ± 0.001	13.8 ± 0.001	37.3 ± 0.02	13.7 ±0.002	**11.9 ± 0.001**	13.1 ± 0.001
la12	13.6 ± 0.002	12.5 ± 0.002	12.2 ± 0.002	30.1 ± 0.025	13.2 ± 0.002	11.1 ± 0.002	**11.1 ± 0.001**

{naqvi,szhe,alanqi}@purdue.edu, jpye@umich.edu