Mixture of Bayesian SVM Experts

Presentation for EE491A

Archit Sharma*, Dr. Piyush Rai#

*Department of Electrical Engineering, IIT Kanpur
#Department of Computer Science and Engineering, IIT Kanpur

Table of Contents

Introduction

SVM

- *Support Vector Machines* (SVM) [1] are extremely popular algorithms for binary classification
- Can be extended to regression, multi-class classification and non-linear learning.
- Hyperparamaters difficult to tune when using Kernels.
- Prone to overfitting.

The SVM objective can be written as

$$\mathcal{L}(w, R) = \sum_{i=1}^{N} \max(1 - y_i x_i^T w, 0) + R(w)$$

which needs to be minimized for estimating w.

Bayesian SVM was formulated by Polson et al. [4]. They showed that

$$\exp(-2\max(1 - y_i x_i^T w, 0)) = \int_0^\infty \frac{1}{\sqrt{2\pi\gamma_i}} \exp(-\frac{1}{2}\frac{(1+\gamma_i - y_i x_i^T w)^2}{\gamma_i}) d\gamma_i$$

This inspired that data augmentation scheme, which allows casts SVM objective into a posterior maximization/estimation problem, something which is well studied by the Bayesians.

Mixture of Experts

Mixture of Experts [2] is powerful framework, which essentially pools the effort of relatively simple experts to model a harder problem. There are two components for a mixture of experts framework:

- *Experts*: Local learner, usually simple, models a subset of input data.
- *Gating Network*: Maps the input to the expert.

These models are trained using Expectation Maximization (EM), as the input-expert assignment is not known.

Mixture of Bayesian SVMs

Mixture of Bayesian SVMs

Bayesian SVMs are proposed as local learners in a Mixture of Experts model in this work. Three gating network architectures are experimented with:

- Softmax Gating Network
- Generative Gating Network
- Polya-Gamma augmented Softmax Gating Network

The proposed model is trained using Expectation Maximization (EM). MCMC routines are easy to derive. Assume $W = \{w_i\}_{i=1}^{K}$, as the K Bayesian SVM experts.

Softmax Gating Network

The most naive architecture for Softmax Gating Network, as proposed in [2]. Assume $V = \{v_i\}_{i=1}^K$ softmax gating vectors. The probability of input x_i being assigned to expert j is given by

$$\pi_j(x_i) = \frac{\exp(v_j^T x_l)}{\sum_{l=1}^K \exp(v_l^T x_i)}$$

Major Drawback: *No closed form updates for the gating vector.*

Softmax Gating Network

EM Algorithm

E Step:

$$\eta_{ij} \leftarrow \exp(x_i^T v_j - 2\max(0, 1 - y_i x_i^T w_j))$$

$$\eta_{ij} \leftarrow \frac{\eta_{ij}}{\sum_{l=1}^{K} \eta_{il}}$$

$$\tau_{ij} \leftarrow |1 - y_i x_i^T w_j|^{-1}$$

M Step:

$$A_j \leftarrow diag(\frac{\eta_{1j}}{\tau_{1j}} \dots \frac{\eta_{Nj}}{\tau_{Nj}})$$

$$w_j \leftarrow (X^T A_j X + \lambda I)^{-1}(\sum_{i=1}^{N} \eta_{ij} \frac{\tau_{ij} + 1}{\tau_{ij}} y_i x_i)$$

$$Iterate : v_j \leftarrow v_j - \alpha(\sum_{i=1}^{N} \left[\eta_{ij} - \frac{\exp(v_j^T x_i)}{\sum_{l=1}^{K} \exp(v_l^T x_i)}\right] x_i - \beta v_j)$$

Generative Gating Network

Generative gating network makes two major changes:

- *Gating network*: Each gate models the input (much like Gaussian Mixture Models). The gating parameters are $\{\alpha_k, \mu_k, \Sigma_k\}_{k=1}^K$ and $\pi_j(x_i) \propto \alpha_j \mathcal{N}(x_i|\mu_j, \Sigma_j)$.
- *Changed Objective*: The model now maximizes $\mathbb{E}[\log p(y, X, \gamma, z|\Theta)]$ instead of $\mathbb{E}[\log p(y, \gamma, z|X, \Theta)]$. Here, γ, z are latent variables corresponding to Bayesian SVM and input-expert assignment.

Drawbacks: The number of parameters has increased significantly. The objective proposed be solved is harder, and indirect to what we are interested in.

Generative Gating Network

EM Algorithm

E Step:

$$\eta_{ij} \leftarrow \exp(-2\max(0, 1 - y_i x_i^T w_j))\mathcal{N}(x_i|\mu_j, \Sigma_j)\alpha_j$$
$$\eta_{ij} \leftarrow \frac{\eta_{ij}}{\sum_{l=1}^{K} \eta_{il}}$$
$$\tau_{ij} \leftarrow |1 - y_i x_i^T w_j|^{-1}$$

M Step:

$$A_j \leftarrow diag(\frac{\eta_{1j}}{\tau_{1j}} \dots \frac{\eta_{Nj}}{\tau_{Nj}})$$

$$w_j \leftarrow (X^T A_j X + \lambda I)^{-1}(\sum_{i=1}^{N} \eta_{ij} \frac{\tau_{ij} + 1}{\tau_{ij}} y_i x_i)$$

$$\alpha_j \leftarrow \frac{\sum_{i=1}^{N} \eta_{ij}}{N}, \ \mu_j \leftarrow \frac{\sum_{i=1}^{N} \eta_{ij} x_i}{\sum_{i=1}^{N} \eta_{ij}}, \ \Sigma_j \leftarrow \frac{\sum_{i=1}^{N} \eta_{ij}(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{N} \eta_{ij}}$$

Polya-Gamma Augmentation

Polya-Gamma augmentation [3, 5] augments latent variables β to give a Bayesian treatment to logistic regression. In particular, [3] shows that

$$\frac{\left(e^{\psi}\right)^{a}}{\left(1+e^{\psi}\right)^{b}} = 2^{-b} e^{(a-b/2)\psi} \int_{0}^{\infty} e^{-\beta\psi^2/2} p(\beta) d\beta$$

This augmentation can be extended to multinomial regression. We can get closed for updates for softmax gating networks using this. Drawbacks: Multiple augmentation, potentially unstable and more initialization dependent.

EM Algorithm

E Step:

$$\eta_{ij} \leftarrow \exp(x_i^T v_j - 2\max(0, 1 - y_i x_i^T w_j))$$

$$\psi_{ij} \leftarrow x_i^T v_j - \log \sum_{l=1, l\neq j}^{K} \exp(x_i^T v_l)$$

$$\eta_{ij} \leftarrow \frac{\eta_{ij}}{\sum_{l=1}^{K} \eta_{il}},\ \ \tau_{ij} \leftarrow |1 - y_i x_i^T w_j|^{-1},\ \ \beta_{ij} \leftarrow \frac{1}{2\psi_{ij}} \tanh(\psi_{ij})$$

M Step:

$$A_j \leftarrow diag(\frac{\eta_{nj}}{\tau_{nj}})_{n=1}^{N},\ \Omega_j \leftarrow diag(\beta_{nj}\eta_{nj})_{n=1}^{N}$$

$$\kappa_j^T \leftarrow [\eta_{nj}(\frac{1}{2} + \beta_{nj} \log \sum_{l=1, l\neq j}^{N} \exp(x_n^T \hat{v}_l))]_{n=1}^{N}$$

$$w_j \leftarrow (X^T A_j X + \lambda I)^{-1}(\sum_{i=1}^{N} \eta_{ij} \frac{\tau_{ij}+1}{\tau_{ij}} y_i x_i),\ \ v_j \leftarrow (X^T \Omega_j X)^{-1} X^T \kappa_j$$

Results

Some Results

Table 1: **LR**: Logistic Regression, **SVM**: Support Vector Machine with RBF Kernel, **SS-ζ (T=5)**: SS-softplus regression with $K_{max} = 20$ and $T = 5$ [6], **M-GG**: Mixture of Bayesian SVM Experts with Generative Gating, **M-PG**: Mixture of Bayesian SVM experts with Polya-Gamma augmented Softmax gating Networks.

Dataset	LR	SVM	SS-ζ	M-GG	M-PG
banana(3)	43.38	10.28	11.28	**9.43**	17.72
breast cancer(10)	24.37	23.92	25.43	**18.19**	19.49
titanic(4)	21.69	21.7	21.49	**20.73**	21.31
waveform(22)	12.74	9.87	11.0	12.92	**8.14**
german(21)	21.93	20.79	21.77	19	**18.33**
image(19)	16.48	2.32	**2.2**	3.87	9.41

Discussion

- Results strongly support this formulation
- More analysis for PG and GG gating networks required.
- Explore different extensions: Regression, Multiclass Classification, Kernel learning

Questions?

References i

C. Cortes and V. Vapnik.
Support-vector networks.
Mach. Learn., 20(3):273–297, Sept. 1995.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton.
Adaptive mixtures of local experts.
Neural computation, 3(1):79–87, 1991.

N. G. Polson, J. G. Scott, and J. Windle.
Bayesian inference for logistic models using pólya–gamma latent variables.
Journal of the American statistical Association, 108(504):1339–1349, 2013.

N. G. Polson, S. L. Scott, et al.
Data augmentation for support vector machines.
Bayesian Analysis, 6(1):1–23, 2011.

J. G. Scott and L. Sun.
Expectation-maximization for logistic regression.
arXiv preprint arXiv:1306.0040, 2013.

M. Zhou.
Softplus regressions and convex polytopes.
arXiv preprint arXiv:1608.06383, 2016.