# Probabilistic Approaches for Sparse Modeling

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

# Recap: Probabilistic Linear Regression

- Assume training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, with features $\boldsymbol{x}_n \in \mathbb{R}^D$ and responses $y_n \in \mathbb{R}$
- Likelihood model

Equivalent to a noisy linear Gaussian model: $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \beta^{-1}\mathbf{I_N})$

But now it's also a hyperparameter which can be learned

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^N \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1}) \quad = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N)$$

Imp: Precision/variance controls extent of regularization

- Prior distribution on $\boldsymbol{w}$: $p(\boldsymbol{w}) = \prod_{d=1}^D p(w_d) = \prod_{d=1}^D \mathcal{N}(w_d|0, \lambda^{-1}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D) = \left(\frac{\lambda}{2\pi}\right)^{\frac{D}{2}} \exp\left[-\frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}\right]$
- Posterior distribution on $\boldsymbol{w}$: $p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) = \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$

Assuming $\beta, \lambda$ are known

Posterior's mean has the same expression as MAP solution (also similar form as ridge regression)

$$\boldsymbol{\Sigma}_N = (\beta \sum_{n=1}^N \boldsymbol{x}_n \boldsymbol{x}_n^\top + \lambda \mathbf{I}_D)^{-1} = (\beta \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1} \qquad \boldsymbol{\mu}_N = \boldsymbol{\Sigma}_N \left[\beta \sum_{n=1}^N y_n \boldsymbol{x}_n\right] = \boldsymbol{\Sigma}_N \left[\beta \mathbf{X}^\top \boldsymbol{y}\right] = (\mathbf{X}^\top \mathbf{X} + \frac{\lambda}{\beta}\mathbf{I}_D)^{-1} \mathbf{X}^\top \boldsymbol{y}$$

- Posterior Predictive distribution

Variance from noisy Gaussian likelihood/observation model

Predictive variance due to uncertainty in $\boldsymbol{w}$ (also input-dependent)

But we also have posterior covariance (thus uncertainty estimate). Also, in this Bayesian set-up, we can also estimate $\lambda$ and $\beta$ from data

$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \int p(y_*|\boldsymbol{w}, \boldsymbol{x}_*, \beta) p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) d\boldsymbol{w} = \mathcal{N}(\boldsymbol{\mu}_N^\top \boldsymbol{x}_*, \beta^{-1} + \boldsymbol{x}_*^\top \boldsymbol{\Sigma}_N \boldsymbol{x}_*)$$

Increases as we move far away from training inputs

- Hyperparams can be chosen via cross-valid, or learned from training data, e.g., point estimation based on maximizing the marginal likelihood or marginal posterior of hyperparams, or by inferring (approximately) the joint post of $\boldsymbol{w}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$)

# Sparse Modeling

- Often the model (or various parts of the model) defined by of a set of weights
- Example: Some output defined as a weighted comb. of a set of inputs, e.g.,

Output of a linear model with $D$ features

In a deep net, the pre-activation of each hidden unit also has the same form (a linear model)

$$y_* = \sum_{d=1}^{D} w_d x_{*d}$$

Output of a kernelized model with $N$ training examples

$$y_* = \sum_{n=1}^{N} w_n k(\boldsymbol{x}_n, \boldsymbol{x}_*)$$

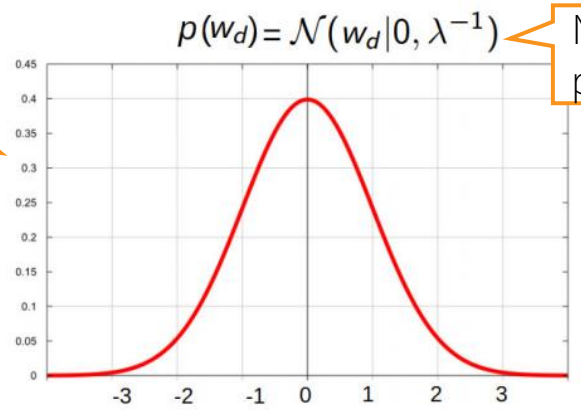
- Often desirable to have only very few of these weights as nonzero. Benefits:
  - Can help in automatic feature selection (nonzero wts only for relevant features)
  - Can help identify which training inputs are relevant (see the above kernelized model)
  - Can help in model compression (fewer weights to store since zero weights need not be stored; useful in massive-sized models, such as deep neural networks)
- Can use sparsity-promoting regularizers such as $\ell_1$ or $\ell_0$ to learn sparsity
- Alternatively, with a probabilistic approach, can use suitable priors on weights

# Probabilistic Approaches for Sparse Modeling

- Recall the standard zero-mean Gaussian prior on each scalar-valued weight $w_d$

Promotes the weight $w_d$ to be small but not so much

Also, can't get exact zero for the weight with such a prior




Note: Can even use separate precisions/variances for priors of different $w_d$'s

- Some alternatives
  - Spike-and-slab mixture based priors

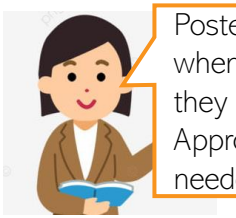
$w_d$, a priori, is zero with prob $\pi$ and drawn from a zero mean Gaussian with prob $1 - \pi$

Inference (even point estimation) is usually intractable. Usually MCMC or variational inference or greedy methods are required

$$p(w_d) = \pi\delta(w_d) + (1 - \pi)\mathcal{N}(w_d|0, \lambda^{-1})$$

  - Priors that are more sharply peaked around zero than a zero-mean Gaussian

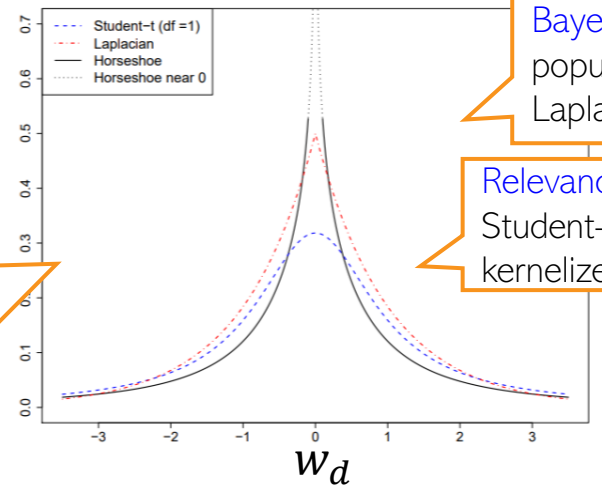
# Priors With Sharper Peaks Around Zero



Posterior harder to compute when using such priors since they are not in exp-family. Approx. inference methods needed (will see later)

- There are various distributions that have a sharper peak than a Gaussian

$p(w_d)$




Bayesian LASSO is a popular sparse model with Laplace prior on weights

Relevance Vector Machine uses Student-t prior on weights of a kernelized model

Horseshoe prior also used in Bayesian neural nets to get sparse weights (see "Model Selection in Bayesian Neural Networks via Horseshoe Priors" by Ghosh et al, 2017)

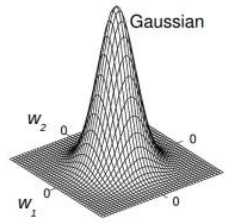
$$p(w_d|\lambda_d) = \mathcal{N}(0, \lambda_d^{-1})$$

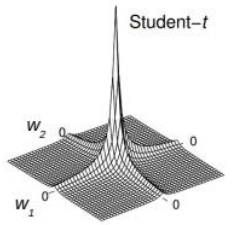
Various possibilities (e.g., gamma, Cauchy, exp, etc)

$$p(w_d) = \int p(w_d|\lambda_d)p(\lambda_d)d\lambda_d$$

Zero-mean Gaussian

Marginal prior no longer a Gaussian but some other sharper-peaked distribution around zero




- These can often be defined as "scale-mixture" of Gaussians
  - Scale-mixture: Take a Gaussian and integrate out its scale (variance/precision) parameter
  - Important: Using a scale-mixture instead of directly using a sparse marginal prior (e.g., Laplace) makes posterior inference easier (since we still have a Gaussian conditional prior) when doing approx. inference
- Depending on the prior on $\lambda_d$, different types of sparse priors obtained, e.g.,
  - A gamma prior on $\lambda_d$ results in $p(w_d)$ being a Student-t/Laplace distribution
  - A Cauchy prior $\lambda_d$ results in $p(w_d)$ being a Horseshoe distribution

e.g., EM, Gibbs sampling, variational inference, etc

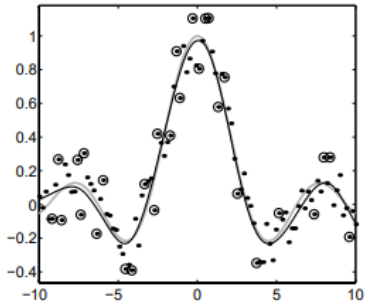
Figure courtesy: Handling Sparsity via the Horseshoe (Carvalho et al, 2009)

# An Illustration of Sparse Prior: RVM

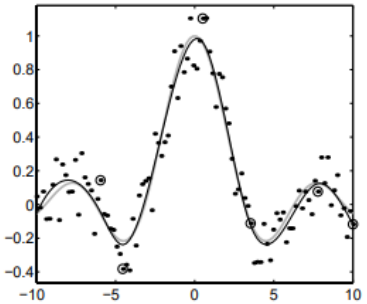
- Relevance Vector Machine (RVM) sparsifies kernel methods which predict as

$$y_* = \sum_{n=1}^{N} w_n k(\boldsymbol{x}_n, \boldsymbol{x}_*)$$

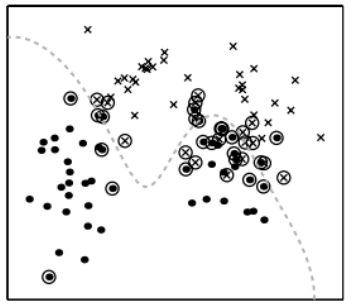
- For standard kernel methods, prediction cost is $O(N)$ for $N$ training inputs
- When using SVMs, $\boldsymbol{w_n}$'s are nonzero only for the support vectors but often there are still a significant number of S.V.s
- RVM explicitly encourages $\boldsymbol{w_n}$'s to be sparse using a Student-t distribution

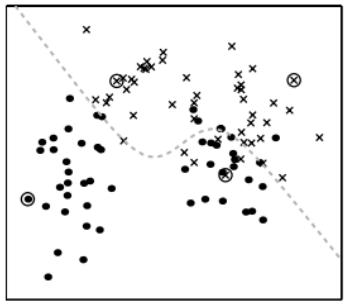


Regression using Kernelized Support Vector Regression (SVR)



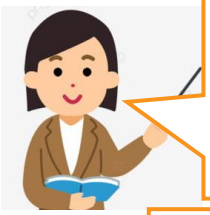
Regression using RVM





Classification using Kernelized Support Vector Machine (SVM)

Classification using RVM



Can think of RVM as a probabilistic/Bayesian analogue of SVMs with many benefits (can learn posterior, PPD, hyperparams, impose sparsity on weights, etc)

Gaussian Process (GP) is another such powerful Bayesian alternative to SVMs but doesn't have sparsity explicitly (but ways exist to make GPs sparse), Will study GPs later

Figure courtesy: Sparse Bayesian Learning and the Relevance Vector Machine (Tipping, 2001)

# Coming Up

- Probabilistic Models for Classification
- Laplace Approximation of the posterior