Modeling sparsity in classical and deep latent variable models

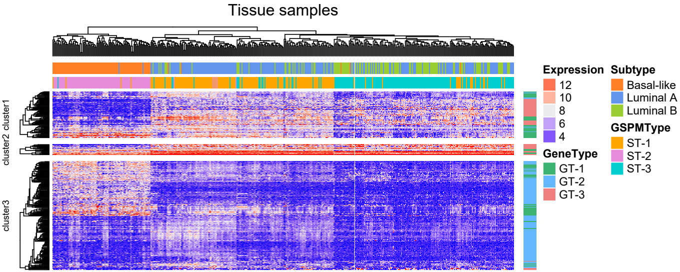
Clint P. George

School of Mathematics and Computer Science
Indian Institute of Technology Goa

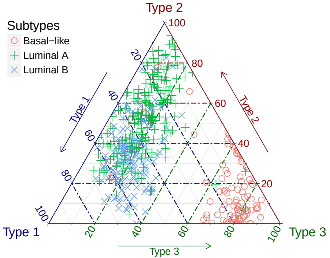
November 8, 2023

Introduction

Modeling sparsity — gene expressions



- Given $(\boldsymbol{x}_i, y_i), i = 1, \ldots, n$, select a subset of features $(x_1, x_2, \ldots, x_p)$
- Interpretability



Understanding or interpreting data

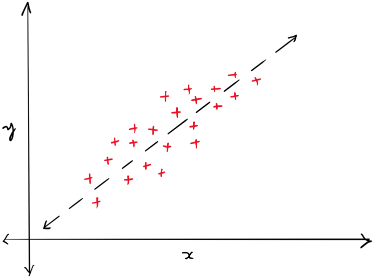
- We have some measurements of some properties from two instruments.
- Interpretation: search for a pattern—e.g., one instrument consistency measures higher

Understanding or interpreting data

- We have some measurements of some properties from two instruments.
- Interpretation: search for a pattern—e.g., one instrument consistency measures higher
- *Statistical modeling*
  - systematic effects — aims to summarize data
  - random effects — aims to summarize the nature and magnitude of unexplained or random variation

Modeling patterns

- Goal: generate patterns of numbers that can replace the data at some point

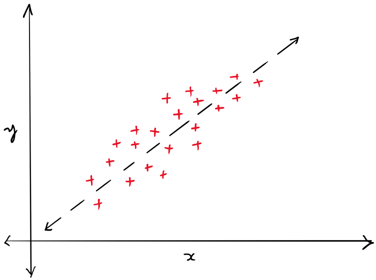


Modeling patterns

- Goal: generate patterns of numbers that can replace the data at some point
- Consider a simple model[a]

$$y = \beta x + \alpha$$

  - Connects y and x via the parameter pair (α, β)
  - Models straight-line relationship between y and x



[a] dates back to Gauss and Legendre's work on astronomical data

Modeling patterns

- If we have $x_1, x_2, \ldots, x_n$, given (α, β), y takes the values $\beta x_1 + \alpha, \beta x_2 + \alpha, \ldots, \beta x_n + \alpha$.
- In practice, y has measurement error and the relation x–y is *approximately* linear

$$y = \beta x + \alpha + \epsilon$$

Statistical modeling of patterns[1]

- The observation vector $\boldsymbol{y}$ with n components $y_1, y_2, \ldots, y_n$ is a realization of a r.v. $\boldsymbol{Y}$, whose components are independently distributed with means $\boldsymbol{\mu}$

$$\boldsymbol{\mu} = \sum_{j=1}^{p} \boldsymbol{x}_j \beta_j,$$

where β_js are unknown parameters. And,

$$E[Y_i] = \mu_i = \sum_{j=1}^{p} x_{ij}\beta_j; i = 1, 2, \ldots, n$$

- The errors follow a Gaussian with constant variance σ^2

[1] McCullagh and Nelder (1989). Generalized Linear Models

Estimating $\boldsymbol{\beta}$

- Maximize the likelihood of the parameters for the observed data
- Let $f(y_i; \boldsymbol{\beta})$ be the density for observation y_i given $\boldsymbol{\beta}$, then

$$\mathcal{L}(\boldsymbol{\mu}; \boldsymbol{y}) = \sum_{i=1}^{n} \log f(y_i; \boldsymbol{\beta})$$

- Assuming normality with constant variance,

$$\mathcal{L}(\mu_i; y_i) = \frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \underbrace{(y_i - \mu_i)^2}_{\text{residual squares}},$$

for observation i

Shrinkage methods

Ridge regression

- Shrinks the regression coefficients by imposing a penalty[2].

$$\hat{\boldsymbol{\beta}}_{\textsf{ridge}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \underbrace{\sum_{j=1}^{p} x_{ij}\beta_j}_{\mu_i} \right)^2 + \underbrace{\lambda \sum_{j=1}^{p} {\beta_j}^2}_{\text{penalty term}} \right\}, \lambda \geq 0$$

[2] Hoerl & Kennard (1970). *Ridge regression: Biased estimation for ...*

Ridge regression

- Shrinks the regression coefficients by imposing a penalty[2].

$$\hat{\boldsymbol{\beta}}_{\textsf{ridge}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \underbrace{\sum_{j=1}^{p} x_{ij}\beta_j}_{\mu_i} \right)^2 + \underbrace{\lambda \sum_{j=1}^{p} {\beta_j}^2}_{\text{penalty term}} \right\}, \lambda \geq 0$$

- Solution is a linear function of $\boldsymbol{y}$

$$\hat{\boldsymbol{\beta}}_{\textsf{ridge}} = \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda \mathsf{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{1}$$

$\boldsymbol{X}$ is standardized $n \times p$ matrix.

[2] Hoerl & Kennard (1970). *Ridge regression: Biased estimation for ...*

LASSO regression[3]

- The penalty term is different

$$\hat{\boldsymbol{\beta}}_{\text{LASSO}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \underbrace{\sum_{j=1}^{p} x_{ij}\beta_j}_{\mu_i} \right)^2 + \lambda \underbrace{\sum_{j=1}^{p} |\beta_j|}_{\text{penalty term}} \right\}, \lambda \geq 0$$

- The solution is not a linear function of $\boldsymbol{y}$
- It can threshold some coefficients to zero.

[3] Tibshirani (1996). The least absolute shrinkage and selection operator.

$$\min \sum_{i=1}^{n} (y_i - \mu_i)^2 \qquad \text{such that} \quad \begin{cases} \sum_{j=1}^{p} {\beta_j}^2 \leq t & \text{ridge} \\ \\ \sum_{j=1}^{p} |\beta_j| \leq t & \text{lasso} \end{cases}$$

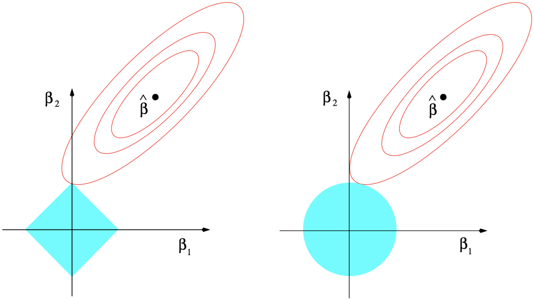
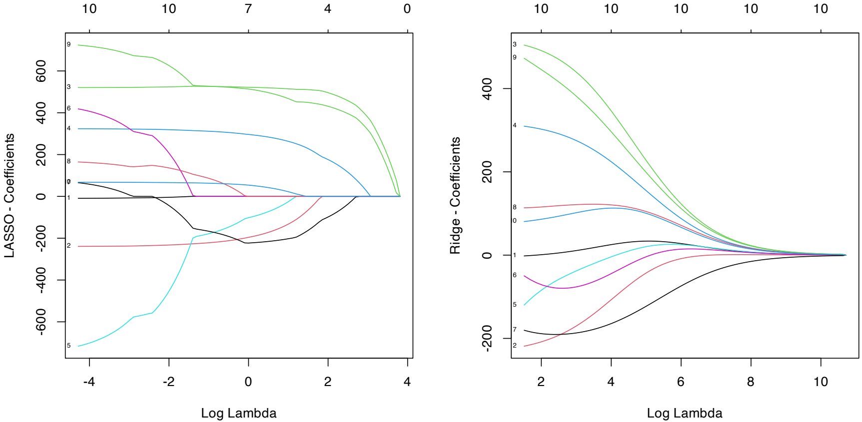


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Hastie et al. (2009, ESL)



Diabetes data (Efron et al. 2004) — 442 samples, 10 features

Bayesian approach

Bayes theorem

$$p(\beta|x) = \frac{p(x|\beta)p(\beta)}{p(x)} \propto p(x|\beta)p(\beta)$$

Bayesian approach

Bayes theorem

$$p(\beta|x) = \frac{p(x|\beta)p(\beta)}{p(x)} \propto p(x|\beta)p(\beta)$$

where

$p(\beta\|x)$	posterior
$p(x\|\beta)$	likelihood
$p(\beta)$	prior

Bayesian ridge regression

- Coefficients $\boldsymbol{\beta}$ have the prior

$$p(\boldsymbol{\beta}|\alpha) = N(\boldsymbol{\beta}|0, \alpha^{-1}I) \propto \frac{\alpha}{2\pi}^{M/2} \exp\left\{\frac{-\alpha}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right\}$$

- Find $\boldsymbol{\beta}$: the most probable value of $\boldsymbol{\beta}$ given the data—i.e., maximize the posterior (MAP)

Bayesian ridge regression

- Coefficients $\boldsymbol{\beta}$ have the prior

$$p(\boldsymbol{\beta}|\alpha) = N(\boldsymbol{\beta}|0, \alpha^{-1}I) \propto \frac{\alpha}{2\pi}^{M/2} \exp\left\{\frac{-\alpha}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right\}$$

- Find $\boldsymbol{\beta}$: the most probable value of $\boldsymbol{\beta}$ given the data—i.e., maximize the posterior (MAP)
- Maximizing the log-posterior is equivalent to minimizing

$$\sum_{i=1}^{n}(y_i - \mu_i)^2 + \frac{\alpha}{2}\sum_{j=1}^{p}{\beta_j}^2$$

Bayesian LASSO

- Lasso minimizes
$$\sum_{i=1}^{n}(y_i-\mu_i)^2+\frac{\lambda}{2}\sum_{j=1}^{p}|\beta_j|$$
- Lasso estimates as MAP estimates when $\boldsymbol{\beta}$ have the priors[4]
$$p_\tau(\boldsymbol{\beta})=\left(\frac{\tau}{2}\right)^p\exp(-\tau\|\boldsymbol{\beta}\|_1)$$
and the data likelihood is
$$p_\sigma(\boldsymbol{y}|\boldsymbol{\beta})=N(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{\beta},\sigma^2\mathsf{I})$$

[4] Tibshirani 1996

Spike and slab priors

- Variable selection under the normal linear model; Bayesian LASSO is ineffective[5]
- Coefficients $\boldsymbol{\beta}$ have Spike and Slab priors[6]

$$\beta_j \sim (1-\gamma_j) \underbrace{\delta_0}_{\text{spike}} + \gamma_j \underbrace{p(\beta_j|\tau^2)}_{\text{slab}}$$

$$\gamma_j \sim \text{Bernoulli}(\lambda)$$

[5] Ghosh et al. (2016), Castilo et al. (2015)
[6] Lempers (1971), Mitchel & Beauchamp (1988), George & McCullagh (1993)

Spike and slab priors

- This prior is considered ideal for sparse Bayesian problems[7]

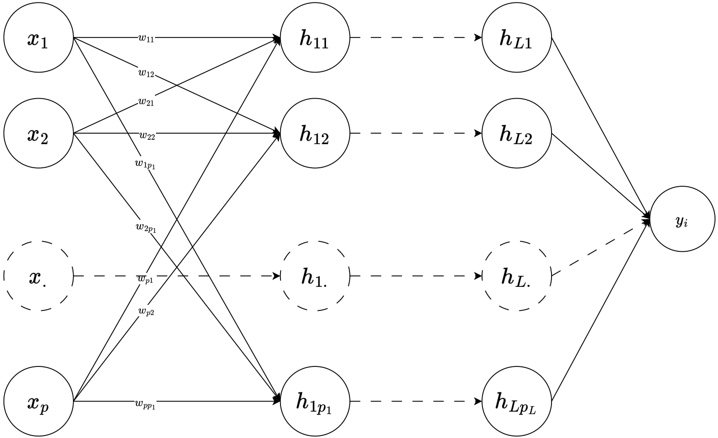
- Exploring the full posterior over the entire model space can be challenging due to the combinatorial complexity of updating discrete indicators $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_p)$
- Solutions in the literature — stochastic search, variational inference

[7] Bai et al. (2020)

Sparse deep learning

- Deep neural networks can model complex patterns
- Network compression, before deployment to tiny devices
- Variable selection

Deep neural network



$$y_i = f_\theta(\vec{x}_i) + \epsilon_i; \quad \epsilon \sim N(0, \sigma^2)$$

Weights w are typically ON all the time

Deep neural network — formal representation

- We model data via L-hidden layer network; each layer l has p_l neurons/nodes
- The weight matrix and bias vector in each layer $l = 1, 2, \ldots, L$ are

$$W_i \in \mathbb{R}^{p_{l-1} \times p_l}, \quad \boldsymbol{b}_i \in \mathbb{R}^{p_l},$$

  which we denote by $\boldsymbol{\theta}$

Deep neural network — formal representation

- We model data via L-hidden layer network; each layer l has p_l neurons/nodes
- The weight matrix and bias vector in each layer $l = 1, 2, \ldots, L$ are

$$W_i \in \mathbb{R}^{p_{l-1} \times p_l}, \quad \boldsymbol{b}_i \in \mathbb{R}^{p_l},$$

  which we denote by $\boldsymbol{\theta}$
- The network can be written as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = W_{L+1}\sigma_L(W_L\sigma_{L-1}(\quad \cdots \quad \sigma_1(W_1\boldsymbol{x})) + \boldsymbol{b}_L) + \boldsymbol{b}_{L+1}$$

  where $\sigma_1, \sigma_2, \ldots, \sigma_L$ are the activation functions

Sparse deep learning

- We approximate the familiar regression model

$$y_i = f_0(\boldsymbol{x}_i) + \epsilon_i, i = 1, 2, \ldots,$$

where $\boldsymbol{x}_i \in \mathbb{R}^p, \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$, with a sparse neural network $f_{\boldsymbol{\theta}}$[8]

[8] Bai et al. (2018). Efficient variational inference for sparse deep learning ...

Sparse deep learning

- We approximate the familiar regression model

$$y_i = f_0(\boldsymbol{x}_i) + \epsilon_i, i = 1, 2, \ldots,$$

where $\boldsymbol{x}_i \in \mathbb{R}^p, \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$, with a sparse neural network $f_{\boldsymbol{\theta}}$[8]

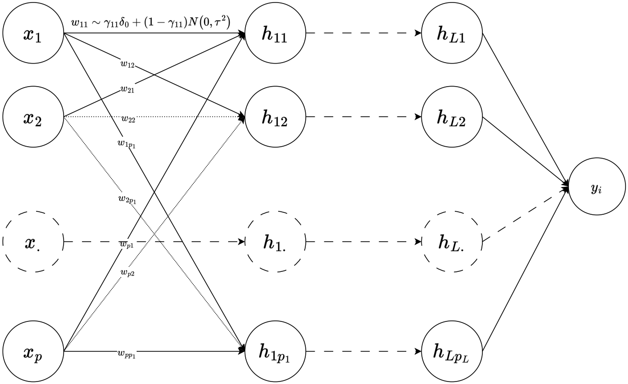
- We assume spike and slab prior for each θ—i.e., weight or bias.

$$\theta \sim (1-\gamma)\underbrace{\delta_0(\theta)}_{\text{spike}} + \gamma \underbrace{N(0, \tau^2)}_{\text{slab}}$$

$$\gamma \sim \text{Bernoulli}(\lambda)$$

[8] Bai et al. (2018). Efficient variational inference for sparse deep learning ...

Sparse deep learning



$$y_i = f_\theta(\vec{x}_i) + \epsilon_i; \quad \epsilon \sim N(0, \sigma^2)$$

Weights w are ON/OFF based on $\gamma \in \{0, 1\}$

Variational Bayes inference

- Inferences from the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{X}) \propto p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

is challenging—so people use MCMC, **variational methods**

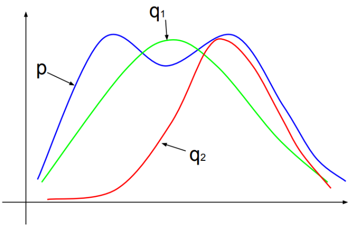
Variational Bayes inference

- Inferences from the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{X}) \propto p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

  is challenging—so people use MCMC, **variational methods**
- Given a variational family of distributions $\mathcal{Q}$, we find a member closest to the true posterior by

$$\arg\min_{q(\boldsymbol{\theta})\in\mathcal{Q}} \mathsf{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}|\boldsymbol{X}))$$



Xing, 10-708

Variational Bayes inference

- Inferences from the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{X}) \propto p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

  is challenging—so people use MCMC, **variational methods**
- Given a variational family of distributions $\mathcal{Q}$, we find a member closest to the true posterior by

$$\underset{q(\boldsymbol{\theta})\in\mathcal{Q}}{\arg\min}\, \mathsf{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}|\boldsymbol{X}))$$

- Equivalent to minimizing the negative ELBO:

$$\Omega = -E_{q(\boldsymbol{\theta})}[\log p(\boldsymbol{X}|\boldsymbol{\theta})] + \mathsf{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}))$$

Variational inference via SGD[9]

$$\Omega = \underbrace{-E_{q(\boldsymbol{\theta})}[\log p(\boldsymbol{X}|\boldsymbol{\theta})]}_{\text{reconstruction error}} + \underbrace{\mathsf{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}))}_{\text{regularizer}}$$

- Integrate the KL term analytically
- Compute the reconstruction error by Monte Carlo estimation

[9] Kigma & Welling (2014). Autoencoding variational Bayes.

Variational inference via SGD[9]

$$\Omega = \underbrace{-E_{q(\boldsymbol{\theta})}[\log p(\boldsymbol{X}|\boldsymbol{\theta})]}_{\text{reconstruction error}} + \underbrace{\mathsf{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}))}_{\text{regularizer}}$$

- Integrate the KL term analytically
- Compute the reconstruction error by Monte Carlo estimation
- Variational family distributions are reparametrized by some differential function $g(\omega, \nu)$ and random variable ν, for back-propagation

$$\tilde{\Omega}^m(\omega) = -\frac{n}{m}\frac{1}{K}\sum_{i=1}^{m}\sum_{k=1}^{K}\log p_{g(\omega,\nu)}(\boldsymbol{x}_i) + \mathsf{KL}(q_\omega(\boldsymbol{\theta})\|p(\boldsymbol{\theta}))$$

[9] Kigma & Welling (2014). Autoencoding variational Bayes.

Sparse deep learning

- The variational family $\mathcal{Q}$ follow spike and slab family. The ELBO Ω is approximated by

$$\tilde{\Omega} = \underbrace{-E_{q(\boldsymbol{\theta}|\gamma)q(\gamma)}[\log p(\boldsymbol{X}|\boldsymbol{\theta})]}_{\text{reconstruction error}} + \underbrace{\sum_{t=1}^{T}\left[\mathsf{KL}(q(\gamma_t)\|p(\gamma_t)) + q(\gamma_t = 1)\mathsf{KL}(N(a_i, b_i^2)\|N(0, \tau^2))\right]}_{\text{regularizer}}$$

[10] Maddison et al. (2017), Jang et al. (2017); Bai et al. (2020, SDL)

Sparse deep learning

- The variational family $\mathcal{Q}$ follow spike and slab family. The ELBO Ω is approximated by

$$\tilde{\Omega} = \underbrace{-E_{q(\boldsymbol{\theta}|\gamma)q(\gamma)}[\log p(\boldsymbol{X}|\boldsymbol{\theta})]}_{\text{reconstruction error}} + \underbrace{\sum_{t=1}^{T} \left[\mathsf{KL}(q(\gamma_t)\|p(\gamma_t)) + q(\gamma_t = 1)\mathsf{KL}(N(a_i, b_i^2)\|N(0, \tau^2))\right]}_{\text{regularizer}}$$

- Approximate the discrete variable γ sampling by[10]

$$\tilde{\gamma} \sim \text{Gumbel-softmax}(\phi, c),$$

c (temperature) controls the convergence to γ.

[10] Maddison et al. (2017), Jang et al. (2017); Bai et al. (2020, SDL)

Thank you!

clint@iitgoa.ac.in