Regularized Regression
A Bayesian point of view

Vincent MICHEL

Director : Gilles Celeux
Supervisor : Bertrand Thirion

Parietal Team, INRIA Saclay Ile-de-France
LRI, Université Paris Sud
CEA, DSV, I2BM, Neurospin

MM - 22 février 2010
Outline

1 Introduction to Regularization

2 Bayesian Regression
Example of Application in fMRI

**Fig.**: Example of application of regression (here SVR) in a fMRI paradigm. For clouds of different numbers of dots, we predict the quantities seen by the subject. In collaboration with E. Eger - INSERM U562.
Regression

Given a set of $m$ images of $n$ variables $\mathbf{X} = x_1, \ldots, x_n \in \mathbb{R}^{n,m}$ and a target $\mathbf{y} \in \mathbb{R}^m$, we have the following regression model:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$$

with $\mathbf{w}$ the parameters of the model ("weights"), and $\epsilon$ the error. The estimation $\hat{\mathbf{w}}$ of $\mathbf{w}$ is commonly done by Ordinary Least Squares:

$$\hat{\mathbf{w}} = \arg \min_{\hat{\mathbf{w}}} \| \mathbf{y} - \hat{\mathbf{w}}\mathbf{X} \|^2 \Rightarrow \hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$
Regularization

Motivations

OLS estimates often have **low bias but large variance**.

All features have a weight $\rightarrow$ Smaller subset with strong effects is more interpretable.

$\Rightarrow$ We need some **shrinkage** (or **regularization**) to constraint $\hat{w}$.

Constraint the values of $\hat{w}$ to have few parameters which explain well the data.

Bridge Regression - *I.E. Frank and J.H. Friedman - 1993*

$$\hat{w} = \arg \min_{\hat{w}} \| y - \hat{w}X \|^2 + \lambda_i \| \hat{w} \|_i^i, \text{ with } i \leq 0$$

Data fit Penalization
Ridge Regression


Ridge regression introduce a regularization with the $L^2$ norm:

$$\hat{w} = \arg \min_{\hat{w}} \| y - \hat{w}X \|^2 + \lambda \| \hat{w} \|^2$$

Properties

- Sacrifice a little of bias to reduce the variance of predicted values → More stable.
- **Grouping effect**: voxels having closed signal will have closed coefficients → Exhibits groups of correlated features.
- **Keeps all the regressors** in the model → not easily interpretable model.
Fig.: Coefficients of Ridge Regression for different values of $\lambda_2$. 
Lasso - "Least Absolute Shrinkage and Selection Operator"

$L^1$ norm - R. Tibshirani - 1996

Lasso regression introduce a regularization with the $L^1$ norm:

$$\hat{w} = \arg \min_{\hat{w}} \|y - \hat{w}X\|^2 + \lambda_1 \|\hat{w}\|_1$$

Properties

✓ Parsimony: only a small subset of features with $\hat{w}_i \neq 0$ are selected
  ⇒ increases the interpretability (especially conjugate with parcels).

✗ If $n > m$, i.e. more features than samples, Lasso will select at most $m$ variables.

✗ More difficult to implement than Ridge Regression.
Fig.: Coefficients of Lasso Regression for different values of $\frac{||\hat{w}||_1}{||\hat{w}_{OLS}||_2}$. 
Outline

1 Introduction to Regularization

2 Bayesian Regression
Questions

- What is a graphical model?
- Why using Bayesian probabilities?
- What is a prior?
- What is a Hierarchical Bayesian Model?
- Why using hyperparameters?

Bayesian decoding of brain images

Karl Friston, Carlton Chu, Janaina Mourão-Miranda, Oliver Hulme, Geraint Rees, Will Penny, and John Ashburner
Bayesian Regression and Graphical model

Model

The regression model: \( y = Xw + \epsilon \) can be seen as the following model:

\[
p(y|X, w, \lambda) = \mathcal{N}(Xw, \lambda) \text{ with } p(\epsilon) = \mathcal{N}(0, \lambda)
\]

Can we add some (prior) knowledge on the parameters?
Bayesian probabilities

Main idea - **Introducing prior information in the modelization**

Baye’s theorem: $p(w|y) = \frac{p(y|w)p(w)}{p(y)}$

where $p(w)$ is called a "prior". This "prior", introduced before any access to the data, reflects our prior knowledge on $w$.
Moreover, the **posteriori probability** $p(w|y)$ gives us the uncertainty in $w$ after we have observed $y$.

**Example**

Unbiased coin tossed three times → 3 heads.
A frequentist approach: $p(\text{head}) = 1$ → always head for all future tosses.
A bayesian approach: we can add a prior (Beta), in order to represent our prior knowledge of unbiased coin → gives less extrem results.

see Bishop, *Pattern Recognition and Machine Learning*, for more details.
Priors for Bayesian Regression - Ridge Regression

Prior for the weights $\mathbf{w}$

Penalization by weighted $L^2$ norm is equivalent to set a Gaussian prior on the weights $\mathbf{w}$:

$$\mathbf{w} \sim \mathcal{N}(0, \alpha I)$$

Bayesian Ridge Regression (C. Bishop 2000).

Prior for the noise $\epsilon$

Classical prior (M.E. Tipping, 2000).

$$\epsilon \sim \mathcal{N}(0, \lambda I_n)$$
Graphical model - Ridge Regression

\[ y \sim \mathcal{N}(0, \lambda I_n) \]

\[ w \sim \mathcal{N}(0, \alpha I) \]

\[ y = Xw + \epsilon \]
Example on Simulated Data - Ridge Regression

Log-density of the distribution of $w$ for the Bayesian Ridge Regression.

*Can we find some more adaptive priors?*
Prior for the weights $\mathbf{w}$

If we have different Gaussian priors for $\mathbf{w}$:

$$\mathbf{w} \sim \mathcal{N}(0, A^{-1}), \quad A = \text{diag}(\alpha_1, ..., \alpha_m)$$

with $\alpha_i \neq \alpha_j$ if $i \neq j$. Automatic Relevance Determination - ARD (MacKay 1994).

Prior for the noise $\epsilon$

Classical prior (M.E. Tipping, 2000).

$$\epsilon \sim \mathcal{N}(0, \lambda I_n)$$
Graphical model - ARD

\[ \epsilon \sim \mathcal{N}(0, \lambda I_n) \]

\[ w \sim \mathcal{N}(0, A^{-1}) \]

\[ A = \text{diag}(\alpha_1, \ldots, \alpha_m) \]
Example on Simulated Data - ARD

Log-densisity of the distribution of $w$ for ARD.
Other (less classical) Priors for Bayesian Regression

LASSO - Laplacian Prior for the weights $\mathbf{w}$

Penalization by weighted $L^1$ norm is equivalent to set Laplacian priors on the weights $\mathbf{w}$:

$$\mathbf{w} \sim C \exp^{-\lambda |\mathbf{w}|}$$

Elastic Net - Complex Prior for the weights $\mathbf{w}$

Penalization by weighted $L^1$ and $L^2$ norms is equivalent to set more complex prior on the weights $\mathbf{w}$:

$$\mathbf{w} \sim C(\lambda, \alpha) \exp^{-\lambda |\mathbf{w}|_1 + \alpha |\mathbf{w}|_2}$$

But...

✗ Difficult to estimate the parameters of the model...
Notions of Hyperpriors

What is an hyperprior?

- The parameters of a prior are called **hyperparameters**.
  E.g. for ARD:

  \[ w \sim \mathcal{N}(0, A^{-1}), \ A = \text{diag}(\alpha_1, ..., \alpha_m) \]

  \( \alpha_i \) is a hyperparameter.

- An hyperprior is a prior on the hyperparameters.

- Allow to adapt the parameters of the model to the data.

- Define a **Hierarchical Bayesian Model**.

Classical hyperpriors

See C. Bishop & M.E. Tipping, 2000
Bayesian Regression - Hierarchical Bayesian Model

\[ p(\lambda) = \Gamma(\lambda_1, \lambda_2) \]

\[ \epsilon \sim \mathcal{N}(0, \lambda I_n) \]

\[ y = Xw + \epsilon \]

\[ w \sim \mathcal{N}(0, A^{-1}) \]

\[ A = \text{diag}(\alpha_1, ..., \alpha_m) \]

\[ p(\alpha_i) = \Gamma(\gamma^i_1, \gamma^i_2) \]

\[ \gamma^i_1, \gamma^i_2 \]

But, how to "learn" the parameters?
THE difficult question: How to estimate our model?

We have a model, we have some data → How to ”learn” the parameters?

Likelihood

The **Log-Likelihood** is the quantity $p(y|X, w, \lambda)$ viewed as a function of the parameter $w$. It expresses **how probable the observed data is for different settings $w$**.

The main idea is to find the parameters which optimizes the **Log-Likelihood** $L(y|w)$, i.e. choosing the parameters for which the probability of the explained data $y$ is maximized.

Simple methods - When $L(y|w)$ is not too complicated

- **Maximum A Posteriori** - MAP
- **Estimation Maximization** - EM (Dempster, 1977). can be used in the presence of latent variables (e.g. mixture of Gaussians).
When $L(y|w)$ IS too complicated...

**Sampling methods**

When we cannot directly evaluate the distributions.
- Metropolis-Hatings, Gibb’s Sampling, simulated annealing.
- Computationally expensive.

2d example of Gibb’s sampling.
When $L(y|w)$ is too complicated...


Based on some (strongs) assumptions and approximations:

- **Laplace approximation**: Gaussian approximation of a probability density.

- **Factorized some distributions**.

Is equivalent to the maximization of the **Free Energy** which is a **lower bound** on the log-likelihood.
Limits of the Variational Bayes framework

Sometimes, the approximations of the VB framework are too strong for the model → the free energy is no longer useful...
Results on Real data - Visual task

**Fig.:** $w$ for: Elastic net (left) - Bayesian Regression (Right)
Conclusion

Bayesian Regression

- Allows to introduce prior information.
- Apriori distribution: uncertainty of the parameters after observing the data.
- Model estimation can be computationally expensive.
- Poor priors will lead to poor results with high confidence → always validate the results by cross-validation!

Thank you for your attention!
ARD - Practical point of view

**EM**

\[
\alpha_{i}^{\text{new}} = \frac{1 + 2\gamma_{i}^{1}}{\mu_{i}^{2} + \Sigma_{ii} + 2\gamma_{i}^{2}}
\]

\[
\lambda^{\text{new}} = \frac{n + 2\lambda_{1}}{(y - X\mu)^2 + \text{Trace}(\Sigma X^T X) + 2\lambda_{2}}
\]

with:

\[
w = \mu = \lambda \Sigma X^T y \quad \text{and} \quad \Sigma = (A + \lambda X^T X)^{-1}
\]

**Gibb’s Sampling**

\[
p(w|\theta - \{w\}) \sim \mathcal{N}(w|\mu, \Sigma)
\]

\[
p(\lambda|\theta - \{\lambda\}) \sim \Gamma(l_{1}, l_{2})
\]

\[
p(\alpha|\theta - \{\alpha\}) \sim \prod_{i=1}^{m} \Gamma(\alpha_{i}|g_{1}^{i}, g_{2}^{i})
\]

with \(g_{1}^{i} = \gamma_{1}^{i} + \frac{1}{2}, g_{2}^{i} = \gamma_{2}^{i} + \frac{1}{2} X_{i}^{2}, l_{1} = \lambda_{1} + n/2 \) and \(l_{2} = \lambda_{2} + \frac{1}{2}(y - Xw)^2\).
Estimation by Maximum Likelihood (ML)

Likelihood
The **Log-Likelihood** is the quantity \( p(y|X, w, \lambda) \) viewed as a function of the parameter \( w \). It expresses **how probable the observed data is for different settings** \( w \).

Estimation by ML
The main idea is to find the parameter \( \hat{w} \) which optimizes the **Log-Likelihood** \( L(y|w) \):

\[
\hat{w} = \arg \max_{\hat{w}} L(y|w)
\]

i.e. choosing \( \hat{w} \) for which the probability of the explained data \( y \) is maximized.

The solution is (hopefully !) the same as the one found by the **OLS**:

\[
\hat{w} = (X^T X)^{-1} X^T y
\]
How to construct the prediction $\hat{Y}$?

Iterative Algorithm

$$\hat{Y} = \sum_{i=1}^{i=n} x_i \hat{\beta}_i = X\hat{\beta}$$

Start with $\hat{Y} = 0$, and at each step:

- Compute the vector of current correlations $c(\hat{Y})$:
  $$\hat{c} = c(\hat{Y}) = X^T(Y - \hat{Y})$$

- Then, take the direction of the greatest current correlation for a small step $\epsilon$:
  $$\hat{Y} \Rightarrow \hat{Y} + \epsilon \cdot \text{sign}(\hat{c}_j) \cdot x_j$$

with $j = \arg \max_i |\hat{c}_i|$.
Choice of $\epsilon$

Forward Selection technique

- We have $\epsilon = |\hat{c}_j|$.
- If a feature seems interesting, we add it with $\beta_j = 1$.
- Could we weight the influence of $x_j$?

Forward Stagewise Linear Regression

- We have $\epsilon \ll 1$.
- If a feature seems interesting, we add it with $\beta_j \ll 1$.
- Another feature can be selected at the next step if it is more correlated with $\hat{Y}$.
- We adapt the weights to the features. But the computation can be time-consuming.
LARS - Least Angle Regression

Aim: accelerate the computation of the Stagewise procedure.

Least Angle Regression

- We compute the optimal $\epsilon$, i.e. the optimal step until another feature becomes more interesting.
- Take the equiangular direction between the selected features.
- Link with Lasso/Elastic net.