STA561: Probabilistic machine learning

Sparse Linear Models (10/7/13)

Lecturer: Barbara Engelhardt *Scribes: Jiaji Huang, Xin Jiang, Albert Oh*

1 Sparsity

- Sparsity has been a "hot topic" in statistics and machine learning since the LASSO paper was published in 1996. It is currently a very active area of research
- There are Bayesian and frequentist interpretations of sparsity
- Although the tools I present here are motivated by (and presented for) regression models, sparsity can be included in any model: factor analysis, mixture models, etc.

1.1 Sparse Linear Models

Consider the general problem of regression, including variables $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, $\beta \in \mathbb{R}^p$ (**sparse** vector), $\epsilon \in \mathbb{R}^n$. We write a linear model with p features and n samples as:

$$y = X\beta + \epsilon.$$

Recall that, in this linear regression model,

- y are the response terms; a noisy version of $X\beta$
- X are the observed predictors or covariates
- β coefficients (unobserved), weigh each of the p features in X depending how well feature j can be used to predict Y
- ϵ describes additive noise inherent in the model (here we will consider $\epsilon \in \Re^{n \times p}$ to be Gaussian noise, $\epsilon_{i,j} \sim \mathcal{N}\left(0, \sigma^2\right)$).

We can put a sparsity-inducing prior on the coefficients β that will allow us to choose a subset of the given features for our prediction model. Here, *sparsity* implies that there are some values of β that are zero, removing the corresponding X covariates from use in the prediction of the response Y. A sparse β means that we are selecting only a few features to describe a phenomenon (often called "feature selection" in Bayesian methods). There are a number of advantages to sparse β vectors:

1. **Prevents over-fitting.** Values of β close to zero often represent sample-specific noise that is being modeled, instead of true signal. If these irrelevant features are removed, we avoid fitting noise in the model. For example, let's say that a doctor wants to make a decision on whether or not a patient has heart disease. In blood tests, they see many features (HDL cholesterol, LDL cholesterol, blood pressure, cell count, etc.), but the doctor wants to make a decision based on a few, relevant factors. So, non-relevant features will be removed, leaving a sparse model of only relevant features (i.e., *biomarkers*),

and enabling this same set of relevant features to be used for diagnosis in other data sets, often with better generalization error (as we have not modeled noise).

2. **Fewer parameters to estimate.** If you have p features (predictors, covariates) and p is very large, it will be difficult to estimate the associated parameters from n samples. Sparsity reduces p so that the effective dimensionality is small. This is important because, if $p \gg n$, the problem is not solvable because there is not enough data from which we can estimate parameters (the problem is underconstrained). Sparsity effectively reduces p.

3. **Interpretability.** Having a sparse model makes interpreting the underlying phenomenon easier. For example, as a doctor, if you see that a few clinical covariates are predictive (i.e., going back to the heart disease example), you do not base your predictions (diagnoses) on the full data set, but a small set of underlying features. As a topic modeling example, generally documents are about a few topics, rather than many topics if the topics are well-defined and well-separated; in the former case, it is straightforward to make decisions about, for example, whether a document is relevant for a particular topic.

2 Bayesian Variable Selection

Bayesian variable selection is one way to induce sparsity into our regression model. In order to do this, we add indicator variables $\gamma_j, j = 1, \cdots, p$ whose value indicates whether a feature is "in" or "out" of our model. When a variable is "out", the corresponding coefficient has value 0; conversely, when a variable is "in", the corresponding coefficient will be non-zero. Specifically,

$$\gamma_j = \begin{cases} 1, & \text{feature } j \text{ is in} \\ 0, & \text{feature } j \text{ is out} \end{cases}.$$

The objective in this simple model is to compute $\hat{\gamma} = \arg\max_\gamma p(\gamma|\mathcal{D})$. By Bayes rule, the objective can be rewritten as $p(\gamma|\mathcal{D}) \propto p(\gamma)p(\mathcal{D}|\gamma)$. However, there are a few drawbacks with this approach:

1. This method requires us to search over all possible γ. However, since $\gamma \in \{0,1\}^p$, we will need to search throughout a set of size 2^p, which grows exponentially with p.

2. This is the maximum a posteriori (MAP) estimator, i.e., the mode of the posterior distribution. However, the mode itself does not necessarily represent the full posterior distribution well, e.g., in the case of well-correlated covariates. The included/excluded variables may be very unstable.

3. Another approach is to use the median model, which computes the median value of the posterior marginal inclusion probabilities instead of the mode of the full posterior. Then we have $\hat{\gamma} = \{j : p(\gamma_j = 1|\mathcal{D}) > t\}$, where t is an arbitrary threshold and can be set to any value in $[0, 1]$. The median estimator can be computed in a number of different ways, including by greedy algorithms that will be introduced in Section 3.

Despite the drawbacks mentioned earlier, MAP estimator is not without its merit because of its simplicity. In the following Section, we introduce the Spike and Slab model for the MAP estimator, which is a more structured sparsity-inducing prior.

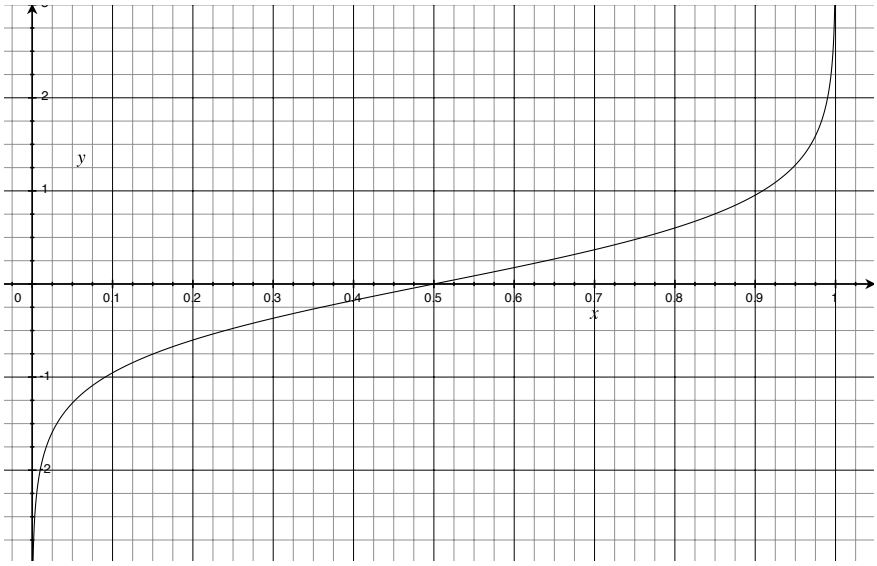


Figure 1: λ (y) vs. π_0 (x)

2.1 Spike and Slab model

By Bayes rule, the posterior can be rewritten in terms of the prior times the likelihood, $p(\gamma|\mathcal{D}) \propto p(\gamma)p(\mathcal{D}|\gamma)$. One way to induce sparsity in a specific model parameter is to use a "Spike and Slab" prior distribution, which is a mixture of a point mass at 0 (forcing $\beta_j = 0$, and excluding that covariate j) and a flat prior (Gaussian, often) on the included variables. In a mixture model, the distribution of the parameters β are dependent on (in this case) a Bernoulli random vector $\gamma \in \mathbb{R}^p$, which acts as the class assignment variable for each predictor j. When $\gamma_j = 0$, the corresponding feature coefficient $\beta_j = 0$; when $\gamma_j = 1$, the corresponding feature β_j is drawn from a Gaussian distribution with a large variance term. The interpretation of the two mixture components is clustering each predictor as noise (the spike at 0; excluded) and signal (the slab; included). Each γ_j is considered to be one of p i.i.d. draws from a Bernoulli with parameter π_0, i.e.,

$$\begin{aligned} p(\gamma|\pi_0) &= \prod_{j=1}^{p} \text{Bern}(\gamma_j|\pi_0) \\ &= \pi_0^{\|\gamma\|_0}(1-\pi_0)^{p-\|\gamma\|_0}, \end{aligned}$$

where $\|\cdot\|_0$ is the ℓ_0 norm, i.e., the number of non-zero elements in the vector. Rewriting the probability distribution and computing the log likelihood of γ, we have

$$\begin{aligned} \log p(\gamma|\pi_0) &= \|\gamma\|_0 \log \pi_0 + (p - \|\gamma\|_0)\log(1-\pi_0) \\ &= \|\gamma\|_0 (\log \pi_0 - \log(1-\pi_0)) + C \\ &= \lambda\|\gamma\|_0 + C, \end{aligned}$$

where $C = p\log(1-\pi_0)$ is a constant, and $\lambda = \log \frac{\pi_0}{1-\pi_0}$; this is the *logit function*. Here the quantity λ, which is a function of π_0, controls how sparse the features are. When π_0 is close to zero, $\lambda < 0$, and the feature vector is very sparse; when π_0 is close to one, $\lambda > 0$, and the feature vector is less sparse. See Figure 1 for the plot of λ (y-axis) versus π_0 (x-axis).

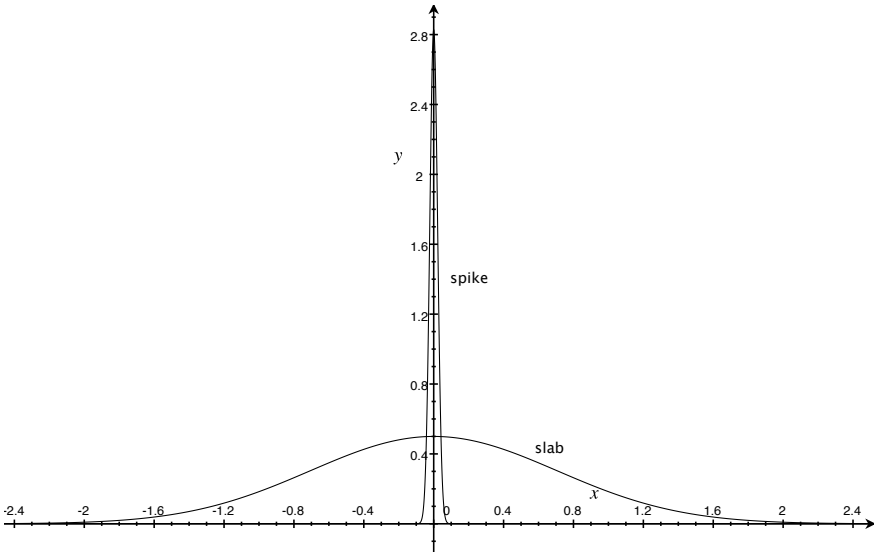


Figure 2: Cartoon of the pike and slab prior distribution for β_j. β_j is on the x-axis; $p(\beta_j|\pi_0, \sigma^2, \sigma_w^2)$ is on the y-axis.

Thus the likelihood function is

$$\begin{aligned} p(\mathcal{D}|\gamma) &= p(y|X, \beta, \gamma) \\ &= \iint p(y|X, \beta, \gamma)p(\beta|\gamma, \sigma^2)p(\sigma^2)d\beta d\sigma^2, \end{aligned}$$

where β_j has the Spike and Slab prior distribution:

$$\beta_j \sim \begin{cases} \delta_0(\beta_j), & \text{if } \gamma_j = 0 \\ \mathcal{N}(\beta_j|0, \sigma^2\sigma_w^2), & \text{if } \gamma_j = 1 \end{cases}.$$

The first term is a 'spike' at the origin. σ_w^2 controls the shrinkage on the coefficients associated with the included variables, which is scaled by the residual variance σ^2. As σ_w^2 approaches infinity, the distribution of β_j approaches a 'slab' or a uniform distribution over all values of β_j on the reals.

See Figure 2 for a cartoon illustration of the prior distribution of β. Thus, when $\gamma_j = 0$, β_j disappears, and the likelihood function becomes

$$p(D|\gamma) = \iint \mathcal{N}(y|X_\gamma\beta_\gamma, \sigma^2 I_n)\mathcal{N}(\beta_\gamma|0, \sigma^2\sigma_w^2 I_\gamma)p(\sigma^2)d\beta_\gamma d\sigma^2,$$

where the subscript γ indicates that we select the corresponding columns/elements/submatrix for $\gamma_j = 1$, so X_γ consists of the columns of X corresponding to $\gamma_j = 1$, β_γ consists of the elements of β corresponding to $\gamma_j = 1$, and $I_\gamma = I_{\|\gamma\|_0 \times \|\gamma\|_0}$.

Now simplify the likelihood function by assuming we have the knowledge of σ^2, and explicitly integrate out the coefficients β. We get

$$\begin{aligned} p(D|\gamma, \sigma^2) &= \int \mathcal{N}(y|X_\gamma\beta_\gamma, \sigma^2 I)\mathcal{N}(\beta_\gamma|0, \sigma^2\Sigma_\gamma)d\beta_\gamma \\ &= \mathcal{N}(y|0, \sigma^2 X_\gamma\Sigma_\gamma X_\gamma^T + \sigma^2 I_n). \end{aligned}$$

Note that the covariance term $\sigma^2 X_\gamma \Sigma_\gamma X_\gamma^T + \sigma^2 I_n$ depends on γ. The underlying question is: which coefficients are "in" and which are "out". Such information can all be found in the vector γ.

If we want to use an EM algorithm to solve this maximum likelihood estimation problem, we can do it by replacing the $\delta(\cdot)$ function with a normal distribution (with very small variance), but the method will suffer from severe local minima, especially with correlated covariates.

3 Greedy Search

Greedy methods are often helpful to solve optimization problems with sparsity and many local optima. Greedy methods are useful because they break down a multi-parameter problem into a series of single-parameter subproblems (i.e., the marginal problems), which are easier to solve because they do not have the associated combinatorial explosion of the joint problem.

3.1 Model Selection

What does it mean to be the 'best' model? Model scores are methods or statistics by which we can compare different models.

Intuitively, we would like to use the likelihood of generating our current data set $\mathcal{D}$ given a specific set of models that we are comparing. But what happens to the likelihood $p(\mathcal{D}|\Theta)$ when we add additional parameters to the model (i.e., add additional covariates to the prediction)? The likelihood will improve (assuming you are fitting the parameters in a reasonable way): you are checking the likelihood of your data that you are using to fit the parameters. Additional parameters will intuitively remove bias and add variance, heading toward overfitting your data set.

Let's introduce the Bayesian Information Criteria (BIC) as a way to analyze a generated model. Where the goal is to minimize:

$$\text{BIC} = -2 \log p(\mathcal{D}|\hat{\theta}_{ML}, M) + K \log(n),$$

where K = the number of parameters to estimate in the model (e.g., regression parameters and intercept), n = sample size, $\hat{\theta}_{ML}$ is the maximum likelihood estimate, and M refers to a conditioning on a certain model M. Smaller values of the BIC indicate superior models.

The first term of the BIC model is concerned with the ML estimate of the model selection. The second term is focused on the number of parameters (K) in the model and the sample size of the data (n). The second term of the BIC explicitly penalizes the complexity of the model scaled by the (log) sample size: additional samples increase (in a log relationship) the relative penalty of each additional parameter. Also note that when K is equal between two different models M_1 and M_2, the BIC is just the ML selection.

These greedy methods rely on a measure of comparison between different models. We may use model scores including Akaike information criterion (AIC) or Bayesian Information Criterion (BIC) or Minimum Description Length (MDL) to compare models with different choices of included covariates. The model parameters are the regression intercept and coefficients of the included covariates (but not the excluded covariates).

3.2 Types of Greedy Methods

3.2.1 Single Best Replacement (SBR)

Given a γ vector, consider all possible single bit changes starting with $\gamma^0 = [0]'$ initialization, and, at each iteration, include or exclude the covariate that improves the score maximally. Stop when no single bit change improves the score. For example:

- Compute score for all one bit changes i.e. compare score for: $\gamma = [110]'$ to score for γs $[010]'$, $[100]'$, and $[111]'$
- Choose the configuration such that the score is best; if the best configuration is worse than the current configuration of γ, stop.

3.2.2 Forward Selection

(Also known as: Forward Stepwise Search or Forward Stepwise Regression (FSR)).

Start with empty set, i.e., initialize with $\gamma^0 = [0]'$. Add single covariate for which the score improves the most at each step. Stop when there is no more improvement. This is identical to SBR except we do not consider removing covariates from the set already included.

3.2.3 Orthogonal Matching Pursuit (OMP)

Pick next covariate to include j^* in the current model by finding the one more correlated with residual.

$$j^* = \arg\min_{j \neq \gamma_t} \left[\min_w \|y - \underbrace{X\beta_t}_{\hat{y}_t} - \beta_j X_j\|^2 \right]$$

Set $\beta_j = \frac{X_j^T r_t}{\|X_j\|^2}$ as the maximum likelihood estimates of the covariate β_j (note that this is the normal equation), where $r_t = \|y - X\beta_t\|$ which is the current residual. Then,

$$j^* = \arg\max_j X_j^T r_t,$$

which is the inner product of the covariates with the current residual.

4 ℓ_1 Regularization

The ideal approach to introducing sparsity is to use the ℓ_0 norm (number of non-zero elements) for coefficients β: this will directly penalize the number of covariates that are included in the model. However, in practice, ℓ_1 norm is often used because it is a convex approximation of the ℓ_0 norm, and thus makes computation much easier. This amounts to introducing a Laplace prior (or a double exponential prior) on β:

$$p(\beta|\lambda) = \prod_{j=1}^{p} \text{Lap}\left(\beta_j|0, 1/\lambda\right) \propto \prod_{j=1}^{p} e^{-\lambda|\beta_j|} = e^{-\lambda\|\beta\|_1}, \tag{1}$$

where $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ is the ℓ_1 norm of β. In general, we can put a uniform prior on the intercept term, i.e., $p(\beta_0) \propto 1$. The Laplace prior is a heavy-tailed prior with significant mass at zero, which means that the elements near zero are shrunk to zero, but the heavy (i.e., sub-exponential) tails mean that large values of β are not highly unlikely (and thus not strongly shrunk toward zero). Then, the negative log posterior can be written as

$$\begin{aligned} -\log p(\beta|\mathcal{D}) &\propto -\log p(\mathcal{D}|\beta) - \log p(\beta|\lambda) \\ &= \sum_{i=1}^{n} \frac{1}{2\sigma^2} \underbrace{(y_i - \beta^\top x_i)^2}_{\text{RSS}(\beta)} + \lambda\|\beta\|_1 \end{aligned}$$

Thus, the MAP estimator $\hat{\beta}_{MAP}$ is obtained by solving the following optimization problem,

$$\min_{\beta} \text{RSS}(\beta) + \lambda\|\beta\|_1. \tag{2}$$

Or equivalently,

$$\begin{aligned} &\min_{\beta} \text{RSS}(\beta) \\ &\text{s.t.} \|\beta\|_1 \leq B, \end{aligned} \tag{3}$$

where B is a given upper bound of the ℓ_1 norm. In the above equations, λ dictates the sparsity weight; where setting a high λ creates a very sparse solution. In practice, the choice of λ depends on your dataset and problem, and we often determine an optimal λ through trying multiple values. This optimization problem is called Lasso.

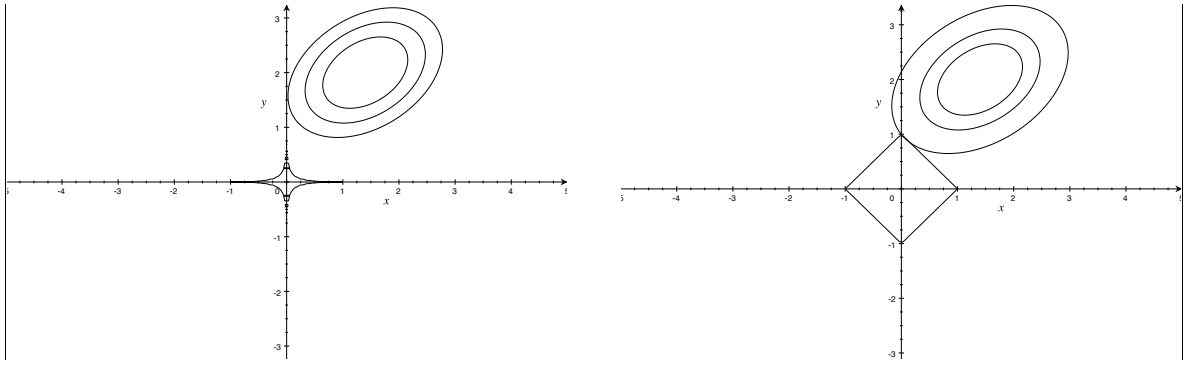
To see why ℓ_1 norm is sparsity-inducing, we show an illustration in Figure 3 on two covariates. The optimal solution is achieved when the lowest level set of RSS intersects the constraint surface (the ℓ_q norm ball). If we grow the constraint surface until it connects with the RSS, we can see that, for the ℓ_1 norm level set we will likely intersect the contour of the RSS at a corner of the diamond (level set), which occurs on a zero coordinate of one of β_1, β_2. If we grow a ℓ_2 norm ball instead, the shape of the level set does not encourage optimal solutions along the zero coordinate axes, so sparsity will not be preferred.

For this reason, as the $q \to 0$, the ℓ_q norm ball is more and more spikey (i.e., encourages solutions closer to the zero axes for more covariates). The solution is likely to occur on the axis and be sparse; however, the space is not convex, so optimization is difficult. In contrast, if we make the $q \to \infty$, the ℓ_q norm level set tends to be square, and the optimal solution tends to have same absolute value over all coordinates.

The exact solution to Lasso regression be obtained for any value of λ.

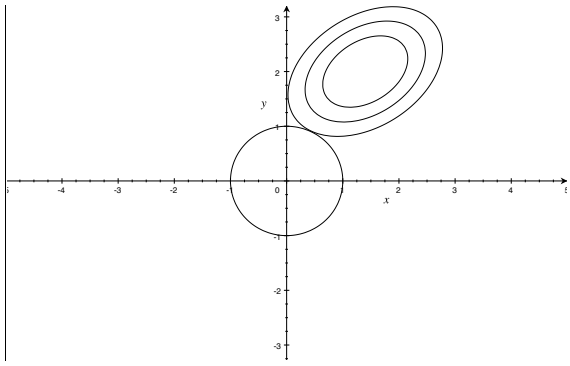
There are additional concepts for the interested student:

- Extensions of Lasso include group Lasso and fused Lasso
- Model averaging or stability selection
- Other Bayesian one-group priors that promote sparsity
- Extending sparsity to other models, e.g., factor analysis

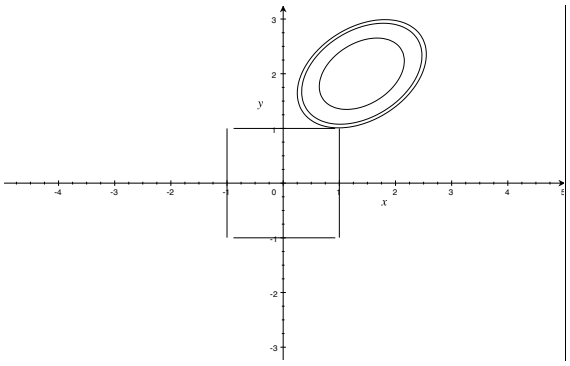


(a) $\ell_{1/3}$ ball

(b) ℓ_1 ball



(c) ℓ_2 ball



(d) ℓ_∞ ball

Figure 3: ℓ_q balls the RSS contours for two covariates (the axes are β_1, β_2).