Machine Learning and Data Mining

Meta-learning, lecture

Maxim Borisyak

2016/09/29

National Research University
Higher School of Economics

HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

Some statistics

Parameter estimation

Consider a distribution F from a parametrized distribution family $\mathcal{F}$ and samples $\{x\}_{i=1}^{N}$ i.i.d. from F. The task is to recover θ from $\{x\}_i$.

Maximum likelihood method

$$\begin{aligned} L(\theta) &= \prod_i p(x \mid \theta); \\ \hat{\theta} &= \arg\max L(\theta); \end{aligned}$$

Estimates of error

Consider decomposition of expected error:

$$\mathrm{Err} = \mathrm{err} + \omega$$

- Err - training (in-sample) error estimation;
- err - generalization error;
- ω - optimistic training bias.

C_p statistic

If d parameters are fit under MSE loss to data:

$$L = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

C_p statistic can be used to estimate expected error.

C_p statistic

$$C_p = \mathrm{err} + 2\frac{d}{N}\hat{\sigma_\varepsilon}^2$$

where: - $\hat{\sigma_\varepsilon}^2$ - estimation of intrinsic noise variance.

Akaike information criterion

If log-likelihood loss function is used:

$$\mathcal{L} = \sum_{i=1}^{N} \log_\theta(y_i)$$

Akaike information criterion for maximum-likelihood estimation θ can be used (holds for large N):

Akaike information criterion

$$\mathrm{AIC} = -2\mathbb{E}[\log P_\theta(Y)] \approx -2\frac{1}{N}\mathbb{E}[\mathcal{L}] + 2\frac{d}{N}$$

Consider a parametrized family of models $\{f_\alpha(x) \mid f \in F_\alpha\}$.
Then we can define:

Generalized AIC

$$\mathrm{AIC}(\alpha) = \mathrm{err}(\alpha) + 2\frac{d(\alpha)}{N}\hat{\sigma}_\varepsilon{}^2$$

Effective number of parameters

$$\mathrm{df}(\hat{y}) = \frac{1}{\hat{\sigma_\varepsilon}^2} \sum_{i=1}^{N} \mathrm{Cov}(\hat{y}_i, y_i)$$

For example, if a Neural Network minimizes $R(w) + \alpha\|w\|_2^2$:

$$\mathrm{df}(\alpha) = \sum_i \frac{\theta_i}{\theta_i + \alpha}$$

where θ_i - i-th eigenvalues of $\partial R / \partial w \partial w^T$

Bayesian approach

Consider model M with parameters θ and dataset D.

$$\begin{aligned} P(M \mid D) &\propto P(M)P(D \mid M) \\ &\propto P(M) \int P(D \mid \theta, M) P(\theta \mid M) d\theta \end{aligned}$$

$P(D \mid M)$ is called Bayesian factor and can be used for model selection.

Bayesian Information Criterion

For maximum-likelihood estimation $\hat{\theta}$:

$$\log P(D \mid M) = \log P(D \mid M, \hat{\theta}) - \frac{d_m}{2} \log N + O(1)$$

Regularization

Regularization

Usually, models represent a very wide family of functions to learn. Thus, minimization of training loss can lead to overfit.

Examples:

- Decision Trees represent a very wide range of functions and can easily lower most of commonly used losses to zero on training data;
- Neural Networks with enough units can interpolate any smooth function with arbitrary precision.

A common way to reduce overfit is to restrict effective model complexity by adding regularization term.

$$L_{\mathrm{reg}} = L + \text{complexity penalty}$$

Regularization

For linear models one of the common ways is to apply l_1 or l_2 regularizations:

$$
\begin{aligned}
L_{\mathrm{reg}}(w) &= L(w) + \lambda \|w\|_1; \\
L_{\mathrm{reg}}(w) &= L(w) + \lambda \|w\|_2^2;
\end{aligned}
$$

where:

$$
\begin{aligned}
\|w\|_1 &= \sum_i |w_i|; \\
\|w\|_2^2 &= \sum_i w_i^2;
\end{aligned}
$$

Ridge regression

Linear regression model with l_2 penalty is called *Ridge Regression*:

$$L(w) = \sum_i (y_i - wx)^2 + \lambda \|w\|_2^2$$

This can be rewritten as:

$$\begin{aligned} w^* = \quad & \arg\min \quad \sum_i (y_i - \mathbf{wx})^2; \\ & \text{subject to} \quad \|w\|_2^2 \leq t; \end{aligned}$$

or:

$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

Ridge regression

Using SVD decompisition:

$$X = UDV^T$$

$$\begin{aligned} Xw^* &= X^TX + \lambda I)^{-1}X^Ty \\ &= UD(D^2 + \lambda I)^{-1}DU^Ty \\ &= \sum_j u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T y; \end{aligned}$$

where:

- u_j - j-th left eigenvector (U matrix);
- d_j - j-th singular value (D_{jj});

Ridge regression

Effective degree of freedom:

$$\mathrm{df}(\lambda) = \sum_j \frac{d_j^2}{d_j^2 + \lambda}$$

Bayesian insight

Consider:

$$y \sim wx + \mathcal{N}(0, \sigma^2)$$

Assuming prior on $w_i \sim \mathcal{N}(0, \tau^2)$ and using Bayesian estimate:

$$w^* = \mathbb{E}\left[w \mid D\right]$$

is equivalent to:

$$L(w) = \sum_i (y_i - wx)^2 + \frac{\sigma^2}{\tau^2} \|w\|_2^2 \to \min$$

LASSO

LASSO regression is Linear regression with l_1 penalty:

$$L(w) = \sum_i (y_i - wx)^2 + \lambda |w|_1$$

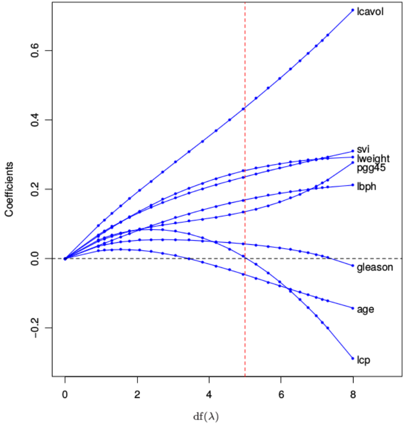
or:

$$\begin{aligned} w^* = \quad & \arg\min \quad \sum_i (y_i - \mathbf{wx})^2; \\ & \text{subject to} \quad \|w\|_1 \leq t; \end{aligned}$$

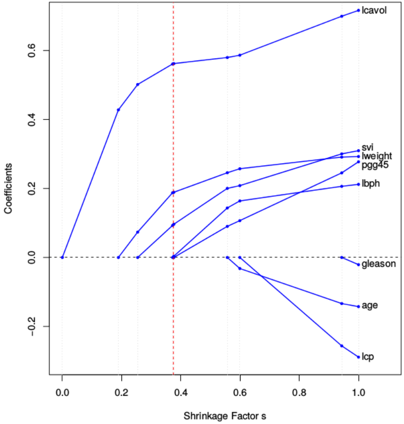
Discussion

Discussion

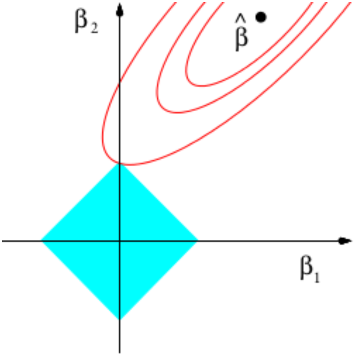
Ridge Regression

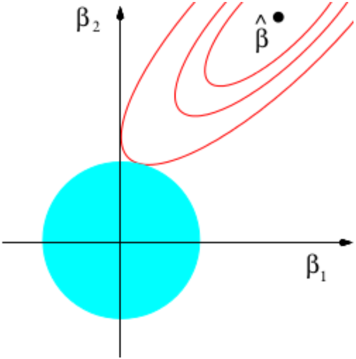


LASSO



l_1 regularization can be used for <u>feature selection</u>.





Tip

Note that regularized model with effective degrees of freedom df and unregularized one with the same number of parameters df, in general, is **not the same**.

Tip

Instead of reularization terms like $\|w\|_2^2$ you can use relative regularization:

$$\|w - w_0\|_2^2$$

Decision Trees

Regularization

Common ways is to restrict:

- maximal depth;
- minimal number of samples in a leaf;
- minimal number of samples to perform a split;
- minimal gain to perform split.

XGBoost regularization

$$\Omega(\mathrm{tree}) = \gamma T + \frac{\lambda}{2}\sum_{i=1}^{T} w_i;$$

where:

- T - number of leaves;
- w_i - i-th leaf score.

Learning with regularization

A Decision Tree is trained in a greedy fashion. Each new split is selected to maximize ***gain****.*

Regularized gain:

$$\mathrm{gain} = \frac{1}{2}\left\{\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} + \frac{(G_R+G_L)^2}{H_L+H_R+\lambda}\right\} - \gamma$$

where:

- G - gain;
- H - sum of loss gradients in the leaf;

This regularization will be discussed in much more details along with Gradient Boosting.