

# Machine Learning and Data Mining

## Meta-learning, lecture

---

Maxim Borisyak

2016/10/02

National Research University  
Higher School of Economics



HIGHER SCHOOL OF ECONOMICS  
NATIONAL RESEARCH UNIVERSITY

## SOME STATISTICS

---

*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  $\theta$  from  $\{x\}_i$ .*

## Maximum likelihood method

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

Consider decomposition of expected error:

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

- Err - training (in-sample) error estimation;
- err - generalization error;
- $\omega$  - optimistic training bias.

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

$$C_p = \text{err} + 2 \frac{d}{N} \hat{\sigma}_\epsilon^2$$

where: -  $\hat{\sigma}_\epsilon^2$  - estimation of intrinsic noise variance.

If log-likelihood loss function is used:

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

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

## Akaike information criterion

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

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

## Generalized AIC

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

## Effective number of parameters

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

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

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

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



Consider model  $M$  with parameters  $\theta$  and dataset  $D$ .

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

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

## Bayesian Information Criterion

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

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

# 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.*

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

# Regularization

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

$$L_{\text{reg}}(w) = L(w) + \lambda \|w\|_1;$$

$$L_{\text{reg}}(w) = L(w) + \lambda \|w\|_2^2;$$

where:

$$\|w\|_1 = \sum_i |w_i|;$$

$$\|w\|_2^2 = \sum_i w_i^2;$$

# 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:

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

or:

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

Using SVD decomposition:

$$X = UDV^T$$

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

where:

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

Effective degree of freedom:

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



Consider:

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

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

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

is equivalent to:

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

LASSO regression is Linear regression with  $l_1$  penalty:

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

or:

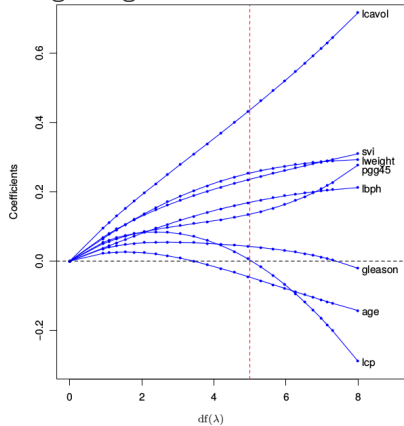
$$\begin{aligned} w^* = & \arg \min \sum_i (y_i - \mathbf{w}\mathbf{x})^2; \\ & \text{subject to } \|\mathbf{w}\|_1 \leq t; \end{aligned}$$

## DISCUSSION

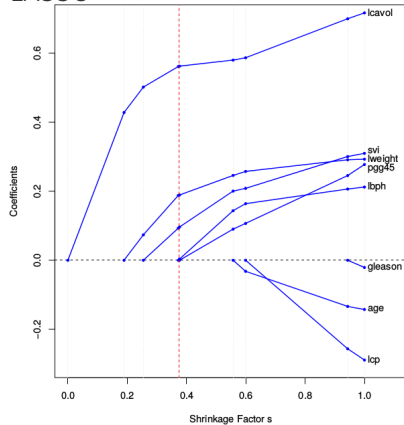
---

# Discussion

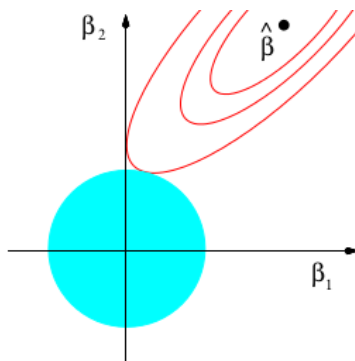
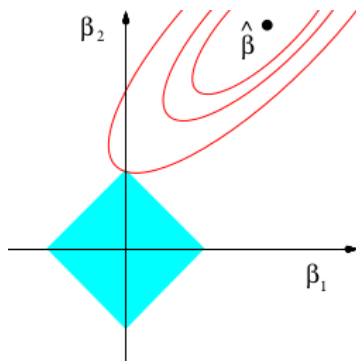
## Ridge Regression



## LASSO



$l_1$  regularization can be used for feature selection.



## 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 regularization terms like  $\|w\|_2^2$  you can use relative regularization:

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

# DECISION TREES

---

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.



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

where:

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

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

Regularized gain:

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

where:

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

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