Machine Learning and Data Mining

Bias-Variance decomposition and regularisation

Maxim Borisyak
National Research University Higher School of Economics (HSE)

September 12, 2017

Bias-Variance decomposition

Bias-Variance decomposition

Settings:

- random variable x and $t \sim x$;
- ground truth: $h(x) = \mathbb{E}[t \mid x]$;
- a regressor $y(x)$;
- MSE loss: $L = \sum_i (y(x_i) - t_i)^2$

Bias-Variance decomposition

Expected loss:

$$\mathcal{L} = \mathop{\mathbb{E}}_{x,t}[L] = \mathop{\mathbb{E}}_{x,t}[(y(x)-t)^2] =$$
$$\mathop{\mathbb{E}}_{x,t}(y(x)-h(x))^2 + \mathop{\mathbb{E}}_{x,t}(h(x)-t)^2 + 2\mathop{\mathbb{E}}_{x,t}(y(x)-h(x))(h(x)-t)$$

- $\mathbb{E}_{x,t}(h(x)-t)^2 = \sigma^2$ - irreducible error;
- $\mathbb{E}_{x,t}(y(x)-h(x))(h(x)-t) = 0$;
- $\mathbb{E}_{x,t}(y(x)-h(x))^2$ - of our main interest;

Bias-Variance decomposition

Our main interest is to derive behavior of $y(x, D)$ for a different training datasets D.
Let $\hat{y}(x) = \mathbb{E}_D\, y(x, D)$:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{x,D}(y(x, D) - h(x))^2 = \\
&\qquad\qquad \mathop{\mathbb{E}}_{x,D}\left[y(x, D) - \hat{y}(x) + \hat{y}(x) - h(x)\right]^2 = \\
&\qquad \mathop{\mathbb{E}}_{x,D}\left[y(x, D) - \hat{y}(x)\right]^2 + \mathop{\mathbb{E}}_{x,D}\left[\hat{y}(x) - h(x)\right]^2 + \\
&\qquad\qquad\qquad 2\mathop{\mathbb{E}}_{x,D}(y(x, D) - \hat{y}(x))(\hat{y}(x) - h(x))
\end{aligned}
$$

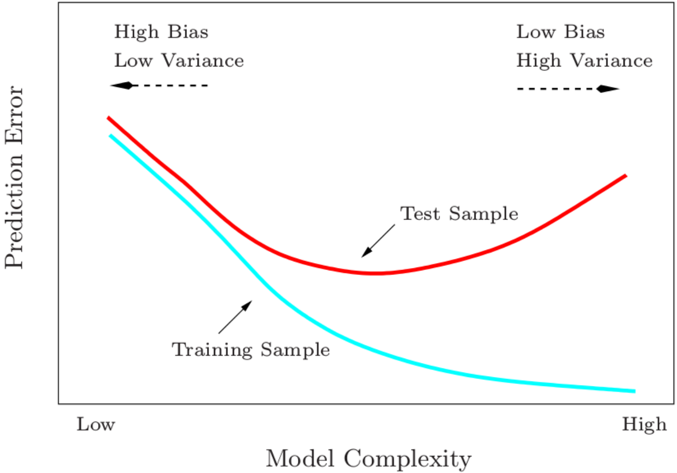
Bias-Variance decomposition

$$\underbrace{\mathop{\mathbb{E}}_{x,D}(y(x,D)-h(x))^2}_{\text{expected error}} =$$

$$\underbrace{\mathop{\mathbb{E}}_{x,D}\left[y(x,D)-\hat{y}(x)\right]^2}_{\text{variance}} + \underbrace{\mathop{\mathbb{E}}_{x,D}\left[\hat{y}(x)-h(x)\right]^2}_{\text{bias}^2} +$$

$$\underbrace{2\mathop{\mathbb{E}}_{x,D}(y(x,D)-\hat{y}(x))(\hat{y}(x)-h(x))}_{=0}$$

Bias-Variance



Bias-Variance

$$\text{high bias} \quad \Leftrightarrow \quad \text{undertrained}$$

$$\text{high variance} \quad \Leftrightarrow \quad \text{overtrained}$$

Regularization

Regularization: origins

Notation:

- X - data (features + labels);
- θ - parameters of algorithm;

Almost every machine learning algorithm ever:

$$\begin{aligned}
P(\theta \mid X) &\to \max; \\
P(\theta \mid X) &= \frac{1}{P(X)} P(X \mid \theta) P(\theta); \\
\mathcal{L} &= -\log P(\theta \mid X) = \\
&- \left[\underbrace{-\log P(X)}_{\text{const}} + \underbrace{\log P(X \mid \theta)}_{\text{likelihood}} + \underbrace{\log P(\theta)}_{\text{regularization}} \right]
\end{aligned}$$

Regularization

Regularization is essentially constraints on parameters:

$$\mathcal{L} = -\log P(X \mid \theta) - \log P(\theta) \to \min;$$

Using Lagrange multipliers:

$$\begin{array}{rcl} -\log P(X \mid \theta) & \to & \min; \\ \text{subject to:} & & \log P(\theta) \leq C \end{array}$$

Discussion

What is happening from Bayesian view when regularization term is omitted (i.e. maximum likelihood fits)?

Regularization: example

Let introduce Gaussian prior over parameters:

$$\theta \sim \mathcal{N}(0, \sigma \mathbb{I})$$

$$- \log P(\theta) =$$

$$- \log \left[\frac{1}{\sqrt{(2\pi)^k}} \cdot \exp \left(-\frac{1}{2\sigma} \|\theta\|_2^2 \right) \right] =$$

$$\text{const} + \frac{1}{2\sigma} \|\theta\|^2$$

Gaussian prior results in familiar l_2 regularization.

Example: logistic regression

Consider logistic regression:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \text{cross-entropy}(f_\theta(x_i), y_i) + \lambda\|w\|^2$$

where:

- $\theta = \{w, b\}$ - parameters;
- $f_\theta(x) = \sigma(wx + b)$ - decision function.

$$\|w\|^2 \leq \frac{1}{\lambda}\log 2$$

Example: l_1 vs l_2

l_1 regularization:

$$\mathcal{L} = -\log P(X \mid \theta) + \lambda|\theta|_1$$

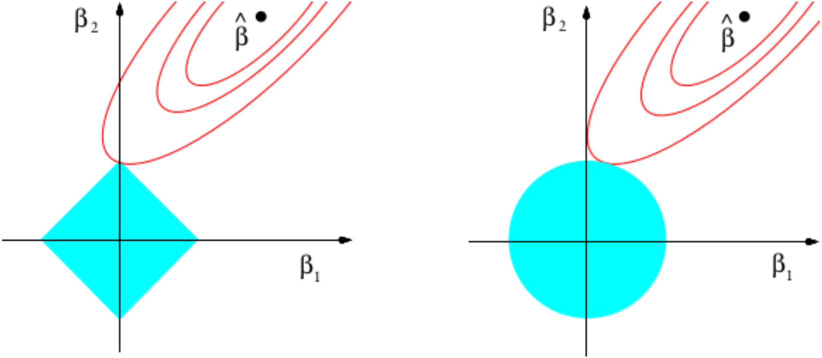
- tends to produce sparse vectors;
- can be used for feature selection;

l_2 regularization:

$$\mathcal{L} = -\log P(X \mid \theta) + \lambda|\theta|_2^2$$

- shrinks coefficients;
- never (almost surely) produces sparse vector.

Example: l_1 vs l_2



Example: ridge regression

Ridge-regression is a linear regression with l_2 regularization:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2 + \lambda\|w\|_2^2$$

Exact solution:

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

Compare to linear regression:

$$w^* = (X^T X)^{-1} X^T y$$

Eigen-values shrink:

$$d_j \to \sqrt{\frac{d_j^2}{d_j^2 + \lambda}}$$

Example: LASSO

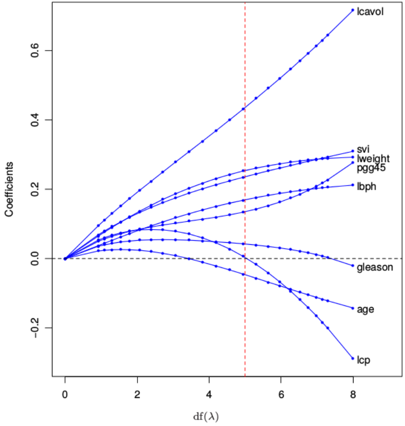
LASSO is a linear regression with l_1 regularization:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2 + \lambda|w|_1$$

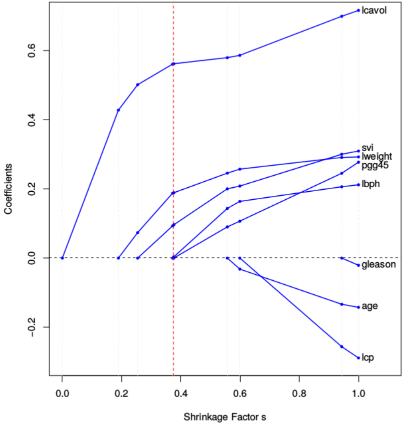
No closed-form solution.

Ridge vs. LASSO

Ridge regression:



LASSO:



Exotic regularizations

Almost every restriction on parameters can be imposed via regularization.

- prior on solution w^0 to a similar problem:

$$\|w - w^0\|_2^2$$

- adaptive regularization:

$$\sum_i c_i w_i^2$$

  where c_i is increasing with i;
- binding weights:

$$\sum_{i,j \in B} \|w_i - w_j\|_2^2$$

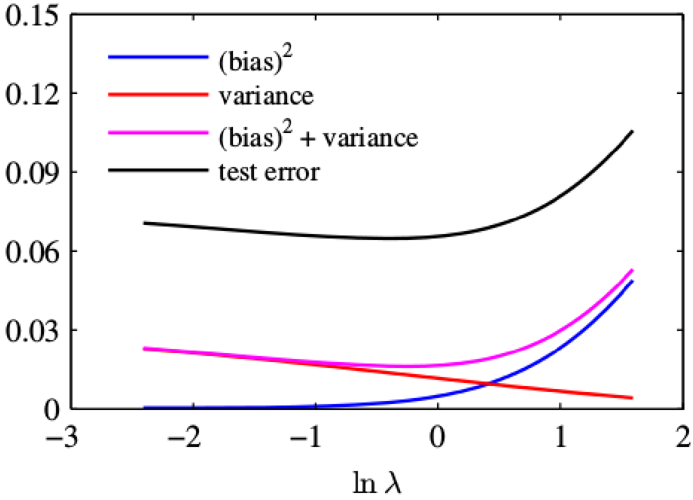
Regularization and bias-variance

Regularization and bias-variance

Regularization allows to control complexity of the model.

$$\text{stronger regularization} \Rightarrow \\ \text{lower model complexity} \Rightarrow \\ \text{lower variance and higher bias}$$

Regularization and bias-variance



Discussion

Is quantity

$$E = \mathrm{bias}^2 + \mathrm{variance}$$

preserved when regularization changes?

Does stronger regularization always imply higher bias?

Does stronger regularization imply lower variance?

Summary

Summary

- expected error can be decomposed into:

$$\text{expected error} = \mathrm{bias}^2 + \mathrm{variance} + \mathrm{irreducable\ noise}$$

- prior knowledge can be expressed via regularization;
- regularization usually controls bias-variance tradeoff.

References

- Bishop, C.M., 2006. Pattern recognition and machine learning. springer.
- Friedman, J., Hastie, T. and Tibshirani, R., 2001. The elements of statistical learning (Vol. 1, pp. 241-249). New York: Springer series in statistics.