Machine Learning

Regression. Classifier metrics

Aleksandr Petiushko

ML Research

August 12th, 2023

Content

1. Non-parametric Regression

Content

1. Non-parametric Regression
2. Bias-Variance trade-off for k-NN Regression

Content

1. Non-parametric Regression
2. Bias-Variance trade-off for k-NN Regression
3. Linear Regression

Content

1. Non-parametric Regression
2. Bias-Variance trade-off for k-NN Regression
3. Linear Regression
4. ML and MAP Principles

Content

Content

Content

Content

Content

1. Non-parametric Regression
2. Bias-Variance trade-off for k-NN Regression
3. Linear Regression
4. ML and MAP Principles
5. Least Squares Method
6. Ridge, LASSO and Elastic Net Regressions
7. Regression Metrics
8. Classification Metrics and Confusion Matrix
9. ROC and AUC

Content

1. Non-parametric Regression
2. Bias-Variance trade-off for k-NN Regression
3. Linear Regression
4. ML and MAP Principles
5. Least Squares Method
6. Ridge, LASSO and Elastic Net Regressions
7. Regression Metrics
8. Classification Metrics and Confusion Matrix
9. ROC and AUC
10. Precision and Recall

Content

Non-parametric Regression

- The main disadvantage of parametric models is that it is necessary to have a parametric model to describe the dependency

Non-parametric Regression

- The main disadvantage of parametric models is that it is necessary to have a parametric model to describe the dependency
- If it is impossible to select an adequate model, it makes sense to use non-parametric regression methods

Non-parametric Regression

- The main disadvantage of parametric models is that it is necessary to have a parametric model to describe the dependency
- If it is impossible to select an adequate model, it makes sense to use non-parametric regression methods

Assumption

Close objects correspond to close answers

Non-parametric Regression

The simplest model

We approximate the desired dependence by a constant in some neighborhood

Nadaraya-Watson kernel regression[1]

If there are several objects from the training sample in the vicinity of the point, then it is reasonable to use the weighted average as a prediction of the algorithm

$$a(x) = \frac{\sum_i y_i \omega_i(x)}{\sum_i \omega_i(x)},$$

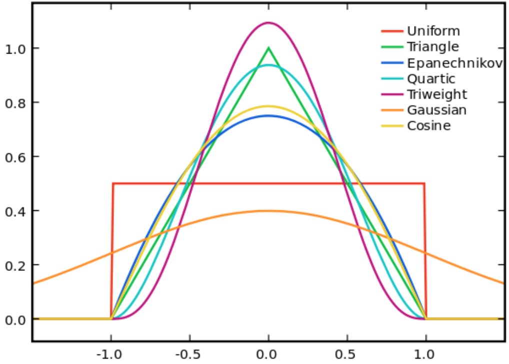
where $\omega_i(x) = K_h(x, x_i)$, a function K_h is called a **kernel** with smoothing window width h.

Note: we used for the homework $\omega_i(x) = \frac{1}{k}$ for the k-NN method.

[1] https://en.wikipedia.org/wiki/Kernel_regression

Examples of Kernels

- $K_h(x, x_i) = K(\frac{||x - x_i||}{h})$
- Typical Examples: [2]



[2] https://en.wikipedia.org/wiki/Kernel_(statistics)

Reminder: bias-variance tradeoff

Definitions

Let $y = y(x) = f(x) + \varepsilon$ be the target dependence, where $f(x)$ is the deterministic function, $\varepsilon \sim N(0, \sigma^2)$ and $a(x)$ is the machine learning algorithm.

$$E(y - a)^2 = \sigma^2 + variance(a) + bias^2(f, a)$$

Bias and Variance of k-NN Regression

Bias

$$bias^2(f,a) = (E(f(x_0) - a(x_0)))^2 = \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_{(i)})\right)^2$$

Bias and Variance of k-NN Regression

Bias

$$bias^2(f,a) = (E(f(x_0) - a(x_0)))^2 = \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_{(i)})\right)^2$$

Variance

$$Variance(a) = D\left(\frac{1}{k}\sum_{i=1}^{k} y(x_{(i)})\right) = \frac{1}{k^2} D\left(\sum_{i=1}^{k} y(x_{(i)})\right) =$$

$$= \frac{1}{k^2} D\left(\sum_{i=1}^{k} (f(x_{(i)}) + \varepsilon_i)\right) = \frac{1}{k^2} D\left(\sum_{i=1}^{k} f(x_{(i)})\right) + \frac{1}{k^2} D\left(\sum_{i=1}^{k} \varepsilon_i\right) =$$

$$= 0 + \frac{1}{k^2} k\sigma^2 = \frac{\sigma^2}{k}$$

Bias-Variance tradeoff of k-NN Regression

$$Error(x_0) = E(a(x_0) - f(x_0))^2 = \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_{(i)})\right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

Bias-Variance tradeoff of k-NN Regression

$$Error(x_0) = E(a(x_0) - f(x_0))^2 = \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_{(i)})\right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

- Higher k, lower variance

Bias-Variance tradeoff of k-NN Regression

$$Error(x_0) = E(a(x_0) - f(x_0))^2 = \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_{(i)})\right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

- Higher k, lower variance
- Higher k, higher bias

Bias-Variance tradeoff of k-NN Regression

$$Error(x_0) = E(a(x_0) - f(x_0))^2 = \left(f(x_0) - \frac{1}{k} \sum_{i=1}^{k} f(x_{(i)}) \right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

- Higher k, lower variance
- Higher k, higher bias

Note: Under "reasonable assumptions" the bias of the 1-NN estimator vanishes entirely as the size of the training set approaches infinity

Conclusion

- The main advantage of non-parametric regression is the absence of assumptions about the form of the dependence model

Conclusion

- The main advantage of non-parametric regression is the absence of assumptions about the form of the dependence model
- The method has a large number of variations to customize

Conclusion

- The main advantage of non-parametric regression is the absence of assumptions about the form of the dependence model
- The method has a large number of variations to customize
  - Metric learning (e.g., l_*-metric variations)

Conclusion

- The main advantage of non-parametric regression is the absence of assumptions about the form of the dependence model
- The method has a large number of variations to customize
  - Metric learning (e.g., l_*-metric variations)
  - Number of nearest neighbors k

Conclusion

- The main advantage of non-parametric regression is the absence of assumptions about the form of the dependence model
- The method has a large number of variations to customize
  - Metric learning (e.g., l_*-metric variations)
  - Number of nearest neighbors k
  - Weights in the weighted version of the method

Conclusion

- The main advantage of non-parametric regression is the absence of assumptions about the form of the dependence model
- The method has a large number of variations to customize
  - Metric learning (e.g., l_*-metric variations)
  - Number of nearest neighbors k
  - Weights in the weighted version of the method
  - Smoothing window width

Time for questions

Problem Statement: Linear Regression

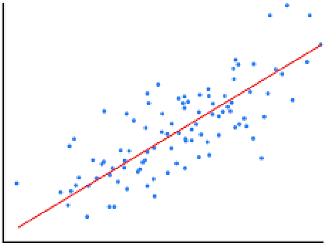
Given

$$p(y_i|x_i) \sim w^T x_i + \varepsilon_i \sim N(w^T x_i, \sigma^2),$$

for $i = 1..,\ell$, where $w \in \mathbf{R}^{n+1}$, $\varepsilon_i \sim N(0, \sigma^2)$

Task

Find w



Two kinds of parameter estimation

Maximum Likelihood (**ML**) Principle

$$w_{ML} = \arg\max_{w} \; p(y|w, x)$$

Two kinds of parameter estimation

Maximum Likelihood (ML) Principle

$$w_{ML} = \arg\max_{w} \; p(y|w, x)$$

Principle of Maximum A Posterior (MAP) Probability

$$w_{MAP} = \arg\max_{w} \; p(w|x, y)$$

Maximum likelihood estimator

$$w_{ML} = \arg\max_{w} \sim p(y|w, x)$$

Maximum likelihood estimator

$$w_{ML} = \arg\max_{w} \sim p(y|w, x)$$

$$w_{ML} = \arg\max_{w} \sim \prod_{i} p(y_i|w, x_i)$$

Maximum likelihood estimator

$$w_{ML} = \arg\max_{w} \sim p(y|w, x)$$

$$w_{ML} = \arg\max_{w} \sim \prod_{i} p(y_i|w, x_i)$$

$$p(y_i|w, x_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right)$$

Maximum likelihood estimator

$$w_{ML} = \arg\max_w \sim p(y|w,x)$$

$$w_{ML} = \arg\max_w \sim \prod_i p(y_i|w,x_i)$$

$$p(y_i|w,x_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right)$$

$$w_{ML} = \arg\max_w \sim \prod_i \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right) = \arg\max_w \sum_i -\frac{(y_i - w^T x_i)^2}{2\sigma^2}$$

Maximum likelihood estimator

$$w_{ML} = \arg\max_w \sim p(y|w, x)$$

$$w_{ML} = \arg\max_w \sim \prod_i p(y_i|w, x_i)$$

$$p(y_i|w, x_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right)$$

$$w_{ML} = \arg\max_w \sim \prod_i \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right) = \arg\max_w \sum_i -\frac{(y_i - w^T x_i)^2}{2\sigma^2}$$

$$w_{ML} = \arg\min_w \sim \sum_i (y_i - w^T x_i)^2$$

Least squares method

Problem statement and assumptions

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$

Least squares method

Problem statement and assumptions

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$
- $a(x) = f_w(x) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$, where $w = (w_0, w_1, \ldots, w_n)^T \in \mathbb{R}^{n+1}$ — model parameters.

Least squares method

Problem statement and assumptions

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$
- $a(x) = f_w(x) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n$, where $w = (w_0, w_1, ..., w_n)^T \in \mathbb{R}^{n+1}$ — model parameters.
- It is convenient to write in vector form

$$a(x) = w^T \cdot x,$$

where $x = (1, x^1, ..., x^n)^T \in \mathbb{R}^{n+1}$.

Least squares method

Problem statement and assumptions

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$
- $a(x) = f_w(x) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n$, where $w = (w_0, w_1, ..., w_n)^T \in \mathbb{R}^{n+1}$ — model parameters.
- It is convenient to write in vector form

$$a(x) = w^T \cdot x,$$

where $x = (1, x^1, ..., x^n)^T \in \mathbb{R}^{n+1}$.

Least squares method

- $L(w, X_{train}) = MSE(w, X_{train}) = \frac{1}{\ell} \sum_i (w^T \cdot x^{(i)} - y_i)^2$ — loss function

Least squares method

Problem statement and assumptions

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$
- $a(x) = f_w(x) = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n$, where $w = (w_0, w_1, ..., w_n)^T \in \mathbb{R}^{n+1}$ — model parameters.
- It is convenient to write in vector form

$$a(x) = w^T \cdot x,$$

where $x = (1, x^1, ..., x^n)^T \in \mathbb{R}^{n+1}$.

Least squares method

- $L(w, X_{train}) = MSE(w, X_{train}) = \frac{1}{\ell}\sum\limits_i (w^T \cdot x^{(i)} - y_i)^2$ — loss function
- The task is to find $\hat{w} = \arg\min\limits_w (L(w, X_{train}))$

Analytical solution

Theorem

The solution to the problem $\arg\min_w(\sum_{i=1}^{\ell}(w^T \cdot x_i - y_i)^2)$ is $\hat{w} = (X^T X)^{-1} \cdot X^T \cdot y$, where $X_{i,j} = x_i^j$, $y = (y_1, ..., y_\ell)$.

Analytical solution

Theorem

The solution to the problem $\arg\min_w(\sum_{i=1}^{\ell}(w^T \cdot x_i - y_i)^2)$ is $\hat{w} = (X^T X)^{-1} \cdot X^T \cdot y$, where $X_{i,j} = x_i^j$, $y = (y_1, ..., y_\ell)$.

Proof idea

Let's write the problem in vector form $||Xw - y||^2 \to \min_w$. The necessary condition for a minimum in matrix form is:

$$\frac{\partial}{\partial w}||Xw - y||^2 = 0$$

Polynomial Regression

Idea

It is possible to generate new features based on existing ones by applying non-linear functions

Polynomial Regression

Idea

It is possible to generate new features based on existing ones by applying non-linear functions

Transformation examples

- Exponentiation
- Pairwise products
- Square root

Pros and cons of linear regression

Pros and cons of linear regression

Advantages

- Simple algorithm, not computationally complex
- Linear regression is a well interpretable model
- Despite its simplicity, it can describe quite complex dependencies (for example, polynomials)

Pros and cons of linear regression

Advantages

- Simple algorithm, not computationally complex
- Linear regression is a well interpretable model
- Despite its simplicity, it can describe quite complex dependencies (for example, polynomials)

Disadvantages

- The algorithm assumes that all features are numeric
- The algorithm assumes that the data is normally distributed, which is not always the case
- The algorithm is highly sensitive to outliers

Time for questions

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\max_{w} \sim p(w|x_1, ...x_\ell, y_1, ..., y_\ell)$$

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\max_{w} \sim p(w|x_1, ...x_\ell, y_1, ..., y_\ell)$$

$$w_{MAP} = \arg\max_{w} \sim \prod_{i} p(y_i|x_i, w)p(w)$$

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\max_{w} \sim p(w|x_1, ...x_\ell, y_1, ..., y_\ell)$$

$$w_{MAP} = \arg\max_{w} \sim \prod_i p(y_i|x_i, w)p(w)$$

$$w_{MAP} = \arg\max_{w} \sim \sum_i \ln p(y_i|x_i, w) + \ln p(w)$$

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\max_w \sim p(w|x_1, ...x_\ell, y_1, ..., y_\ell)$$

$$w_{MAP} = \arg\max_w \sim \prod_i p(y_i|x_i, w)p(w)$$

$$w_{MAP} = \arg\max_w \sim \sum_i \ln p(y_i|x_i, w) + \ln p(w)$$

$$w_{MAP} = \arg\max_w \sim \sum_i -\frac{(y_i - w^T x_i)^2}{2\sigma^2} + \ell \ln p(w)$$

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\max_w \sim p(w|x_1, ...x_\ell, y_1, ..., y_\ell)$$

$$w_{MAP} = \arg\max_w \sim \prod_i p(y_i|x_i, w)p(w)$$

$$w_{MAP} = \arg\max_w \sim \sum_i \ln p(y_i|x_i, w) + \ln p(w)$$

$$w_{MAP} = \arg\max_w \sim \sum_i -\frac{(y_i - w^T x_i)^2}{2\sigma^2} + \ell \ln p(w)$$

$$w_{MAP} = \arg\min_w \sim \sum_i \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \ell \ln p(w)$$

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\max_{w} \sim p(w|x_1, ...x_\ell, y_1, ..., y_\ell)$$

$$w_{MAP} = \arg\max_{w} \sim \prod_{i} p(y_i|x_i, w)p(w)$$

$$w_{MAP} = \arg\max_{w} \sim \sum_{i} \ln p(y_i|x_i, w) + \ln p(w)$$

$$w_{MAP} = \arg\max_{w} \sim \sum_{i} -\frac{(y_i - w^T x_i)^2}{2\sigma^2} + \ell \ln p(w)$$

$$w_{MAP} = \arg\min_{w} \sim \sum_{i} \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \ell \ln p(w)$$

An additional term appeared in the minimization problem, which depends only on the prior distribution on the weights w

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\min_{w} \sim \sum_{i} \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \ell \ln p(w)$$

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\min_w \sim \sum_i \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \ell \ln p(w)$$

Let's assume that $p(w) \sim N(0, \tau^2)$

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\min_{w} \sim \sum_{i} \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \ell \ln p(w)$$

Let's assume that $p(w) \sim N(0, \tau^2)$

$$w_{MAP} = \arg\min_{w} \sim \sum_{i} \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \frac{\ell w^T w}{2\tau^2}$$

Maximum A Posterior Probability Method

$$w_{MAP} = \arg\min_w \sim \sum_i \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \ell \ln p(w)$$

Let's assume that $p(w) \sim N(0, \tau^2)$

$$w_{MAP} = \arg\min_w \sim \sum_i \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \frac{\ell w^T w}{2\tau^2}$$

$$w_{MAP} = \arg\min_w \sim \frac{1}{\ell} \sum_i \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \frac{1}{2\tau^2} ||w||^2$$

Ridge Regression

Ridge Regression

L_2 regularization

- $L(w, X_{train}) = MSE(w, X_{train}) + \frac{\alpha}{2}\sum_{i=0}^{n} w_i^2 = \frac{1}{\ell}\sum_{i}(w^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum_{i=0}^{n} w_i^2$ — loss function

Ridge Regression

L_2 regularization

- $L(w, X_{train}) = MSE(w, X_{train}) + \frac{\alpha}{2}\sum\limits_{i=0}^{n} w_i^2 = \frac{1}{\ell}\sum\limits_{i}(w^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum\limits_{i=0}^{n} w_i^2$ — loss function
- The task is to find $\hat{w} = \arg\min\limits_{w}(L(w, X_{train}))$

Ridge Regression

L_2 regularization

- $L(w, X_{train}) = MSE(w, X_{train}) + \frac{\alpha}{2}\sum\limits_{i=0}^{n} w_i^2 = \frac{1}{\ell}\sum\limits_{i}(w^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum\limits_{i=0}^{n} w_i^2$ — loss function
- The task is to find $\hat{w} = \arg\min\limits_{w}(L(w, X_{train}))$

Note: L_2 regularization and MAP with the normally distributed weights are the same!

Ridge regression: solution

Theorem

The solution of the problem $\arg\min\limits_{w}(\sum\limits_{i=1}^{\ell}(w^T \cdot x^{(i)} - y_i)^2 + \alpha \sum\limits_{i=0}^{n} w_i^2)$ is
$\hat{w} = (X^T X + \alpha I_{n+1})^{-1} \cdot X^T \cdot y$, where $X_{i,j} = x_i^j$, $y = (y_1, ..., y_\ell)$, I_{n+1} is the identity matrix.

Proof idea

Let's write the problem in vector form $||Xw - y||^2 + \alpha||w||^2 \to \min\limits_{w}$. The necessary condition for a minimum in matrix form is:

$$\frac{\partial}{\partial w}\left((Xw - y)^T \cdot (Xw - y) + \alpha w^T w\right) = 0$$

Ridge Regression: Properties

- Regularization prevents model parameters from being too large
- In general, regularization provides better generalization ability
- More resistant to outliers
- A parameter has been added that can be configured using cross-validation

Ridge Regression: Properties

- Regularization prevents model parameters from being too large
- In general, regularization provides better generalization ability
- More resistant to outliers
- A parameter has been added that can be configured using cross-validation

The probabilistic meaning of the α parameter

$\alpha = \frac{1}{\tau^2}$, where τ is the standard deviation of the prior distribution on w

LASSO

L_1-regularization

- $L(w, X_{train}) = MSE(w, X_{train}) + \alpha \sum_{i=0}^{n} |w_i| = \sum_i (w^T \cdot x^{(i)} - y_i)^2 + \alpha \sum_{i=0}^{n} |w_i|$ — loss function

LASSO

L_1-regularization

- $L(w, X_{train}) = MSE(w, X_{train}) + \alpha \sum\limits_{i=0}^{n} |w_i| = \sum\limits_{i} (w^T \cdot x^{(i)} - y_i)^2 + \alpha \sum\limits_{i=0}^{n} |w_i|$ — loss function
- The task is to find $\hat{w} = \arg\min\limits_{w}(L(w, X_{train}))$

LASSO

L_1-regularization

- $L(w, X_{train}) = MSE(w, X_{train}) + \alpha \sum_{i=0}^{n} |w_i| = \sum_i (w^T \cdot x^{(i)} - y_i)^2 + \alpha \sum_{i=0}^{n} |w_i|$ — loss function
- The task is to find $\hat{w} = \arg\min_w (L(w, X_{train}))$

Properties

- This regularization provides feature selection
- No analytical solution

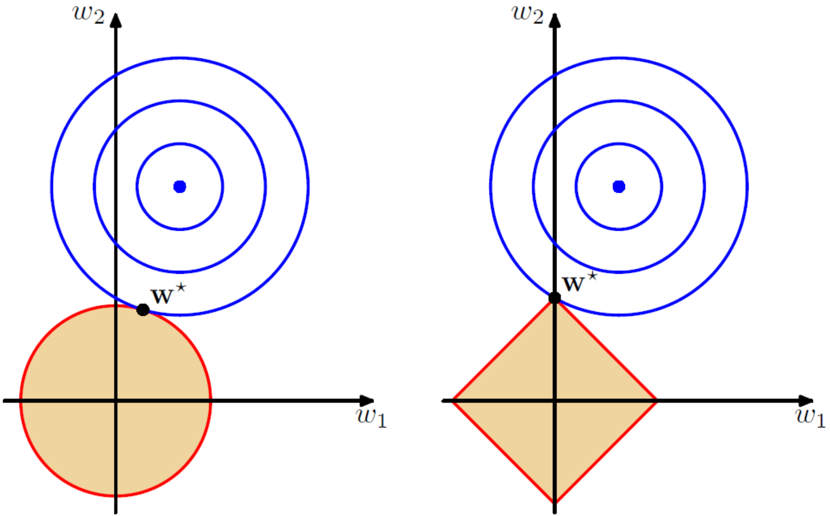
Probabilistic interpretation of LASSO

The probabilistic meaning of the α parameter

The parameter α — is inversely proportional to the standard deviation of the prior distribution by w. In this case, this is the *Laplace* distribution

$$p(w) = \frac{1}{\tau} exp\left(-\frac{||w||}{2\tau}\right)$$

Intuition of feature selection under L_1-regularization



Elastic Net

L_1-regularization and L_2-regularization

- $L(w, X_{train}) = MSE(w, X_{train}) + r\alpha \sum_{i=0}^{n} |w_i| + (1-r)\frac{\alpha}{2} \sum_{i=0}^{n} w_i^2 = \sum_i (w^T \cdot x^{(i)} - y_i)^2 + r\alpha \sum_{i=0}^{n} |w_i| + (1-r)\frac{\alpha}{2} \sum_{i=0}^{n} w_i^2$ — loss function

Elastic Net

L_1-regularization and L_2-regularization

- $L(w, X_{train}) = MSE(w, X_{train}) + r\alpha \sum_{i=0}^{n} |w_i| + (1-r)\frac{\alpha}{2} \sum_{i=0}^{n} w_i^2 = \sum_i (w^T \cdot x^{(i)} - y_i)^2 + r\alpha \sum_{i=0}^{n} |w_i| + (1-r)\frac{\alpha}{2} \sum_{i=0}^{n} w_i^2$ — loss function
- The task is to find $\hat{w} = \arg\min_w (L(w, X_{train}))$

Properties

- No analytical solution
- Combines the positive properties of Ridge regression and LASSO.

Time for questions

Quality Metrics for the Regression Problem

Motivation

- The formulation of a machine learning problem usually begins with the definition of a metric and fixing a test dataset on which this metric will be calculated

Quality Metrics for the Regression Problem

Motivation

- The formulation of a machine learning problem usually begins with the definition of a metric and fixing a test dataset on which this metric will be calculated
- An incorrectly chosen metric can make it difficult to use the machine learning model in real life and nullify the efforts of the team developing the machine learning algorithm

Quality Metrics for the Regression Problem

Motivation

- The formulation of a machine learning problem usually begins with the definition of a metric and fixing a test dataset on which this metric will be calculated
- An incorrectly chosen metric can make it difficult to use the machine learning model in real life and nullify the efforts of the team developing the machine learning algorithm
- As a rule, the customer does not think in terms of metrics and can only explain the problem he wants to solve in business language

Quality Metrics for the Regression Problem

Motivation

- The formulation of a machine learning problem usually begins with the definition of a metric and fixing a test dataset on which this metric will be calculated
- An incorrectly chosen metric can make it difficult to use the machine learning model in real life and nullify the efforts of the team developing the machine learning algorithm
- As a rule, the customer does not think in terms of metrics and can only explain the problem he wants to solve in business language
- Understanding the impact of the choice of a particular metric on the customer's business is the key to successful problem setting

Quality Metrics for the Regression Problem

Mean Square Error

$$MSE = \frac{1}{\ell} \sum_i (y_i - a(x_i))^2$$

Quality Metrics for the Regression Problem

Mean Square Error

$$MSE = \frac{1}{\ell}\sum_i (y_i - a(x_i))^2$$

Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{\ell}\sum_i (y_i - a(x_i))^2}$$

Quality Metrics for the Regression Problem

Mean Square Error

$$MSE = \frac{1}{\ell}\sum_i (y_i - a(x_i))^2$$

Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{\ell}\sum_i (y_i - a(x_i))^2}$$

Mean Absolute Error

$$MAE = \frac{1}{\ell}\sum_i |y_i - a(x_i)|$$

Quality Metrics for the Regression Problem

Max Error

$$ME = max(|y_i - a(x_i)|)$$

Quality Metrics for the Regression Problem

Max Error

$$ME = max(|y_i - a(x_i)|)$$

Mean Squared Logarithmic Error

$$MSLE = \frac{1}{\ell}\sum_{i}(\ln y_i - \ln a(x_i))^2$$

Quality Metrics for the Regression Problem

Max Error

$$ME = max(|y_i - a(x_i)|)$$

Mean Squared Logarithmic Error

$$MSLE = \frac{1}{\ell}\sum_i(\ln y_i - \ln a(x_i))^2$$

R^2 score (also known as Coefficient of Determination)

$$R^2 = 1 - \frac{\sum_i(y_i - a(x_i))^2}{\sum_i(y_i - \bar{y})^2},$$

where $\bar{y} = \frac{1}{\ell}\sum_i y_i$.

Conclusion

- Linear regression — simple, well-interpreted model, but not robust to outliers
- Has a clear probabilistic interpretation
- Regularization is a great way to deal with the overfitting and data noise

Time for questions

Classification of binary classifier responses

- Training set $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- Classification problem into 2 classes: $X \to Y, Y = \{+1, -1\}$
- Classification algorithm $a(x) : X \to Y$
- The class labeled "$+1$" is called "**positive**"
- The class labeled "-1" is called "**negative**"

Classification of binary classifier responses

- Training set $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- Classification problem into 2 classes: $X \to Y, Y = \{+1, -1\}$
- Classification algorithm $a(x) : X \to Y$
- The class labeled "+1" is called "**positive**"
- The class labeled "-1" is called "**negative**"

Таблица: Classification of responses

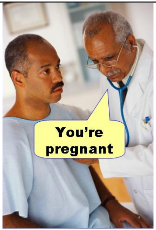
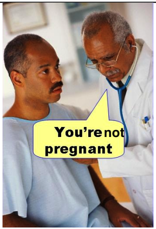
	Algorithm output	Correct answer
TP (True Positive)	$a(x_i) = +1$	$y_i = +1$
TN (True Negative)	$a(x_i) = -1$	$y_i = -1$
FP (False Positive)	$a(x_i) = +1$	$y_i = -1$
FN (False Negative)	$a(x_i) = -1$	$y_i = +1$

Confusion Matrix

More clearly, these relationships can be depicted using **confusion matrix** (matrix of errors)

		Correct answer	
		$y = +1$	$y = -1$
Algorithm Output	$a(x) = +1$	True Positive	False Positive (Type 1 Error)
	$a(x) = -1$	False Negative (Type 2 Error)	True Negative

Confusion Matrix

	$y = +1$	$y = -1$
$a(x) = +1$		
$a(x) = -1$		

The simplest quality metric

- The simplest quality metric is the proportion of correct answers on a test (control sample)
- Common name: **Accuracy**

Accuracy formula

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m}[a(x_i) = y_i] = \frac{TP+TN}{TP+FP+TN+FN}$$

The simplest quality metric

- The simplest quality metric is the proportion of correct answers on a test (control sample)
- Common name: **Accuracy**

Accuracy formula

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m}[a(x_i) = y_i] = \frac{TP+TN}{TP+FP+TN+FN}$$

Disadvantages

- Ignores class imbalance
- The cost of an error on objects of different classes is not taken into account

Metrics based on the positive response of the algorithm

Consider the metrics that are based on the calculation of the proportion of positive responses of the algorithm.

Proportion of *incorrect* positive classifications

Also known as False Positive Rate, or **FPR**.

$FPR(a, X^m) = \frac{\sum_{i=1}^{m}[y_i{=}-1][a(x_i){=}+1]}{\sum_{i=1}^{m}[y_i{=}-1]}$

Metrics based on the positive response of the algorithm

Consider the metrics that are based on the calculation of the proportion of positive responses of the algorithm.

Proportion of *incorrect* positive classifications

Also known as False Positive Rate, or **FPR**.

$$FPR(a, X^m) = \frac{\sum_{i=1}^{m}[y_i{=}-1][a(x_i){=}+1]}{\sum_{i=1}^{m}[y_i{=}-1]}$$

Proportion of *correct* positive classifications

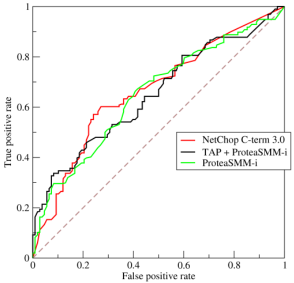
Also known as True Positive Rate, or **TPR**.

$$TPR(a, X^m) = \frac{\sum_{i=1}^{m}[y_i{=}+1][a(x_i){=}+1]}{\sum_{i=1}^{m}[y_i{=}+1]}$$

Note. Notice the different denominators!

Error Curve

Best known as Receiver Operating Characteristic (**ROC-curve**), in which we look at the trade-off between false alarm rate and correct response rate.



FPR is plotted along the X-axis, TPR is plotted along the Y-axis[3].
Note. On this curve, miss rate (FN) s not taken into account in any way.

[3] `https://wikipedia.org`

Area under the ROC curve and types of ROC curves

AUROC

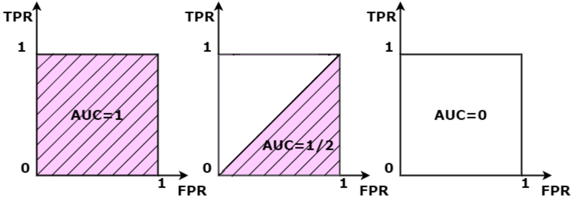
The greater the value of the correct TPR prediction for each FPR error value, the better the classifier performs.
Thus, the area under the curve (Area Under Curve, AUC / AUROC) must be maximized.

Area under the ROC curve and types of ROC curves

AUROC

The greater the value of the correct TPR prediction for each FPR error value, the better the classifier performs.
Thus, the area under the curve (Area Under Curve, AUC / AUROC) must be maximized.

ROC-curves for the best (AUC=1), random (AUC=0.5) and worst (AUC=0) algorithm:



The Task: build ROC, find AUROC

Suppose that the binary classification algorithm $a(x_i)$ on the sample X^m decides to assign a class based on some scalar value $g_\theta(x_i) \in \mathbb{R}$, where θ is the set of model parameters and $g_\theta(x_i)$ is the discriminant function:

- Let's treat Positive response by a (varying) threshold t: $g_\theta(x_i) \geq t$

Task

- We want to build an ROC curve, i.e. find points $\{(FPR_i, TPR_i)\}_{i=1}^{m}$
- Calculate area under curve - AUROC

The Task: build ROC, find AUROC

Suppose that the binary classification algorithm $a(x_i)$ on the sample X^m decides to assign a class based on some scalar value $g_\theta(x_i) \in \mathbb{R}$, where θ is the set of model parameters and $g_\theta(x_i)$ is the discriminant function:

- Let's treat Positive response by a (varying) threshold t: $g_\theta(x_i) \geq t$

Task

- We want to build an ROC curve, i.e. find points $\{(FPR_i, TPR_i)\}_{i=1}^m$
- Calculate area under curve - AUROC

Let's count the number of correct answers of different types:

- $m_+ = \sum_{i=1}^m [y(x_i) = +1]$ (TPR denominator)
- $m_- = \sum_{i=1}^m [y(x_i) = -1]$ (FPR denominator); $m = m_+ + m_-$

Let us order the training set X^m in descending order of the values $g_\theta(x_i)$.
Then the formula for $AUROC = \frac{1}{m_-} \sum_{i=1}^m [y_i = -1] TPR_i$ (see below).

AP

Task solution

Algorithm

We put the first point at the origin: $(FPR_0, TPR_0) = (0, 0), AUROC = 0$.

Task solution

Algorithm

We put the first point at the origin: $(FPR_0, TPR_0) = (0,0)$, $AUROC = 0$.

Loop over ordered selection $i = 1 \dots m$

Threshold — the next value of the discriminant function $t = g_\theta(x_i)$

If $y_i = -1$:

- $(FPR_i, TPR_i) = (FPR_{i-1} + \frac{1}{m_-}, TPR_{i-1})$ (move along the X-axis)
- $AUROC = AUROC + \frac{1}{m_-} TPR_i$

Task solution

Algorithm

We put the first point at the origin: $(FPR_0, TPR_0) = (0,0)$, $AUROC = 0$.

Loop over ordered selection $i = 1 \dots m$

Threshold — the next value of the discriminant function $t = g_\theta(x_i)$

If $y_i = -1$:

- $(FPR_i, TPR_i) = (FPR_{i-1} + \frac{1}{m_-}, TPR_{i-1})$ (move along the X-axis)
- $AUROC = AUROC + \frac{1}{m_-} TPR_i$

If $y_i = +1$:

- $(FPR_i, TPR_i) = (FPR_{i-1}, TPR_{i-1} + \frac{1}{m_+})$ (move along the Y-axis)

Other Important Metrics 1

In information retrieval problems

- **Precision**: $Precision = \frac{TP}{TP+FP}$ (percentage of relevant objects among those found)
- **Recall**: $Recall = \frac{TP}{TP+FN}$ (percentage of found objects among relevant ones)

Other Important Metrics 1

In information retrieval problems

- **Precision**: $Precision = \frac{TP}{TP+FP}$ (percentage of relevant objects among those found)
- **Recall**: $Recall = \frac{TP}{TP+FN}$ (percentage of found objects among relevant ones)

How to apply

- **Precision**: allows you to ensure that there are few false alarms; but it does not say anything about misses (the cost of a false alarm is high, and the price of a miss is low).
- **Recall**: allows you to ensure that there are few misses; but it does not say anything about false alarms (the price of a miss is high, and the price of a false alarm is low).

Remark. Often the task is to optimize one metric while fixing another.

Other Important Metrics 2

In problems of medical diagnostics

- **Sensitivity**: $Sensitivity = \frac{TP}{TP+FN}$ (percentage of correct positive diagnoses)
- **Specificity**: $Specificity = \frac{TN}{TN+FP}$ (percentage of correct negative diagnoses)

Other Important Metrics 2

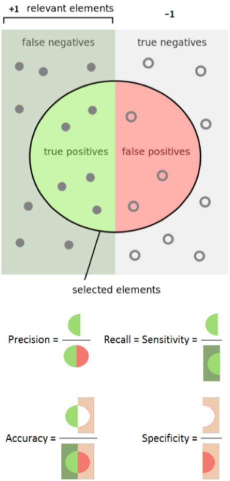
In problems of medical diagnostics

- **Sensitivity**: $Sensitivity = \frac{TP}{TP+FN}$ (percentage of correct positive diagnoses)
- **Specificity**: $Specificity = \frac{TN}{TN+FP}$ (percentage of correct negative diagnoses)

How to apply

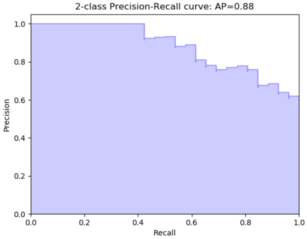
- **Sensitivity**: Maximize the number of true positive diagnoses, but ignore false diagnoses (treatment cost is low and skip cost is high).
- **Specificity**: Maximize the number of correct negative diagnoses, but don't take into account missed diagnoses (treatment cost is high and skip cost is low).

Metrics illustration



Aggregated Metrics over Precision-Recall

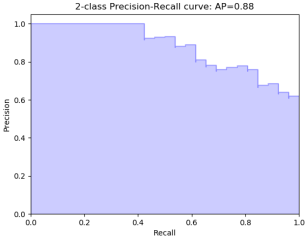
You can build a Precision-Recall (PR-curve) similar to the ROC-curve:



Remark. Note that in this case the curve is not necessarily monotonic!

Aggregated Metrics over Precision-Recall

You can build a Precision-Recall (PR-curve) similar to the ROC-curve:



Remark. Note that in this case the curve is not necessarily monotonic!

AUPRC

- Similar to AUROC, you can calculate the area under the PR curve - AUPRC
- Another name is Average Precision (with some assumptions on the integration method): the more, the better

Multi-class classification

For each class $c \in Y$, denote by TP_c, FP_c, and FN_c true positives, false positives, and false negatives. Then:

Precision and recall with macro-averaging

- $Precision = \frac{\sum_c TP_c}{\sum_c (TP_c + FP_c)}$
- $Recall = \frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)}$
- Insensitive to errors on small classes

Multi-class classification

For each class $c \in Y$, denote by TP_c, FP_c, and FN_c true positives, false positives, and false negatives. Then:

Precision and recall with macro-averaging

- $Precision = \frac{\sum_c TP_c}{\sum_c (TP_c + FP_c)}$
- $Recall = \frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)}$
- Insensitive to errors on small classes

Precision and recall with micro-averaging

- $Precision = \frac{1}{|Y|} \sum_c \frac{TP_c}{TP_c + FP_c}$
- $Recall = \frac{1}{|Y|} \sum_c \frac{TP_c}{TP_c + FN_c}$
- Sensitive to errors on small classes

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small
- Sensitivity and specificity are suitable for problems with unbalanced classes (as in medicine, for example)

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small
- Sensitivity and specificity are suitable for problems with unbalanced classes (as in medicine, for example)
- AUROC is suitable for quality assessment with a non-fixed error (miss rate) cost ratio

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small
- Sensitivity and specificity are suitable for problems with unbalanced classes (as in medicine, for example)
- AUROC is suitable for quality assessment with a non-fixed error (miss rate) cost ratio
- Another aggregated quality score - F-measure:
  $F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$
  - This is the *harmonic mean* that goes to zero when at least one of the values goes to zero

Time for questions

Thank you!