Tutorial
Introduction to Machine Learning
SS 2019

Week 3, March 4-8th, 2019 (Regression, Classification)

Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `https://las.inf.ethz.ch/teaching/introml-s19`
Email questions to:
laurie.prelot@inf.ethz.ch,

Disclaimer: The pdf version of the tutorial has been adapted to take into account student's questions encountered during the week, and therefore can differ slightly from the in-class version.

1 Definitions

Cost and Loss

Usage 1: In some books, cost and loss functions are used in an interchangeable manner.
Usage 2: However, when a distinction is made, the loss is defined on a data point $L\left(f\left(\mathbf{x}_i\right), \mathbf{y}_i\right)$ and the cost is defined over the points of the training set $\frac{1}{n}\sum_{i=1}^{n}\left(L\left(f\left(\mathbf{x}_i\right), \mathbf{y}_i\right)\right)$.
The cost can include the regularization term as well.

Objective function

The objective is a general term used for any function that is optimized during training.

What happens during the learning phase?
The model is built. In other words, the model is "fit" and the parameters are estimated. An optimization algorithm is needed to find the hypothesis $\hat{h}$ which minimizes the true risk (expected error).

$$L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}, \quad f = \underset{h \in \mathcal{F}}{\operatorname{argmin}} \underbrace{\mathbb{E}\left(L\left(f\left(\mathbf{x}\right), \mathbf{y}\right)\right)}_{R(h)}$$

However, the true risk cannot be computed in practice. An estimate of the true risk on a sample data set is given by the empirical risk. For a training set $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_n, y_n)\}$

$$L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}, \qquad f = \underset{h \in \mathcal{F}}{\operatorname{argmin}} \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(L\left(f\left(\mathbf{x}_i\right), \mathbf{y}_i\right)\right)}_{R_n(h)}$$

The law of large numbers gives:

$$\forall h \in \mathcal{F}, \qquad R_n\left(h\right) \xrightarrow[n\to\infty]{} R\left(h\right)$$

Performance, Evaluation Metric

After the model parameters are learnt, the performance is evaluated. Test samples are fed to the model, and the number of mistakes is recorded. Useful models should be capable of extracting all relevant information from the training data, but also predict well on unseen samples. This is called generalization.

Remark on cross validation

The simple cross validation scheme is conducted for a better performance assessment. The training and testing steps are repeated k times. The error is averaged over k sets. If k is large enough, our estimator should tend to be unbiased.

2 Loss functions examples

2.1 Regression

$$L\text{quadratic} : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \qquad \mathbf{y}, f(\mathbf{x}) \to (\mathbf{y} - f(\mathbf{x}))^2$$

Penalizes highly outliers (”very large” mistakes)

$$L\text{absolute} : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \qquad \mathbf{y}, f(\mathbf{x}) \to |\mathbf{y} - f(\mathbf{x})|$$

It is more robust to outliers

$$L\epsilon\text{-insensitive} : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \qquad \mathbf{y}, f(\mathbf{x}) \to max\left(0, |\mathbf{y} - f(\mathbf{x})| - \epsilon\right)$$

Tolerance for small errors

$$L\text{Huber} : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \qquad \mathbf{y}, f(\mathbf{x}) \to \begin{cases} \frac{1}{2}(\mathbf{y} - f(\mathbf{x}))^2 & \text{if} \quad |\mathbf{y} - f(\mathbf{x})| < \delta \\ \delta\,|\mathbf{y} - f(\mathbf{x})| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

Good trade off between
- quadratic loss, when $\mathbf{y} - f(\mathbf{x})$ is small
- absolute loss, when $\mathbf{y} - f(\mathbf{x})$ is large

2.2 Classification

$$L0/1 : \{-1, 1\} \times \mathbb{R} \to \mathbb{R} \qquad \mathbf{y}, f(\mathbf{x}) \to \begin{cases} 1 & \text{if } sign\left(f(\mathbf{x})\right) \neq \mathbf{y} \\ 0 & \text{otherwise} \end{cases}$$

Direct minimization of the empirical error rate is difficult due to the non-convexity of the 0-1 loss function. Instead, many classification procedures attempt to minimize convex surrogate loss functions.

Fisher consistency

Intuition:
Given a surrogate loss function $\psi : \mathcal{Y} \times \mathcal{S} \to \mathbb{R}$, the surrogate is said to be consistent with respect to the loss $L : \mathcal{Y} \times \mathcal{S} \to \mathbb{R}$, if every minimizer f of the surrogate risk function $R_\psi(f)$, is also a minimizer of the risk function $R_L(f)$
Example: The hinge and the logistic losses are consistent with respect to the 0-1 loss

Classification Losses

$L\text{perceptron} : \{-1, 1\} \times \mathbb{R} \to \mathbb{R}$ $\quad \mathbf{y}, f(\mathbf{x}) \to max\left(0, -\mathbf{y} f(\mathbf{x})\right)$
Find the best separation hyperplane

$L\text{hinge} : \{-1, 1\} \times \mathbb{R} \to \mathbb{R}$ $\quad \mathbf{y}, f(\mathbf{x}) \to max\left(0, 1 - \mathbf{y} f(\mathbf{x})\right)$
Find large separation margin

$L\text{logistic} : \{-1, 1\} \times \mathbb{R} \to \mathbb{R}$ $\quad \mathbf{y}, f(\mathbf{x}) \to log\left(1 + exp\left(-\mathbf{y} f(\mathbf{x})\right)\right)$
Link to cross entropy and probabilistic
interpretation, (cf. lecture logistic regression)

3 Performance criteria

3.1 Regression

The goal is to measure the distance between the predicted and the target values

Mean squared error

$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - f(\mathbf{x}_i)\right)^2$, Penalizes highly outliers ("very large" mistakes)

Root Mean squared error (RMSE)

$\sqrt{\text{MSE}}$, Errors are in the same unit as the target

Mean absolute error

$\frac{1}{n}\sum_{i=1}^{n}\left|y_i - f(\mathbf{x}_i)\right|$, Robustness to outliers

Root Mean squared log error (RMSLE)

$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[log\left(f(\mathbf{x}_i) + 1\right) - log\left(\mathbf{y}_i + 1\right)\right]^2}$, Useful to deal with large values

Coefficient of determination

Square of correlation coefficient: $R = \frac{\sum_{i=1}^{n}\left(\mathbf{y}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{y}_j\right)\left(f(\mathbf{x}_i) - \frac{1}{n}\sum_{j=1}^{n}f(\mathbf{x}_j)\right)}{\sqrt{\sum_{i=1}^{n}\left(\mathbf{y}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{y}_j\right)^2}\sqrt{\sum_{i=1}^{n}\left(f(\mathbf{x}_i) - \frac{1}{n}\sum_{j=1}^{n}f(\mathbf{x}_j)\right)^2}} \quad \in [0, 1]$
Interpretability. A good value should be close to 1

3.2 Classification

Performance is evaluated by looking at the numbers of samples correctly classified

Confusion matrix

	0	1
0	True Negative (TN)	False negative (FN)
1	False Positive (FP)	True Positive (TP)

Accuracy

$\frac{TP+TN}{TP+TN+FP+FN}$

Recall, Sensitivity or True Positive Rate (TPR)

$\frac{TP}{TP+FN}$

Specificity or True Negative Rate (TNR)

$\frac{TN}{TN+FP}$

Precision

$\frac{TP}{TP+FP}$

F-score or F1 score

$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

Receiver Operator Characteristic Area under the Curve (ROC - AUC)

Obtained by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR)
$\text{False Positive Rate } = 1 - \text{True Negative Rate}$

During the second part of the tutorial we corrected Problem 3 of Homework 1