Introduction to Machine Learning

Lecture 5: Model Selection and Validation

Alexis Zubiolo
alexis.zubiolo@gmail.com

Data Science Team Lead @ Adcash

December 1, 2016

Shameless advertisement: There will be a more advanced course starting in January 2017!

More info:

http://itstep.bg/news-bg/kurs-machine-learning-from-scratch/

Introduction

Model evaluation is a key component of a machine learning pipeline.

Introduction

Model evaluation is a key component of a machine learning pipeline.

It makes it possible to chose a set of hyperparameters so that

- The model is accurate enough
- The model generalizes well (*i.e.* does not overfit)

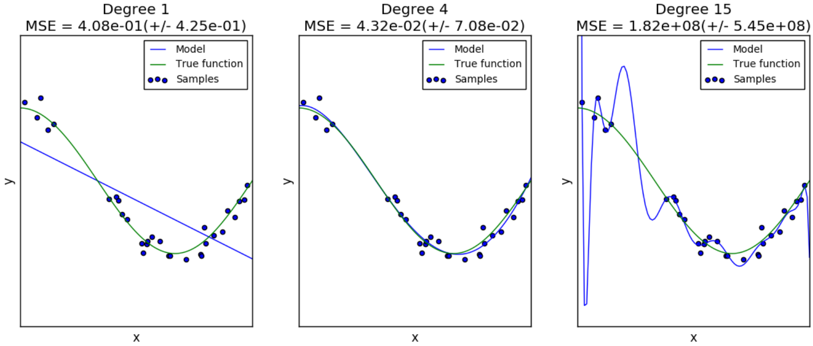
Introduction

Model evaluation is a key component of a machine learning pipeline.

It makes it possible to chose a set of hyperparameters so that

- The model is accurate enough
- The model generalizes well (*i.e.* does not overfit)

Recall from lecture 3:



Course outline:

- Evaluation metrics, what they mean
- How/when/why yo apply them

Evaluation metrics

Applying evaluation metrics

Train-test split

Reminder: ML algorithms (classification/regression) often rely on **many parameters**. How to **tune** them properly given a dataset?

Train-test split

Reminder: ML algorithms (classification/regression) often rely on **many parameters**. How to **tune** them properly given a dataset?

The most commonly used principle is the train-test split:

- **Split the data** into a training set and a test set
- **Train** on the training set
- **Test** on the test set

Train-test split

Reminder: ML algorithms (classification/regression) often rely on **many parameters**. How to **tune** them properly given a dataset?

The most commonly used principle is the train-test split:

- **Split the data** into a training set and a test set
- **Train** on the training set
- **Test** on the test set

This is often referred to as **cross-validation**.

Cross-validation

Standard technique: k-fold cross-validation

- Split the data into k equally sized folds
- Remove 1 fold (= test fold)
- Train on the other folds
- Test of the

Cross-validation

Standard technique: k-fold cross-validation

- Split the data into k equally sized folds
- Remove 1 fold (= test fold)
- Train on the other folds
- Test of the

Note: It is often advised to perform a **stratified** cross-validation, *i.e.* each fold contains approximately the **same percentage** of samples of each target class **as the complete set**.

Small dataset

Suppose you have dataset to train on. How to evaluate a classifier in this case?

Small dataset

Suppose you have dataset to train on. How to evaluate a classifier in this case?

k-fold cross-validation? Even with $k = 2$, it would make the training set even smaller and make it hard to fit a proper model.

Small dataset

Suppose you have dataset to train on. How to evaluate a classifier in this case?

k-fold cross-validation? Even with $k = 2$, it would make the training set even smaller and make it hard to fit a proper model.

Other option: **Leave-one-out** (LOO) cross-validation:

Small dataset

Suppose you have dataset to train on. How to evaluate a classifier in this case?

k-fold cross-validation? Even with $k = 2$, it would make the training set even smaller and make it hard to fit a proper model.

Other option: **Leave-one-out** (LOO) cross-validation:

- **Remove 1 sample** from the dataset
- Train on **all the other samples**
- Test on the sample you've removed
- **Evaluate** the prediction
- Do it **for each sample of the dataset**
- **Aggregate** the evaluations

Small dataset

Suppose you have dataset to train on. How to evaluate a classifier in this case?

k-fold cross-validation? Even with $k = 2$, it would make the training set even smaller and make it hard to fit a proper model.

Other option: **Leave-one-out** (LOO) cross-validation:

- **Remove 1 sample** from the dataset
- Train on **all the other samples**
- Test on the sample you've removed
- **Evaluate** the prediction
- Do it **for each sample of the dataset**
- **Aggregate** the evaluations

Remark: This could lead to many iterations even if the dataset is small.

Parameter selection

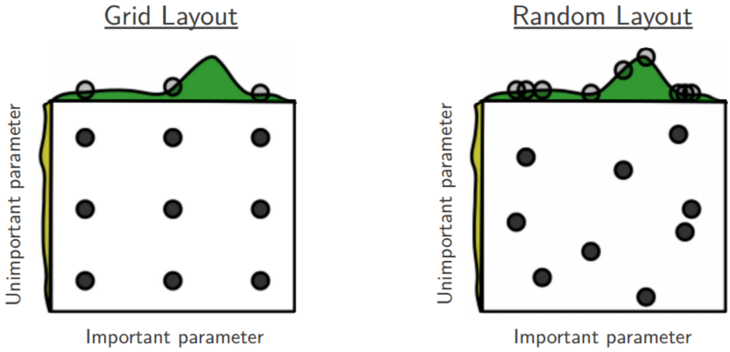
One of the goals of model evaluation is to select a **good model**.

Parameter selection

One of the goals of model evaluation is to select a **good model**. Hence we want to choose one (or several) hyperparameters. How do we do?

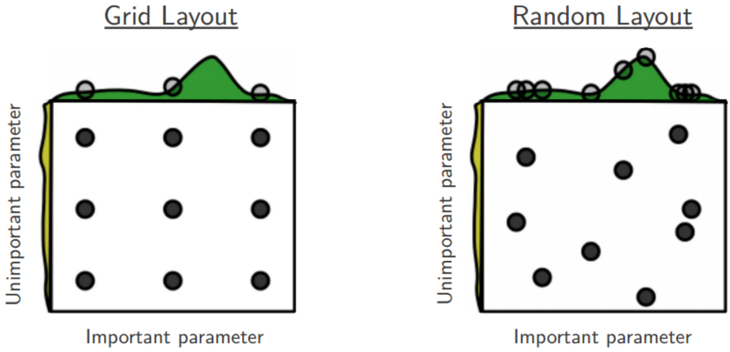
Parameter selection

One of the goals of model evaluation is to select a **good model**. Hence we want to choose one (or several) hyperparameters. How do we do?



Parameter selection

One of the goals of model evaluation is to select a **good model**. Hence we want to choose one (or several) hyperparameters. How do we do?



In any case, you need to know upper/lower bounds on the parameters

Conclusion

There are several ways to evaluate a model depending on

- **Your data**
- **What you value** in your application

Conclusion

There are several ways to evaluate a model depending on

- **Your data**
- **What you value** in your application

Think about this before applying a **random algorithm** and evaluating it with a **random metric**!

Thank you! Questions?