Machine Learning Samenvatting

Samenvatting

Manuel Mol .

Samenvatting

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Samenvatting

Published May 25, 2024

Introduction:

Some important definitions for machine learning are:

- **Tasks:** Tasks are problems that can be solved with machine learning
- **Model**: A model is a mathematical representation of a real-world process. It is a set of rules that describe the relationship between input and output variables.

Tasks:

There are a few different types of tasks in machine learning. The first is **predictive tasks.** Here we predict a target value from a number of features:

- **Regression:** The target value is continuous
- **Classification:** The target value is discrete
- **Predictive clustering:** The target value is a cluster

We also have **descriptive tasks,** where we describe the data according to some underlying structure:

- **Descriptive clustering:** We cluster the data
- **Association rule learning:** We find rules that describe the data
- **Subgroup discovery:** We find subgroups of the data

Another way to categorize tasks is by the way they learn:

- **Supervised learning:** The model is trained on labeled data
- **Unsupervised learning:** The model is trained on unlabeled data

Model:

A model is a mathematical representation of a real-world process. It is a set of rules that describe the relationship between input and output variables. You can categorize models by there **main intuition:**

- **Geometric model:** The model is based on geometric function such as distance. All instances can be represented in in **instance space**. An example of a geometric model is the **Linear classifier** model.
- **Probabilistic models:** aim for reducing uncertainty using probability distributions. An example of a probabilistic model is the **Naive Bayes** model.
- **Logical models:** use logical expressions to describe the relationship between input and output variables. An example of a logical model is the **Decision tree** model.

You can alo categorize the models by the **modus operandi** (mode of operation):

- **Grouping models:** dividing the instance space into segments; in each segment a very simple (e.g., constant) model is learned. An example of a grouping model is the **tree model**. It can’t distinguish between individual instances beyond this resolution.
- **Grading models:** A single, global model over an instance space. An example of a grading model is the **linear model**. It can distinguish between individual instances and the resolution is not limited.

Model phases:

- **Training/learning:** where te model is trained on the data
- **Inference:** where the model is used to make predictions

Features:

A measurement that can be performed on any instance. When features are not in the correct form, they can be transformed into a new feature space:

- **Feature construction:** turn images into pixels etc.
- **Discretisation:** Numerical features are transformed into categorical features
- **Feature transformation:** project the data into a new feature space
- **Feature selection:** removing irrelevant features

Binary classification:

In binary classification, the target value is binary.

How can we evaluate performance?

Contingency table: A table that shows the number of true positives, false positives, true negatives, and false negatives. We can calculate a few metrics from this table:

- **Accuracy:**

$$\text{Accuracy} = \frac{\text{correct}}{\text{total}} \tag{1}$$

- **Error rate:**

$$\text{Error rate} = \frac{\text{incorrect}}{\text{total}} \tag{2}$$

- **True positive rate / Sensitivity / Recall:** How many sick people are identified as having the illness, How many relevant items are selected

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

- **True negative rate / Specificity:** How many negative items are selected that are truly negative

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4}$$

- **False positive rate:**

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{5}$$

- **False negative rate:**

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{6}$$

- **Precision:** How many selected items are truly relevant

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

- **F score:** The harmonic mean of precision and recall

$$F = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

We can use a train test split to evaluate the model. We can split the data in a training set and a test set. We can then train the model on folds of the training set and evaluate the model on the test set.

We can also use a separate validation set to tune the hyperparameters of the model.

Train set error: The error on the training set
Test set error: The error on the test set

Overfitting: When the model performs well on the training set but poorly on the test set. This is because the model has learned the noise in the training set.
Underfitting: When the model performs poorly on the training set. This is because the model is too simple.

ROC curve: A curve that shows the true positive rate against the false positive rate. The area under the curve is a measure of the model's performance. The closer the area is to 1, the better the model. The points on the curve are the result of changing the threshold of the model. If the line is above the diagonal, the model is better than random.

Scoring and ranking:

A **scoring classifier** is a classifier that outputs a score for each instance.

Margin: The distance between the score of the correct class and the score of the incorrect class. The margin is a measure of the confidence of the model.
Loss function: A function that measures the error of the model.

Ranking: The order of the instances based on the score of the classifier.

Ranking error rate: The error rate of the ranking.

Class probability estimation:

The probability that an instance belongs to a certain class. The output of the classifier shows how likely it is that the instance belongs to a specific class rather than which class it will belong to.

To determine how good these probabilities are, we can use the **squared error loss function**. This function measures the difference between the predicted probability and the true probability. The **mean squared error** is the average of the squared error loss function.

Multi-class classification:

In multi-class classification, the target value is a class from a set of classes. There are 2 different types of algorithms for multi-class classification:

- **Inherently non-binary:** Algorithm like decision trees
- **Inherently binary:** Support Vector Machines

Turning a binary classifier into a multi-class classifier can be done. There are 2 main approaches:

- **One-vs-all:** Train a binary classifier for each class. You get an incorrect classification if one of the classifiers classifies the instance as the class incorrectly.
- **One-vs-one:** Train a binary classifier for each pair of classes

$$\text{num}_c = \frac{k(k-1)}{2} \tag{9}$$

num_c classifiers are needed to create a symmetric multi-class classifier with k classes.

One versus one is more accurate than one versus all, but one versus all is faster.

How to measure performance:

Accuracy can still be calculated in the same way as in binary.

Precision and recall are calculated for each class. The precision of the model is then calculated by taking the average of the precision of each class. The same goes for recall.

We can still use an **ROC curve** to evaluate the model. The ROC curve is calculated for each class. The area under the curve is then calculated by taking the average of the area under the curve of each class.

Regression:

Regression is a predictive task where the target value is continuous.

Regression functions are evaluated by using a loss function on the residuals. The residuals are the difference between the predicted value and the true value.

To prevent overfitting, the number of parameters should be considerably less than the number of data points

Bias: refers to the error introduced by approximating a real-world problem, which may be complex, by a simplified model. High bias means the model makes strong assumptions about the data, which can lead to systematic errors and underfitting.

Variance: refers to how much your model's predictions change when you use different training data. High variance means the model is very sensitive to the training data and may not perform well on new, unseen data

$$\begin{gathered}\text{High variance} = \text{overfitting} \\ \text{High bias} = \text{underfitting}\end{gathered} \tag{10}$$

Unsupervised and descriptive learning:

In descriptive learning the task is to come up with a description of the data. We can use clustering to group the data.

Clustering:

Clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups.

To evaluate performance with a ground truth we can use the **Rand index.** which is similar to the accuracy in classification. We can also use **purity**.

$$\text{purity} = \frac{\text{most frequent in cluster}}{\text{total}} \tag{11}$$

How can we check the performance without **ground truth?**

Silhouette Coefficient:

$$s = \frac{b - a}{\max(a, b)} \tag{12}$$

Where:
a: the mean distance between an instance and all other points in the same cluster.
b: the mean distance between an instance and all other points in the next nearest cluster.

The silhouette ranges from −1 to +1. A high value indicates that the instance is well matched to its own cluster and poorly matched to neighboring clusters.

Subgroup discovery:

Subgroup discovery is a supervised learning task but it is different from classification, as it addresses different goals → discovery of interesting population subgroups instead of maximizing classification accuracy of the induced rule set. *Finding patterns in traffic accident*

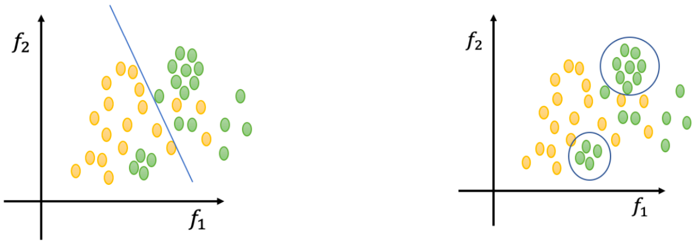

Figure 1: **(left)** classifier **(right)** subgroup discovery

A **Chi-squared test** can be used to determine if the subgroup is significant.