**2**

This chapter

Sentiment analysis (also Opinion Mining)

* Text Mining in general
  + Information Retrieval as first step OR database
  + Linguistic analysis, NLP (POS-tagging)
* TM process
  + Structure data
  + find patterns
  + evaluate results
* TM topics
  + text categorization
  + text summarization
  + entity relation modeling
  + sentiment analysis
  + …
* TM concepts?
  + TFIDF

Plutchik

8 created emotions 🡪 Buch6

resources at WordNet

Machine learning

supervised/unsupervised/…

regression/classification

Ordinal classification / Multiple Class Classification

Cost-sensitive learning

In many classification problems the goal is the optimization of some performance measurement, most often the accuracy. However, the best solution in respect to the accuracy does not have to be the one with minimized costs, which is also often a goal to be achieved. Costs can not only arise from misclassifications, but also from additional tests or time for the creation of a better model. \citeauthor{Turney2002} divides the reasons for costs that a machine learning problem can entail into nine groups \autocite{Turney2002}:

\begin{itemize}[noitemsep]

\item misclassification errors

\item tests

\item teacher

\item intervention

\item unwanted achievements

\item computation

\item cases

\item human-computer interaction

\item instability

\end{itemize}

In each of the nine groups, we can distinguish between constant and conditional costs. In the case of test costs, we speak of constant costs if a certain test has a fixed cost value for its execution. Note that different test may have different costs if they still remain constant for all instances in all circumstances. On the contrary, conditional test costs depend on a criterion. For a medical test this criterion might be the test result, the age of a patient or the result of prior tests. In the remaining subsection, of all cost types only the misclassification costs are discussed and therefore abbreviated to costs.

The standard approach to evaluate the performance of a classifier is the calculation of the accuracy, i.e. the number of correct classified instances in relation to the total number of instances. The error costs result directly from the number of misclassified instances, which only has to be multiplied by a constant cost factor. Thus, \citeauthor{Turney2002} calls this value constant misclassification error cost. However, there are various problems in machine learning where such a simplification would lead to a miscalculation of the total costs because the conditional error costs are ignored.

In many cases, the equal treatment of all error types is the cause of the incorrect cost estimate. For this we first consider the binary classification with only two classes and therefore only two possible error types. A false positive (FP) occurs when the outcome is incorrectly predicted as yes (or positive) when it is actually no (or negative). The reverse case is called false negative (FN). False positives and false negatives have rarely equal costs in real-world applications. In spam-classification, a no-spam e-mail classified as spam is worse than a spam e-mail which have not been detected correctly. An accepted customer which is not capable of paying back a loan has bigger costs than a customer wrongly classified as insolvent. A sick patient who is classified as healthy is more problematic than a healthy patient who is classified as sick. In order to map this, a cost value is determined for each of the two errors FP and FN or the cost ratio is determined.

If we transfer the cost calculation to the multiclass classification, there are more error types and therefore more cost factors to be defined: Each instance of class $i$, misclassified as class $j$ will be multiplied with the correspondent cost factor $c\_{ij}$. For a multiclass classification problem with $n$ classes we get a $n\timesn$ cost matrix similar to the confusion matrix, which has $n(n-1)$ possibly different cost factors and whose main diagonal contains zeros.

A special case arises for classification problems with ordinal classes. The value of $c\_{ij}$ should reflect the extent of the difference between $i$ and $j$. Thus, it is common to assume that $c\_{ij}=0$ when $i=j$. In addition, the cost $c\_{ij}$ is assumed to be larger when $i$ is further away from $j$. Two common functions satisfy the requirements and have been widely used in practice:

\begin{itemize}[noitemsep]

\item \makebox[7cm][l]{Absolute cost vectors} $c\_{ij}=\left|i-j\right|$

\item \makebox[7cm][l]{Squared cost vectors} $c\_{ij}=\left(i-j\right)^{2}$

\end{itemize}

Note that the squared cost charges more than the absolute cost when $i$ is further away from $j$. The cost matrix of a multiclass classification problem with $n$ classes using absolute cost vectors looks like this \autocite[Section~3]{Kotsiantis2004}:

$\left[ \begin{array}{ccccc}

{0} & {1} & {2} & {\dots} & {n-1} \\

{1} & {0} & {1} & {\dots} & {n-2} \\

{\dots} & {\dots} & {\dots} & {\dots} & {\dots} \\

{n-2} & {\dots} & {1} & {0} & {1} \\

{n-1} & {\dots} & {2} & {1} & {0}

\end{array}\right]$

By squaring each element of the matrix, one obtains the matrix of squared cost vectors. In both cases it is a symmetrical matrix, i.e. $c\_{ij}=c\_{ji}\foralli,j\in\{1, \dots, m\}$. In some cases, it may be useful to weight the over- and underestimation of classes differently. If the overestimation of a class is to be penalized more severely than the underestimation, all elements to the right of the main diagonal are multiplied by a constant value $\lambda>1,\lambda\in\mathbb{R}$. In a comparable way all values of the matrix can be adjusted as required to the problem which has to be solved.

Given such a cost matrix, you can calculate the cost of a particular learned model on a given test set just by summing the relevant elements of the cost matrix for the model’s prediction for each test instance. Here, the costs are ignored when making predictions, but taken into account when evaluating them \autocite[Chapter~5.7]{Witten2005}. In this way the classifier obviously does not have to deliver the cost-optimal result. However, there are different possibilities to include costs in advance \autocite[Chapter~2.3]{Qin2010}:

\begin{itemize}[noitemsep]

\item Change of the class distribution of the training data

This approach incorporates the misclassification cost into the data pre-processing step by re-sampling or re-weighting the training data in proportion of their misclassification cost.

\item Modification of the learning algorithm

This approach modifies the error-based classifiers directly to handle misclassification cost, but each classifier needs to be modified separately.

\item Boosting approach

This approach generates a set of different weak classifiers in sequential trial and then constructs a composite classifier by voting them. The advantage of this approach is that it is applicable to any kind of error-based classifiers.

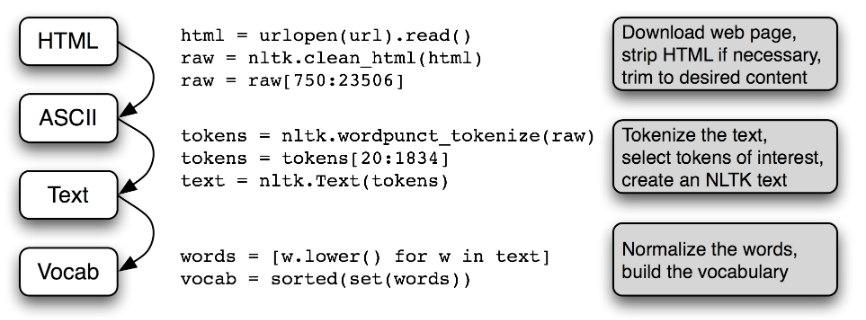
\item Conditional probability estimates / Direct cost-sensitive learning

This approach incorporates the misclassification cost into the data post-processing step by using the probability estimation generated by error-based classifiers and the cost function to directly compute the optimal class for each test example. This approach is easy to implement, but needs good calibration methods to generate accurate probability estimation.

\end{itemize}

Note that the use of the costs as in one of the approaches presented here does not preclude a subsequent evaluation of the costs. In this way it can be checked whether a classifier who incorporates the costs into the learning process is actually superior to a classifier without knowledge of the costs.

NLP & NLTK



further reading chapter 7 (POS-tagging, chunking)

Word2Vec / Word Embedding

neuronal network with two layers

idea: word vector, the more similar words, the more similar the vectors

adding of vectors

comparison of vectors based on angle (0°-90°(?))

CBOW (context 🡪 word) vs skip-gram (word 🡪 context)

glove