Multimedia Retrieval

Chapter 1: Performance Evaluation

1.1 Introduction
1.2 Defining a Benchmark for Retrieval
1.3 Boolean Retrieval
1.4 Retrieval with Ordering
1.5 Perfomance of Machine Learning
1.6 Literature and Links

Dr. Roger Weber, roger.weber@ubs.com

1.1 Introduction

- In this course, we investigate a number of retrieval models, feature extraction algorithms, and search algorithms. At some point, we need to understand the performance of an approach to determine what is the best retrieval method. Often, there is no absolute answer but the results my vary in the context of their application:
  - Vector space retrieval was proven to outperform Boolean retrieval many times (similarly: probabilistic retrieval). Nevertheless, web search engine such as AltaVista (used vector space retrieval) and Inktomi (probabilistic retrieval) could not compete with Google (Boolean retrieval)
  - When searching for similar images (still photos), it is well accepted that color is more important than texture, and texture is more important than shape. In medical imagery, however, the contrary is true: color is often meaningless (X-ray, MRI, CT, ...), texture often plays an important role to detect the type of tissue, but shape is of highest importance (e.g., skin cancer).
  - Machine learning with Deep Neuronal Networks outperforms most other classification methods, but comes at high computational costs. Is the additional effort worth the better performance, or is simpler & faster just good enough?
- We conclude that the performance of an approach depends on
  - the collection, and
  - the type of queries / learning scenarios
  - the information needs of users
  - … and some non-functional constraints

  With other words: for each retrieval and learning task, a new evaluation is required to determine the best approach. Generalization do work to a certain degree, but not always.

- Evaluation of retrieval systems differentiates between two types:
  - Boolean approaches that return an (unordered) set of documents, and
  - Retrieval approaches that return a ranking of documents ordered by their relevance for the current information need (i.e., how well it matches the query)

  An important criteria of the evaluation is the so-called relevancy ordering, i.e., the information whether and how well a document matches the query. We may use a simple black & white view, i.e., the document is "relevant" or "not relevant". Or, we can employ a more sophisticated approach with pair-wise assessment of documents with regard to their relevance to the query. For example, the preference pair ($A <_p B$) denotes that "B is more useful/relevant than A". A retrieval system works well (or as expected), if it ranks B before A, and does similarly for any other given pair of preference.

  Defining a sound benchmark is the first step of an evaluation. A benchmark consists of a collection, query types, and relevance assessments. There are a number of benchmarks available for different retrieval areas. In the following, we shortly consider the benchmark provided by INEX.

- Evaluation of learning methods depend on the desired task and output. Assessment of
  - binary classification is very similar to Boolean retrieval (precision, recall)
  - multi-class classification uses so-called confusion matrices to understand for which combinations of classes the algorithm performs good/bad
  - classification with scores and thresholds requires us to determine good thresholds and a metric to compare different methods (given different thresholds)
  - classification with probability distributions is often based on entropy (log-loss)
  - regression tasks (fitting real valued output data) uses mean squared error (MSE)
  - deep learning employs a whole bag of methods to define what "good" means

1.2 Defining a Benchmark for Retrieval

- So what makes a good benchmark? First of all, we need a sound collection that provides a rich set of different documents. We also need queries (and many of them) covering various aspects of the retrieval task. We further need to have assessments of all documents against all queries. And, finally, we need an evaluation method to capture the essence of the challenge.
  - Challenge 1: non trivial queries that can distinguish different methods. Queries that are easily answered by all methods do not add much differentiation.
  - Challenge 2: how do we find the "correct" answers for all queries given the size of the collection. For instance, if you evaluate web search engine, how do you know what should be the best answer to a query?

- Here is an example form the past: INEX started in 2002 to provide a yearly competition among research groups focusing on XML Retrieval (similar to TREC from the classical area). To define the challenge, the following steps were taken (see next pages for details):
  - Selection of an appropriate collection
  - Definition of queries
  - Relevance assessments for each query over the collection
  - Evaluation method (see Section 1.3ff)

- The collection for competition in 2002 consisted of 12’107 Articles of IEEE journals between 1995 and 2001 (about 500 MB).

```
<article>
  <fm>
    <ti>IEEE Transactions on ...<ti>
    <atl>Construction of ...</atl>
    <au>
      <fnm>John</fnm><snm>Smith</snm>
      <aff>University of ...</aff>
    </au>
    <au>...</au>
  </fm>
  <bdy>
    <sec>
      <st>Introduction</st>
      <p>...</p>  ...
    </sec>
    <sec>
      <st>...</st>  ...
      <ss1>...</ss1>
      <ss1>...</ss1>  ...
    </sec>  ...
  </bdy>
    <bm>
      <bib>
      <bb>
      <au>...</au><ti>...</ti>
      ...
      </bb>
      ...
      </bib>
    </bm>
</article>
```

- There were two types of queries: "Content-and-structure" (CAS) queries, and "Content-only" (CO) queries. An example for a CO-Query was (about 30 such queries were defined):

```
<INEX-Topic topic-id="45" query-type="CO" ct-no="056">
  <Title>
    <cw>augmented reality and medicine</cw>
  </Title>
  <Description>
    How virtual (or augmented) reality can contribute to improve the medical and
    surgical practice. and
  </Description>
  <Narrative>
    In order to be considered relevant, a document/component must include
    considerations about applications of computer graphics and especially
    augmented (or virtual) reality to medecine (including surgery).
  </Narrative>
  <Keywords>
    augmented virtual reality medicine surgery improve computer assisted aided
    image
  </Keywords>
</INEX-Topic>
```

- An example of a CAS-Query is given below (about 30 such queries existed):

```
<INEX-Topic topic-id="09" query-type="CAS" ct-no="048">
  <Title>
      <te>article</te>
      <cw>nonmonotonic reasoning</cw> <ce>bdy/sec</ce>
      <cw>1999 2000</cw> <ce>hdr//yr</ce>
      <cw>-calendar</cw> <ce>tig/atl</ce>
      <cw>belief revision</cw>
  </Title>
  <Description>
      Retrieve all articles from the years 1999-2000 that deal with works on
      nonmonotonic reasoning. Do not retrieve articles that are calendar/calls for
      papers.
  </Description>
  <Narrative>
      Retrieve all articles from the years 1999-2000 that deal with works on
      nonmonotonic reasoning. Do not retrieve articles that are calendar/calls for
      papers.
  </Narrative>
  <Keywords>
      nonmonotonic reasoning belief revision
  </Keywords>
</INEX-Topic>
```

- How do we get the relevance assessments? Do we really have to assess each of the 12’107 articles for each of the 60 queries? Such an approach is clearly too labor intensive to be practical…
- A better approach is the following one rather than evaluating absolute performance, relative performance is sufficient. Assume each retrieval method returns a set of documents but misses one important answer. Clearly, that missed answer will not change the relative ordering of the methods. We conclude from this observation, that the relative ordering of the methods only depend on the set of documents returned by any of the methods (the union of all results). This massively simplifies the approach. Furthermore, to avoid any bias in the relevance assessment towards one or the other method, each participant has to assess the results for a subset of the queries. In summary, the approach taken by INEX is (and similar approaches are taken by other benchmarks):
  - The coordinator selects a collection, defines the queries (sometimes submitted by the participants), and sets an evaluation metric (usually precision/recall graphs)
  - Each participant evaluates all queries with its retrieval method, and submits its result lists to the coordinator
  - The coordinator then asks each participant to assess a subset of the queries against the union of returned answers of methods in the competition (typically this is well below a 1000 documents)
  - The assessment results are collected by the coordinator who then computes the performance value for each participant

1.3 Boolean Retrieval

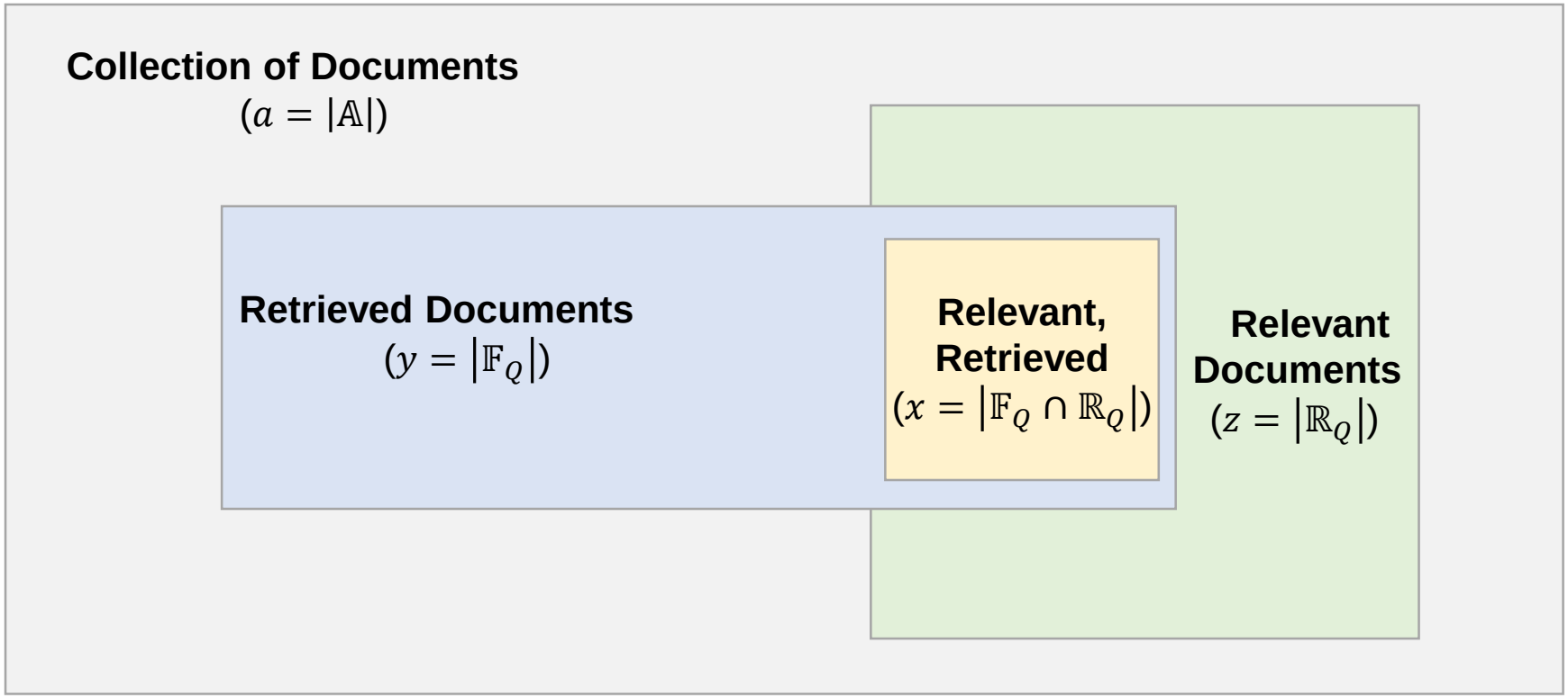
- Boolean retrieval simply returns a set of documents without ordering them. In other words, the retrieval method does not distinguish between "highly relevant" and “maybe relevant"
- **Precision** and **recall** are the most prominent measures used for the evaluation of algorithms. Precision denotes how many answers of a system are actually relevant from a user's perspective. Recall describes the percentage of retrieved and relevant answers over all relevant documents in the collection. A further measure, **fallout**, is used to describe a system's ability to discard non-relevant documents to the users (false hits).
- Notations:

$\mathbb{A}$	Set of all documents
$\mathbb{R}_Q$	Set of relevant documents for a query Q in the collection $\mathbb{A}$
$\mathbb{F}_Q$	Set of documents retrieved by a system for query Q

- Then, precision p, recall r and fallout f are defined as follows:

$$p = \frac{|\mathbb{F}_Q \cap \mathbb{R}_Q|}{|\mathbb{F}_Q|} \qquad r = \frac{|\mathbb{F}_Q \cap \mathbb{R}_Q|}{|\mathbb{R}_Q|} \qquad f = \frac{|\mathbb{F}_Q \setminus \mathbb{R}_Q|}{|\mathbb{A} \setminus \mathbb{R}_Q|}$$

- Visualization



Precision: $p = \frac{x}{y}$ Recall: $r = \frac{x}{z}$ Fallout: $f = \frac{y - x}{a - z}$

- Next to precision, recall and fallout, literature names a few further measures. We will see more definitions when we measure the performance of machine learning tasks.
  - **Total Recall:** (how many relevant documents are in the collection?)

$$g = \frac{|\mathbb{R}_Q|}{|\mathbb{A}|}$$

It follows that:

$$f \cdot p \cdot (1 - g) = r \cdot g \cdot (1 - p)$$

  - **F-Measure:** Combines Precision and Recall to a single value. The parameter β determines how more important Recall over Precision shall be. With $\beta = 0$ only Precision counts; with $\beta = \infty$ only Recall counts.

$$F_\beta = \frac{(\beta^2 + 1) \cdot p \cdot r}{\beta^2 \cdot p + r}$$

The larger the F-Measure, the better an algorithm or system works. A typical value is $\beta = 1$. Having a single measure instead of two values simplifies comparisons; β is pushing either precision (need some relevant documents) or recall (need all relevant documents).

- Usually, we are not just using a single experiment to assess the performance of methods. Rather, we run a series of queries and then compute an "average" precision and recall. Let N be the number of queries, and for each query Q_i, we obtain a set $\mathbb{F}_i$ (retrieved documents for query Q_i) and a set $\mathbb{R}_i$ (relevant documents for query Q_i). For each query, we can compute the precision-recall pair (p_i, r_i). To obtain an average value, two method exist:
  - **Macro Evaluation**: p and r are given as average value over p_i and r_i , respectively:

$$p = \frac{1}{N}\sum_{i=1}^{N} p_i = \frac{1}{N}\sum_{i=1}^{N} \frac{|\mathbb{F}_i \cap \mathbb{R}_i|}{|\mathbb{F}_i|} \qquad r = \frac{1}{N}\sum_{i=1}^{N} r_i = \frac{1}{N}\sum_{i=1}^{N} \frac{|\mathbb{F}_i \cap \mathbb{R}_i|}{|\mathbb{R}_i|}$$

  - **Micro Evaluation**: summing up numerators and denominators leads to:

$$p = \frac{\sum_{i=1}^{N}|\mathbb{F}_i \cap \mathbb{R}_i|}{\sum_{i=1}^{N}|\mathbb{F}_i|} \qquad r = \frac{\sum_{i=1}^{N}|\mathbb{F}_i \cap \mathbb{R}_i|}{\sum_{i=1}^{N}|\mathbb{R}_i|}$$

The micro evaluation is more stable if the sets $\mathbb{F}_i$ and $\mathbb{R}_i$ vary significantly in size.

1.4 Retrieval with Ordering

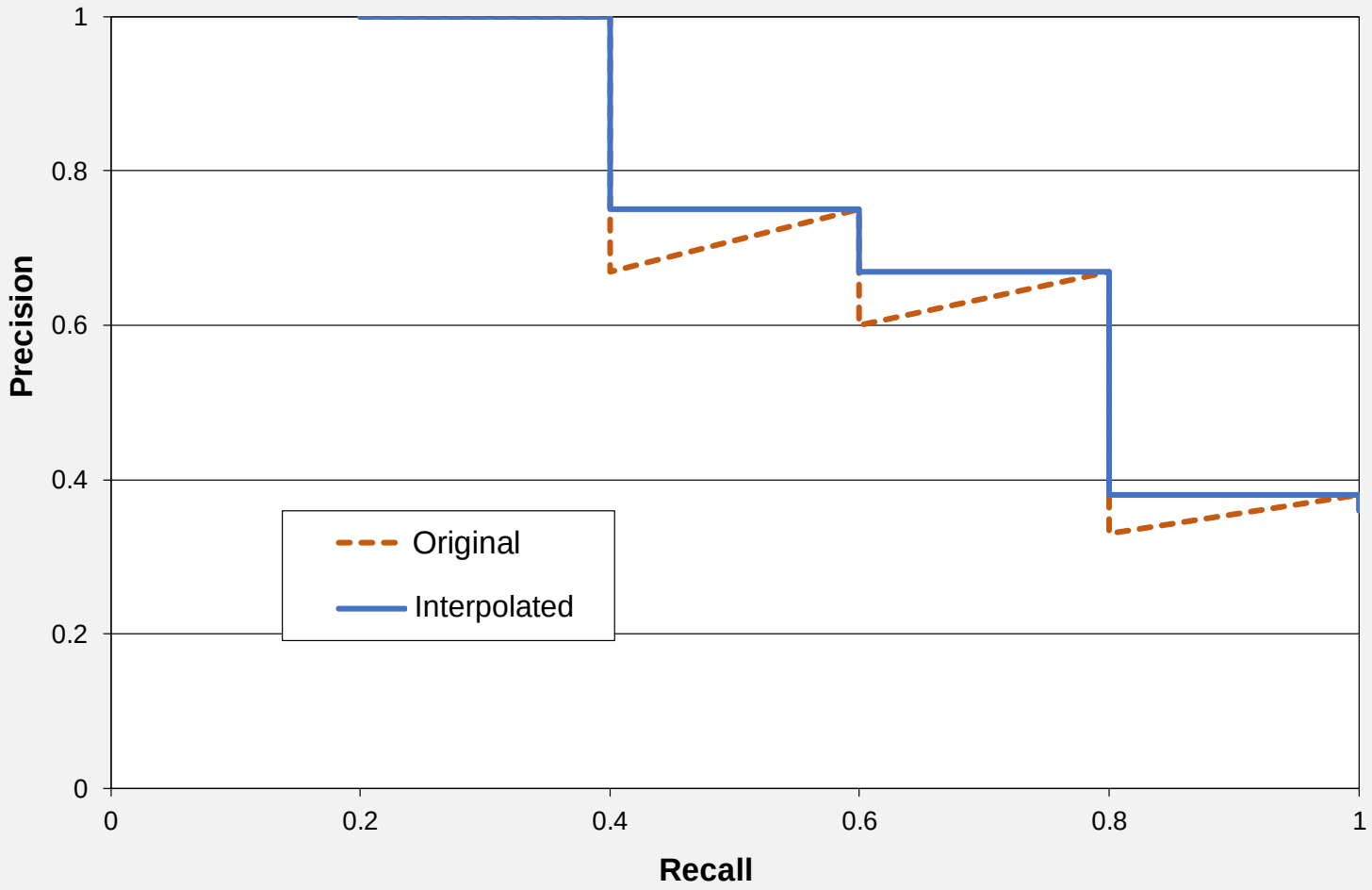
- Most retrieval methods return a ranked list, and we want to take the ranking somehow into account. Intuitively, a method that has a lot of relevant documents at the top of the list is perceived better than a method that shows the relevant document only later in the list.
- The precision-recall curve addresses this as follows: at each rank position, the precision and recall up to this point is computed. These precision-recall pairs are then depicted in a two dimensional plot. Let's look at an example: assume that the collection has exactly 5 relevant documents for a query Q and a retrieval system produces the following ranked list:

$rank$	docID	relevance	p_i	r_i
1	588	x	1.00	0.20
2	589	x	1.00	0.40
3	576		0.67	0.40
4	590	x	0.75	0.60
5	986		0.60	0.60
6	592	x	0.67	0.80
7	984		0.57	0.80
8	988		0.50	0.80
9	578		0.44	0.80
10	985		0.40	0.80
11	103		0.36	0.80
12	591		0.33	0.80
13	772	x	0.38	1.00
14	990		0.36	1.00

P-R pair for the first 4 documents: we observe 3 relevant documents, hence $p = 3/4$, and we have seen 3 of 5 relevant documents, hence, $r = 3/5$.

(generally: we compute p and r for the first $\boldsymbol{rank}$ documents in the result)

- We now can draw the P-R pairs of the example in a 2-dimensional plot. Notice that recall values only increase while precision values increase whenever a new relevant document is in the list, and decrease otherwise. To smooth the P-R curve, we often interpolate the values to obtain a step curve as depicted below in blue.



- Interpretation of P-R-Curve:
  - Close to $(r = 0, p = 1)$: most retrieved documents are relevant but not all relevant documents were found. This case is optimal for queries where one is just interested in a correct answer, for example: "is this mushroom poisonous"
  - Close to $(r = 1, p = 0)$: all relevant documents were retrieved but lots of the retrieved document are non-relevant. High recall is important for queries like "is there a patent"
  - $p = 1$ is usually difficult to achieve; $r = 1$ is simple: just return all documents
- To simplify comparison and ranking, we want to obtain a single value out of the many precision-recall pairs. Intuitively, we want to favor high precision and recall values. But given the observations above, high recall values are only seldom required. More frequently, we want to favor a high precision with a reasonable recall. Thus, there are different ways to summarize pairs:
  - **System Efficiency:** prefers an ideal system that returns all relevant and only relevant documents. That is, we prefer both high precision and high recall values. In the precision-recall plot, if the curve of a method A lies closer to the point $(r = 1, p = 1)$ than the curve of a method B, then we consider A to outperform B. Let d be the minimal distance of the precision-recall pairs to $(r = 1, p = 1)$. The system efficiency E is then given as:

$$E = 1 - \frac{d}{\sqrt{2}}$$

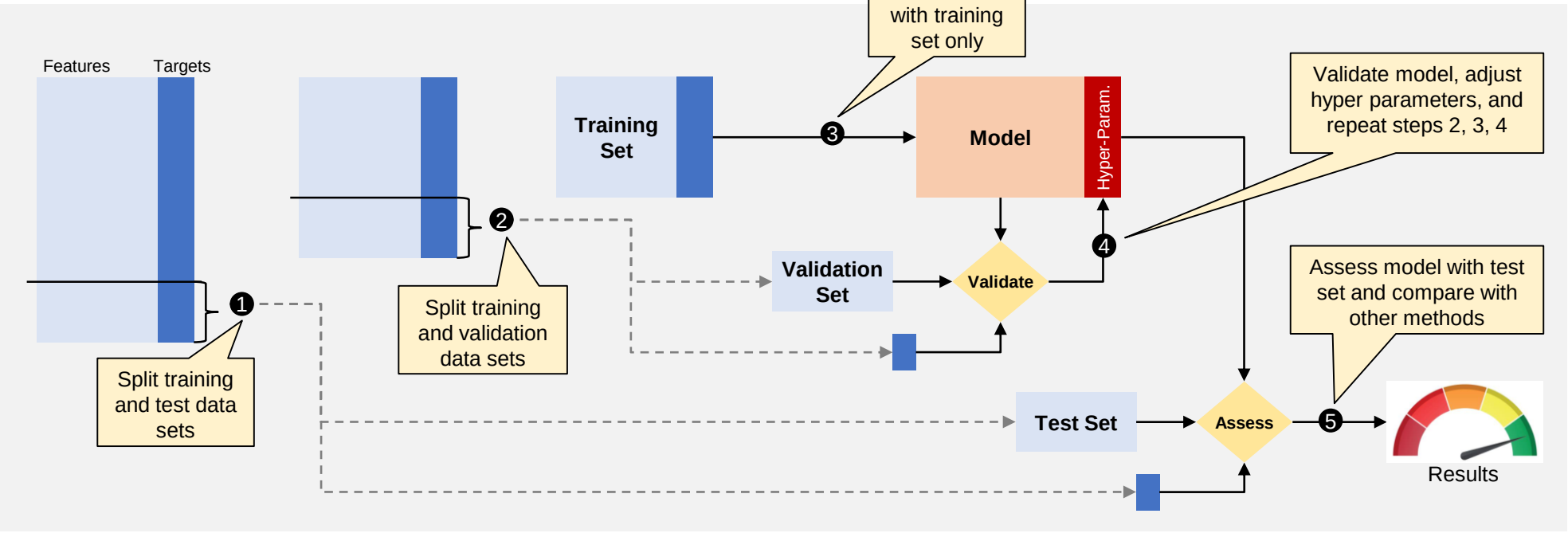
  - **R-Precision:** if we favor precision over recall, the R-Precision is a good alternative. It denotes the precision of a method after having retrieved a given percentage of all relevant documents:

$$RP = \max_{p,r} \begin{cases} 0 & if\ r < r_{threshold} \\ p & if\ r \geq r_{threshold} \end{cases}$$

- An other method to summarize the pairs is to compute the **Area Under the Curve** (AUC, we will see this later again for the ROC curve). Like with the system efficiency, an ideal system that only returns relevant documents and all of them will obtain the maximum value of 1. The method prefers high precision values across all recall values.

- As with Boolean retrieval, we conduct a benchmark with many experiments and obtain several sets of precision-recall curves. How do we "average" these different sets of pairs to obtain a meaningful average curve? Again, different methods exist depending on the objective of the benchmark:
  - Compute the average precision and recall values over all queries for the first 5, 10, 15, 20, …, results and use these average values for the precision-recall curve. The method is simple but sensitive to outliers.
  - Alternatively, determine the precision over all queries for fixed recall values (see R-Precision) and average the precision values to obtain the pairs for the fixed recall values. This method correspond to the approach of "averaging" the curves by drawing a vertical line and determine the mean precisions along the intersections of the vertical line with the precision-recall curves. The method is more robust to outliers and provides an intuitive meaning to what "average" is.

1.5 Perfomance of Machine Learning



- In machine learning, the performance measure is not only used for final evaluations. Some methods also employ performance metrics to validate some of the hyper-parameters of the model. This validation is used to prevent under-fitting and over-fitting to the training data and essentially alters the internal structure in a defined way. For example, using polynomial regression, the degree of polynoms is such a hyper-parameter.
- In addition, some methods like neural networks and regression use the performance metric as an error or loss function that needs to be optimized (find parameters/weights of the model that minimize the error). In some cases, we can use different metrics to train, validate, and test the system to optimize different aspects of the model.

- To evaluate (and improve) a machine learning algorithm, we need to provide a quantitative measure for the "accuracy" of carrying out the task T. Different types of measures exists:
- **Binary classification** (0-1 decisions) uses a **confusion matrix** to assess the performance, and provides numeric summary values to optimize for a desired optimum for the task

		Actual Condition (as observed)			
	Population	Positive (P)	Negative (N)		
Predicated Condition (as computed)	"Yes"	**True Positive** (TP)	**False Positive** (FP)	Positive Predictive Value (PPV), Precision	False Discovery Rate (FDR)
	"No"	**False Negative** (FN)	**True Negative** (TN)	False Omission Rate (FOR)	Negative Predictive Value (NPV)
		True Positive Rate (TPR), Sensitivity, Recall, Hit Rate	False Positive Rate (FPR), Fall-Out	Accuracy (ACC)	
		False Negative Rate (FNR), Miss Rate	True Negative Rate (TNR), Specificity		Error Rate (ERR), Misclassification Rate

$$TPR = \frac{TP}{P} \qquad FNR = \frac{FN}{P} = 1 - TPR \qquad ACC = \frac{TP + TN}{P + N}$$

$$TNR = \frac{TN}{N} \qquad FPR = \frac{FP}{N} = 1 - TNR \qquad ERR = \frac{FP + FN}{P + N} = 1 - ACC$$

$$PPV = \frac{TP}{TP + FP} \qquad FDR = \frac{FP}{FP + TP} = 1 - PPV$$

$$NPV = \frac{TN}{TN + FN} \qquad FOR = \frac{FN}{FN + TN} = 1 - NPV$$

– Example: Cancer test

	Population (*2030*)	Actual Condition (as observed): Positive (*P=30*)	Actual Condition (as observed): Negative (*N=2000*)	
Predicated Condition (as computed)	"Yes" (*200*)	**True Positive** (*TP=20*)	**False Positive** (*FP=180*)	$PPV = \frac{20}{200} = 10\%$ (precision)
	"No" (*1830*)	**False Negative** (*FN=10*)	**True Negative** (*TN=1820*)	$NPV = \frac{1820}{1830} = 99.5\%$
		$TPR = \frac{20}{30} = 67\%$ (recall)	$TNR = \frac{1820}{2000} = 91\%$	$ACC = \frac{1840}{2030} = 90.6\%$

– Is this a good test for cancer?
  - We note that the false discovery rate ($1 - PPV = 90\%$) is very high, i.e., a lot of tests are positive but the patient does not have cancer. Hence, there is little confidence in positive outcomes and further tests are required.
  - We further note that the false omission rate ($1 - NPV = 0.5\%$) is very low, i.e., a negative test result is almost always a true negative case. This is an important element of the diagnosis of exclusion, especially if the above test is very cheap to conduct. The high true negative rate ($TNR = 91\%$) indicates that the elimination is in 91% successful.
– Using NPV as a driving performance metric is very common in cases where most of the population is considered negative.
– Accuracy (ACC) is not a reliable metric: assume an "oracle" that always predicts "No". This oracle yields an accuracy of $\frac{0+2000}{2030} = 98.5\%$ and, hence, beating the predictions of the above example. On the other side, $PPV = 0\%$, $NPV = 98.5\%$, $TPR = 0\%$ and $TNR = 100\%$ clearly indicate the limitations of this oracle.

- **Multi-class classification** (one out of a set of classes) requires a generalized **confusion matrix** resulting in a table such as the following one for people recognition in images:

		Actual Class		
	Population (*100*)	Woman (*20*)	Man (*20*)	*Child (60)*
Recognized Class	Woman (*19*)	*13*	*4*	*2*
	Man (*18*)	*2*	*15*	*1*
	Child (*63*)	*5*	*1*	*57*

  - The confusion matrix allows to easily spot correct classifications (on the diagonal) and prediction errors (outside the diagonal). The table also allows to observe where the algorithm has troubles to distinguish classes. In the example above, the algorithm recognized
    - 13 of 20 women correctly found, but 2 were wrongly classified as man and 5 as children
    - 19 women recognized in total but only 68% (13) were actually women
    - 57 of 60 children correctly found, and children were more often confused with women than men
  - Accuracy is given by the sum of the diagonal over all examples, i.e., $ACC = \frac{13+15+57}{100} = 85\%$, and the error rate is $ERR = 1 - ACC = 15\%$. Again, accuracy alone is not capable to tell us the entire story as we see that the algorithm struggles with recognizing women correctly. To better analyze the situation, we can create additional confusion matrices focusing on the correct classification of one class only. See next page for an example for the class “Woman” and “Child”

		Actual Class		
	Total Population	Woman ($P=20$)	Not a Woman ($N=80$)	
Recognized Class	Woman (19)	**True Positive** ($TP=13$)	**False Positive** ($FP=6$)	precision $PPV = \frac{13}{19} = 68\%$
	Not a Woman (81)	**False Negative** ($FN=7$)	**True Negative** ($TN=74$)	$NPV = \frac{74}{81} = 91\%$
		$TPR = \frac{13}{20} = 65\%$ recall	$TNR = \frac{74}{80} = 93\%$	$ACC = \frac{87}{100} = 87\%$

		Actual Class		
	Total Population	Child ($P=60$)	Not a Child ($N=40$)	
Recognized Class	Child (63)	**True Positive** ($TP=57$)	**False Positive** ($FP=6$)	precision $PPV = \frac{57}{63} = 90\%$
	Not a Child (37)	**False Negative** ($FN=3$)	**True Negative** ($TN=34$)	$NPV = \frac{34}{37} = 92\%$
		$TPR = \frac{57}{60} = 95\%$ recall	$TNR = \frac{34}{40} = 85\%$	$ACC = \frac{91}{100} = 91\%$

– Note that the accuracy for both classes “Woman” and “Child” are high and almost the same. However, it is wrong to conclude that the recognition of both classes works equally good. The reason for the good accuracy of class “Woman” is due to the large number of negative examples that are correctly dismissed. But precision (68%) and recall (65%) are much lower than for class “Child” documenting only mediocre capabilities to recognize women correctly.

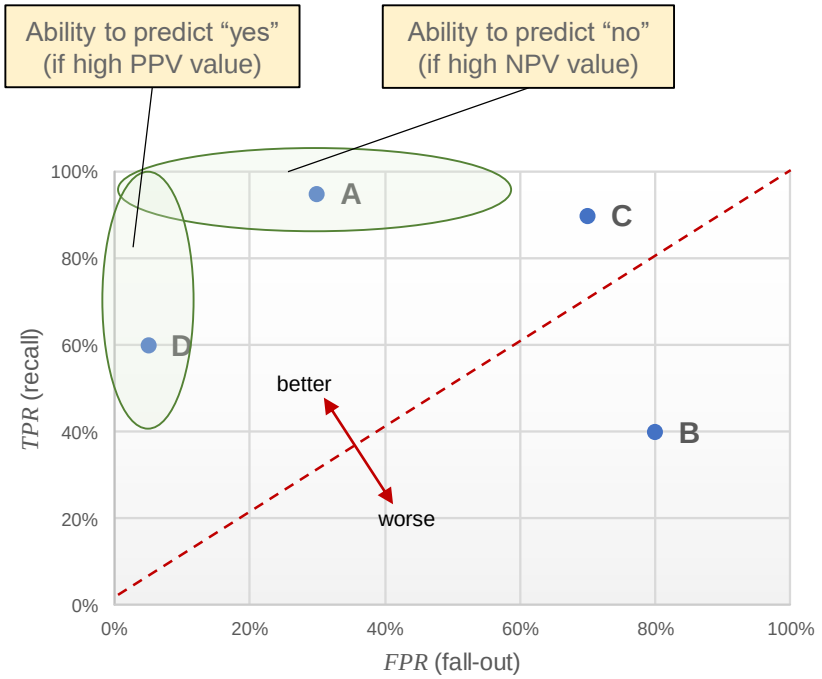
- **Binary classification with scores and thresholds** require an extension to the simple confusion matrix. Firstly, how do we calculate true/false positives/negatives if the algorithm says "Yes" only if the score exceeds a given threshold? Secondly, how do we favor algorithms that assign higher scores for positives (and lower for negatives)? The Receiver Operating Characteristic Curve (ROC Curve) is a simple tool to answer these questions.
  - The ROC curve is a 2-dimensional plot with the x-axis denoting the false positive rate (FPR) and the y-axis denoting the true positive rate (TPR). The ideal point is (0,1), i.e., the upper-left corner with accuracy (ACC), precision (PPV), and recall (TPR) at 100% and fall-out (FPR) and miss rate (FNR) at 0%. In general, the more north-west the better, the more south-east the worse becomes the performance.
  - Example without scores and thresholds:

A	
$TP = 95$	$FP = 30$
$FN = 5$	$TN = 70$
$TPR = 95\%$	$FPR = 30\%$
$PPV = 76\%$	$NPV = 93\%$
$ACC = 83\%$	

B	
$TP = 40$	$FP = 80$
$FN = 60$	$TN = 20$
$TPR = 40\%$	$FPR = 80\%$
$PPV = 33\%$	$NPV = 25\%$
$ACC = 30\%$	

C	
$TP = 90$	$FP = 70$
$FN = 10$	$TN = 30$
$TPR = 90\%$	$FPR = 70\%$
$PPV = 56\%$	$NPV = 75\%$
$ACC = 60\%$	

D	
$TP = 60$	$FP = 5$
$FN = 40$	$TN = 95$
$TPR = 60\%$	$FPR = 5\%$
$PPV = 92\%$	$NPV = 70\%$
$ACC = 78\%$	

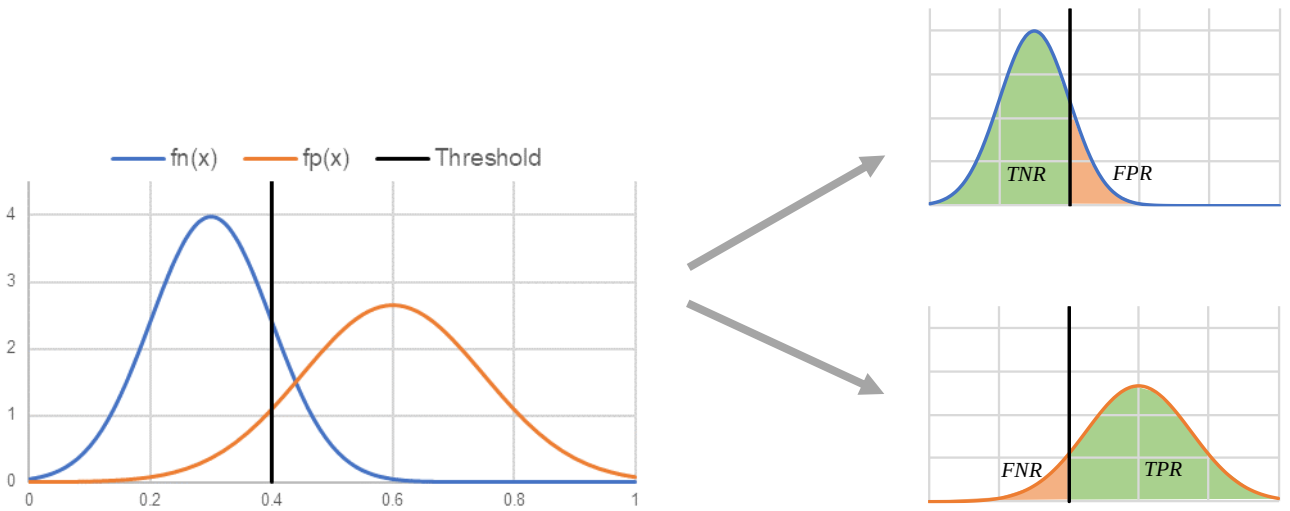


– Adding scores and threshold changes the way the algorithm decides. With binary classification, assume that the prediction is based on a random variable X which is a score for the current instance. The higher the score, the more likely it is a positive case, the lower the score the more likely it is a negative case. A threshold T is required such that the algorithms yields "Yes" if $X > T$ and "No" otherwise.
  - Let $f_p(x)$ denote the probability density of X if the instance belongs to class "positive"
  - Let $f_n(x)$ denote the probability density of X if the instance belongs to class "negative"
– We can calculate the various rates as a function of the threshold T as follows

$$TPR(T) = \int_T^{\infty} f_p(x)\,dx \qquad FNR(T) = \int_{-\infty}^{T} f_p(x)\,dx$$

$$TNR(T) = \int_{-\infty}^{T} f_n(x)\,dx \qquad FPR(T) = \int_T^{\infty} f_n(x)\,dx$$

or visually

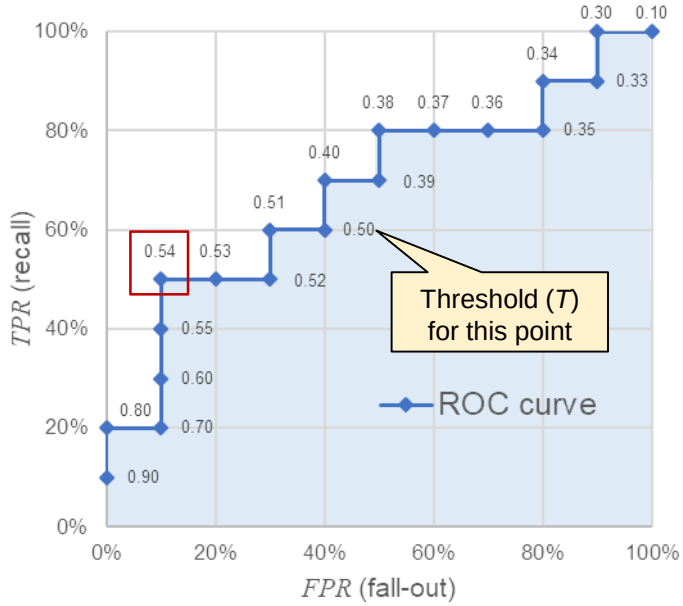


– The ROC curve serves two purposes: 1) optimize the threshold T, and 2) assess the performance of the algorithm. Let us consider the following simple example with 20 instances that have given labels P("positive") and N ("negative"). The machine learning method returns a score between 0 and 1, the higher the score the more likely it is positive (i.e., yields a "yes").

Use score of current row as threshold

Highest accuracy with T=0.54

Class	Score	TP	FP	FN	TN	TPR	FPR	ACC
P	0.90	1	0	9	10	10%	0%	55%
P	0.80	2	0	8	10	20%	0%	60%
N	0.70	2	1	8	9	20%	10%	55%
P	0.60	3	1	7	9	30%	10%	60%
P	0.55	4	1	6	9	40%	10%	65%
P	0.54	5	1	5	9	50%	10%	70%
N	0.53	5	2	5	8	50%	20%	65%
N	0.52	5	3	5	7	50%	30%	60%
P	0.51	6	3	4	7	60%	30%	65%
N	0.50	6	4	4	6	60%	40%	60%
P	0.40	7	4	3	6	70%	40%	65%
N	0.39	7	5	3	5	70%	50%	60%
P	0.38	8	5	2	5	80%	50%	65%
N	0.37	8	6	2	4	80%	60%	60%
N	0.36	8	7	2	3	80%	70%	55%
N	0.35	8	8	2	2	80%	80%	50%
P	0.34	9	8	1	2	90%	80%	55%
N	0.33	9	9	1	1	90%	90%	50%
P	0.30	10	9	0	1	100%	90%	55%
N	0.10	10	10	0	0	100%	100%	50%



– An interesting aspect is that the scores do not have to be absolutely correct, i.e., the correct probability that an instance is positive. Rather, we only require relative correctness to distinguish positive from negative cases.

– In general, higher thresholds tend to be more "conservative" (less false positive) while lower thresholds are more "liberal" (more true positives). Accuracy is only one way to select a threshold. Other values like precision, recall or fall-out can be used as well.

– Performance of an algorithm can be measured as the **area under the ROC curve** (see the blue area in the right hand figure); the bigger the area, the better the algorithm.

- **Multi-class Classification with Probabilities** measures the performance based on the probabilities on the class labels of an object. An instance x is part of the class C_k if $c_k(x) = 1$, and is not part of that class if $c_k(x) = 0$ (c_k denotes the true membership). The algorithm predicts probabilities $y_k(x)$ for an instance x with $y_k(x)$ being large if x is likely to belong to class C_k.
  - In information theory, the **cross-entropy H** measures how accurate a model distribution q matches the true distribution p over a set of events ε

$$H(p,q) = -\sum_{\varepsilon} p_\varepsilon \log q_\varepsilon$$

If we do not state otherwise log always refers to the natural logarithm. However, for our purpose, the base is irrelevant as it only scales the result but does not change order

  - The **log-loss** measure is a simplification of the cross-entropy with exactly two events: 1) x is part of class C_k, and 2) x is not part of class C_k. The true distribution p then becomes $p \in \{c_k(x), 1 - c_k(x)\}$ and the model distribution q becomes $q \in \{y_k(x), 1 - y_k(x)\}$. Thus:

$$H_{k,x}(p,q) = -\sum_{\varepsilon} p_\varepsilon \log q_\varepsilon = -c_k(x)\log\big(y_k(x)\big) - \big(1 - c_k(x)\big)\log\big(1 - y_k(x)\big)$$

  - Summing over all instances x and classes C_k, the performance is measured as

$$P = -\sum_{x}\sum_{k}\Big(c_k(x)\log\big(y_k(x)\big) + \big(1 - c_k(x)\big)\log\big(1 - y_k(x)\big)\Big)$$

  - Note: To improve the numerical stability of the log-calculations, $y_k(x)$ is often adjusted by a small value Δ (e.g., $\Delta = 10^{-15}$): $\hat{y}_k(x) = \max\Big(\Delta, \min\big(1 - \Delta, y_k(x)\big)\Big)$

- With **Regression** tasks, we measure the performance as the **mean squared error (MSE)** between the actual values and the predicted ones. Let $\boldsymbol{Y}$ be the vector of observed values with $\boldsymbol{Y} \in \mathbb{R}^N$, thus we have N samples. Let $\widehat{\boldsymbol{Y}}$ be the vector with the predicted values, again with $\widehat{\boldsymbol{Y}} \in \mathbb{R}^N$. The MSE is given as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{Y}_i - Y_i\right)^2 = \frac{1}{N}\left\|\widehat{\boldsymbol{Y}} - \boldsymbol{Y}\right\|_2^2$$

  - Regression methods model the prediction with a function f having parameters $\boldsymbol{\theta}$ to map an input vector $\boldsymbol{x}_i$ to an output value $\widehat{Y}_i$, i.e., $\widehat{Y}_i = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ with $f: \mathbb{R}^M \rightarrow \mathbb{R}$ and $\boldsymbol{\theta} \in \mathbb{R}^D$. The number D of parameters depends on the chosen function. With linear regression, $D = M$ and $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^{\top}\boldsymbol{x}$.
  - To find the best solution, a regression algorithm must find the parameters $\boldsymbol{\theta}^*$ which minimize the MSE; in other words. Let $\widehat{Y} = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$

$$\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}} \left\|\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \boldsymbol{Y}\right\|_2^2$$

Note that the factor $1/N$ does not change the solution $\boldsymbol{\theta}^*$, hence we can omit it here

  - To solve the above equation, we need find values for $\boldsymbol{\theta}$ where the gradient is $\boldsymbol{0}$:

$$\nabla_{\boldsymbol{\theta}} \left\|\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \boldsymbol{Y}\right\|_2^2 = 0$$

  - With simple regression models, we can use calculus to analytically find the exact solution. In more complex cases, a numeric solution with **gradient descent** is often sufficient even if we find only a local instead of the global minimum (approximate result). The use of squared error simplifies the gradient calculations significantly.
  - Backpropagation in neural networks use a similar method to train the weights in the network through (stoachastic) gradient descent.

1.6 Literature and Links

- David A. Grossman, Ophir Frieder, “Information Retrieval Algorithms and Heuristics.“, Kluwer Academic Publishers, 1998
- [TREC] – “Text REtrieval Conference“ http://trec.nist.gov/
- [INEX] – „Initiative for the Evaluation of XML retrieval“ http://qmir.dcs.qmw.ac.uk/INEX/
- Zou, Kelly H.; O'Malley, A. James; Mauri, Laura (2007); *Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models*, Circulation, 115(5):654–7
- Hanley, James A.; McNeil, Barbara J. (1982). "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve". *Radiology*. **143** (1): 29-36. PMID 7063747. doi:10.1148/radiology.143.1.7063747.