Performance Evaluation of Machine Learning Systems

R. Basili, S. Filice

University of Roma Tor Vergata

Deep Learning 2023/2024

Motivations

- Is a ML system performing properly?
- Among a set of different algorithms/models, which one is performing better on a given task?
- What can I do to improve my target classification system?

Overview

- Performance Evaluation Metrics
  - Classifier Evaluation Metrics
  - Information Retrieval Systems Evaluation Metrics
- Tuning and Evaluation Methods
- Error Diagnostics

Classifier Evaluation: Confusion Matrix

		PREDICTED VALUE		
ACTUAL VALUE		Class A	Class B	Class C
	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

$$accuracy = \frac{\#correct\ classifications}{\#classifications} = \frac{38 + 43 + 44}{150} = 83.33\%$$

$$error\ rate = \frac{\#incorrect\ classifications}{\#classifications} = \frac{12 + 5 + 2 + 6}{150} = 16.67\%$$

Evaluation with skewed data

- Accuracy is not a suitable metric for task with imbalanced classes (for instance a spam detector)

		PREDICTED VALUE	
ACTUAL VALUE		Spam	Non-Spam
	Spam	0	10
	Non-Spam	0	9990

Very bad performance on the Spam class, that is the target of the classifier!!

… nonetheless …

$$accuracy = \frac{\#correct\ classifications}{\#classifications} = \frac{9990}{10000} = 99.9\%$$

Single Class Metrics

		PREDICTED VALUE	
		Class C	**Not Class C**
ACTUAL VALUE	**Class C**	**TP** True Positive	**FN** False Negative
	Not Class C	**FP** False Positive	**TN** True Negative

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

what percentage of instances the classifier labeled as positive are actually positive?

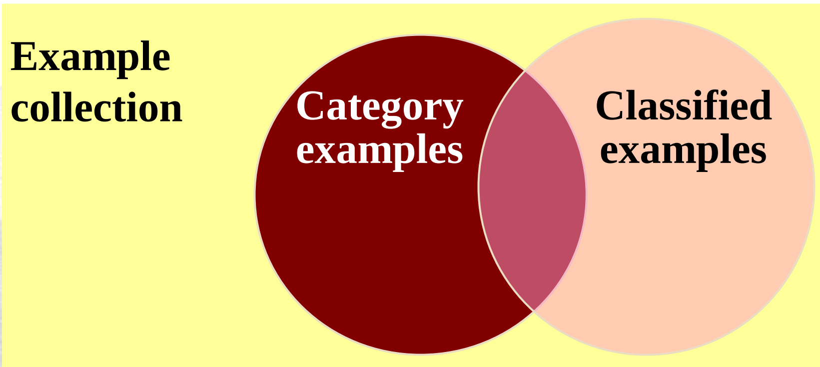
$$recall = \frac{TP}{TP + FN}$$

what percentage of positive instances did the classifier label as positive?

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

F-measure is the harmonic mean of precision and recall

Class-based evaluation



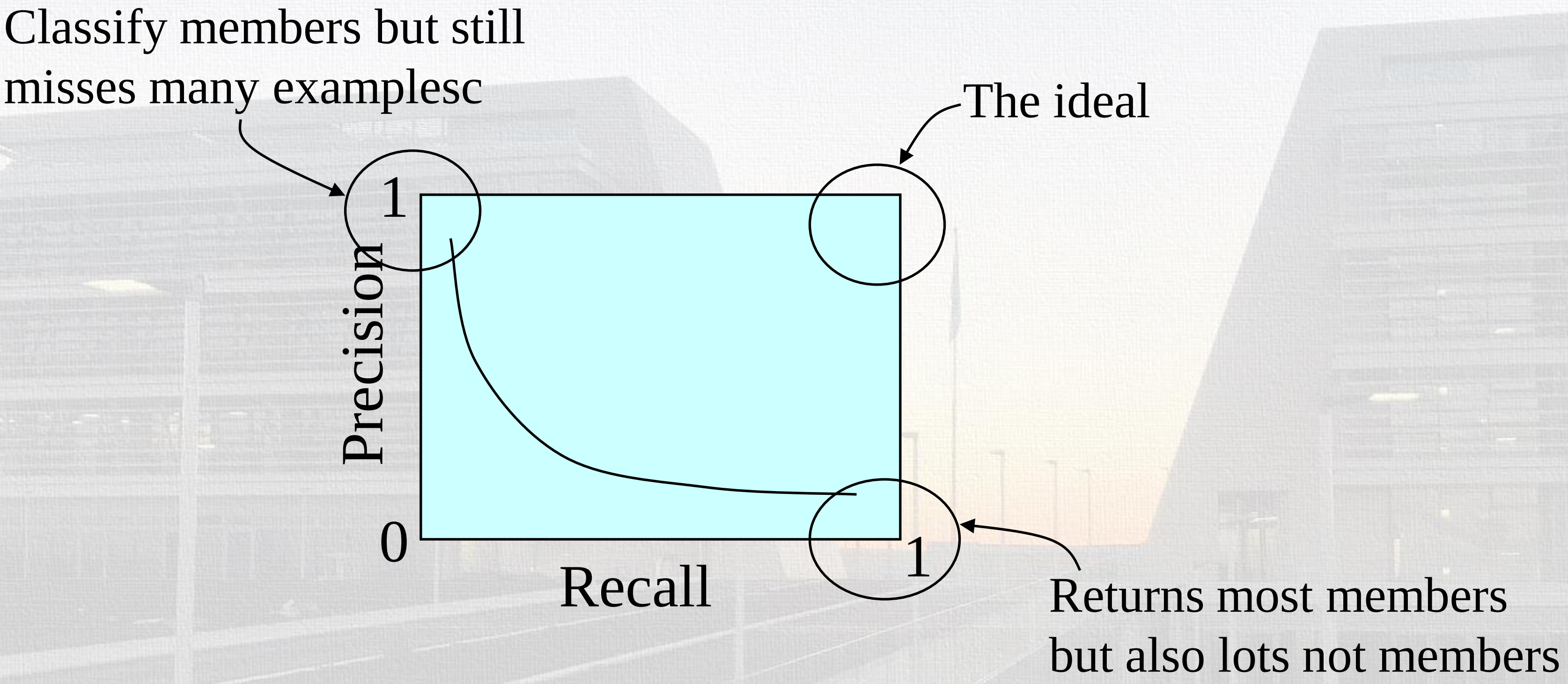
	Classified	Rejected
Not Members	Classified but not Members	Rejected & not Members
Members	Classified & Members	Rejected but Members

$$precision = \frac{\#\ of\ Members\ Classified}{\#\ of\ Members\ Classified\ + \#\ of\ Classified\ not\ Members}$$

$$recall = \frac{\#\ of\ Members\ Classified}{\#\ of\ Members\ Classified\ + \#\ of\ Rejected\ Members}$$

What about accuracy???

Trade-off between Precision and Recall



Other class based measures

Precision and Recall of C_i

- a_i, corrects (TP_i)
- b_i, mistakes (FP_i)
- c_i, instances of a $Class_i$ that are not actually retrieved, (FN_i)

The *Precision* and *Recall* are defined by the above counts:

$$Precision_i = \frac{a_i}{a_i + b_i}$$

$$Recall_i = \frac{a_i}{a_i + c_i}$$

		PREDICTED VALUE		
		Class A	Class B	Class C
ACTUAL VALUE	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

- $\text{Precision}_A = 38/(38+5+6) = 38/49$
- $\text{Recall}_A = 38/(38+12) = 38/50$

- $\text{Precision}_B = 43/(43+12) = 43/55$
- $\text{Recall}_C = 44/(44+6) = 44/50$

Performance Measurements (cont'd)

- Breakeven Point
  - Find thresholds for which

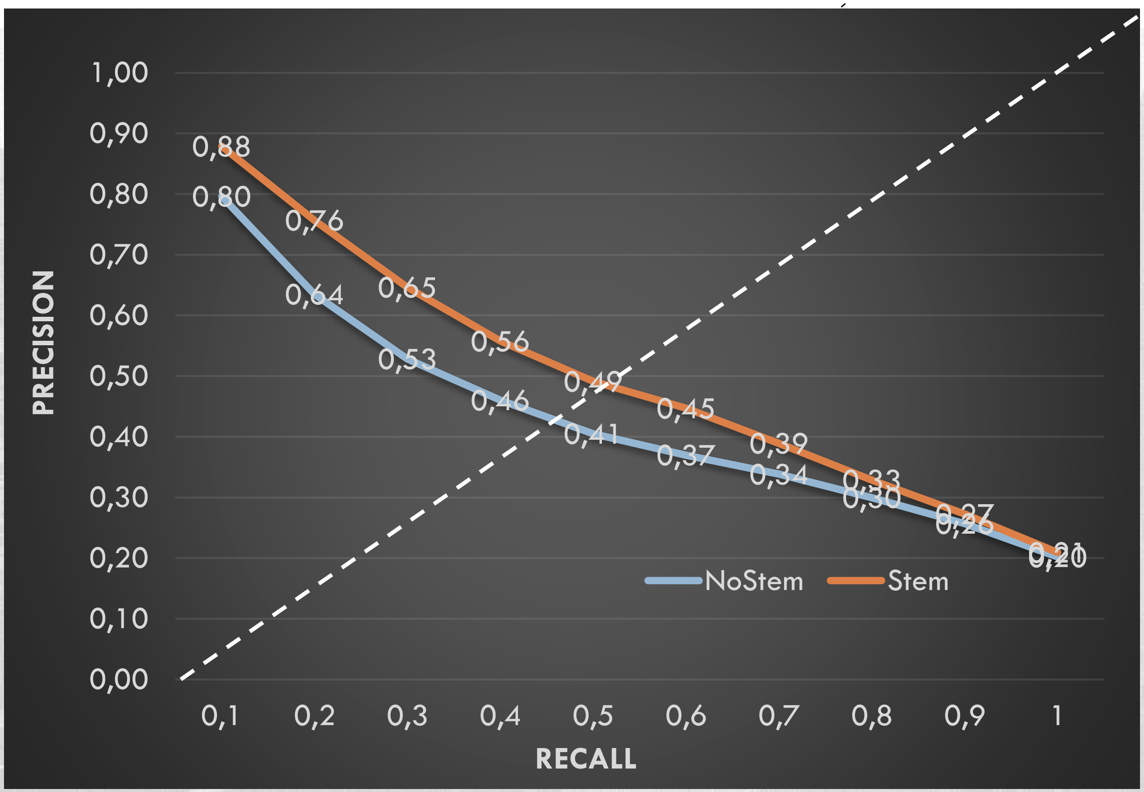
    Recall = Precision
  - Interpolation
- F-measure

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

  - Harmonic mean between precision and recall
- Global performance on more than two categories
  - Micro-average
    - The counts refer to classifiers
  - Macro-average (average measures over all categories)

Break-even Point

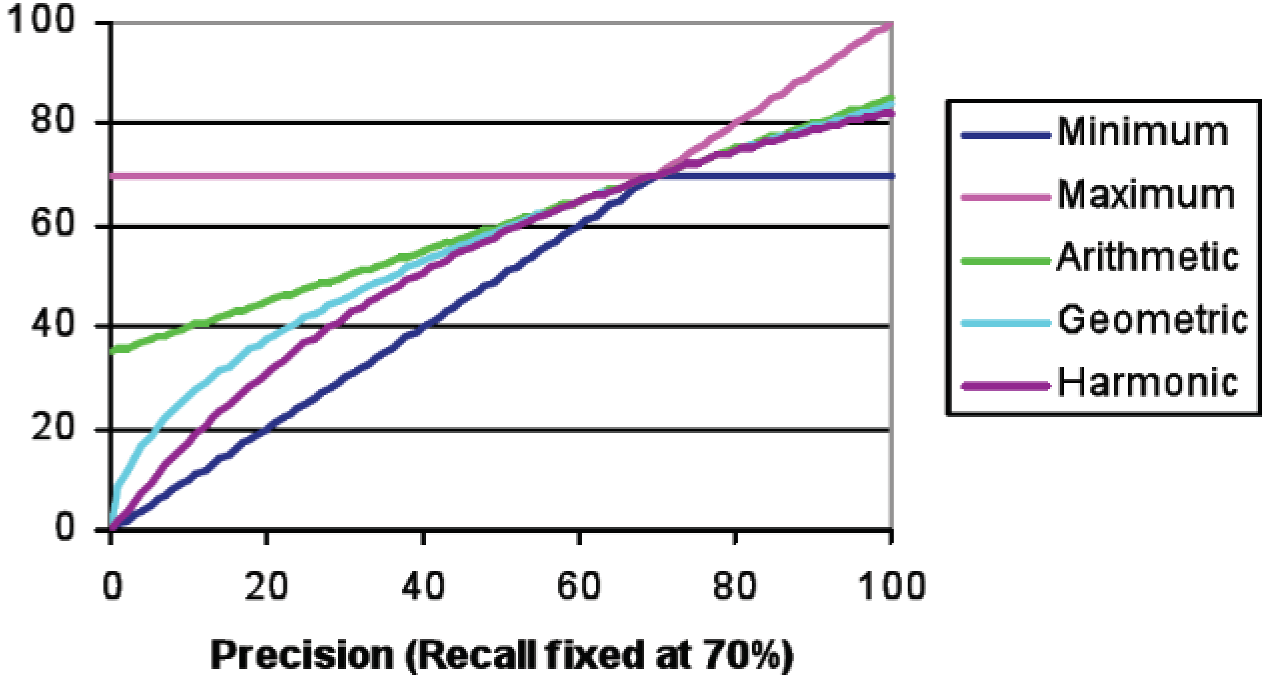
- The BEP is the interpolated estimate of the value for which Recall=Precision



- It shows the superiority of methods whose behavior is closer to the (1,1) ideal performance

Averaging Precision & Recall:
A comparison

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$



$$\min(p,r)$$
$$\max(p,r)$$

$$arithM(p,r) = \frac{p+r}{2}$$

$$geomM(p,r) = \sqrt{p \cdot r}$$

$$harmM(p,r) = \frac{2}{p^{-1} + r^{-1}}$$

Averaging Precision & Recall:
cross-categorical analysis

- Individual scores characterize the performance about each specific class
- Simple macro averaging can be applied to have

$$MPrecision = \sum_{i=1}^{n} Precision_i$$

$$MRecall = \sum_{i=1}^{n} Recall_i$$

$$MF_1 = \frac{2 \cdot MPrecision \cdot MRecall}{MPrecision + MRecall}$$

F-measure e MicroAverages

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$\mu Precision = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + b_i}$$

$$\mu Recall = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + c_i}$$

$$\mu BEP = \frac{\mu Precision + \mu Recall}{2}$$

$$\mu f_1 = \frac{2 \times \mu Precision \times \mu Recall}{\mu Precision + \mu Recall}$$

		PREDICTED VALUE		
		Class A	Class B	Class C
ACTUAL VALUE	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

- $Precision_A = 38/(38+5+6) = 38/49$
- $Precision_B = 43/(43+12) = 43/55$
- Segue che:

$$Mprecision = 1/3(38/49 + 43/55 + \ldots)$$

		PREDICTED VALUE		
ACTUAL VALUE		Class A	Class B	Class C
	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

- $Precision_A = 38/(38+5+6) = 38/49$
- $Precision_B = 43/(43+12) = 43/55$
- Segue che:

$\mu Precision = (38+43+44)/(38+43+44+11+12+2)$

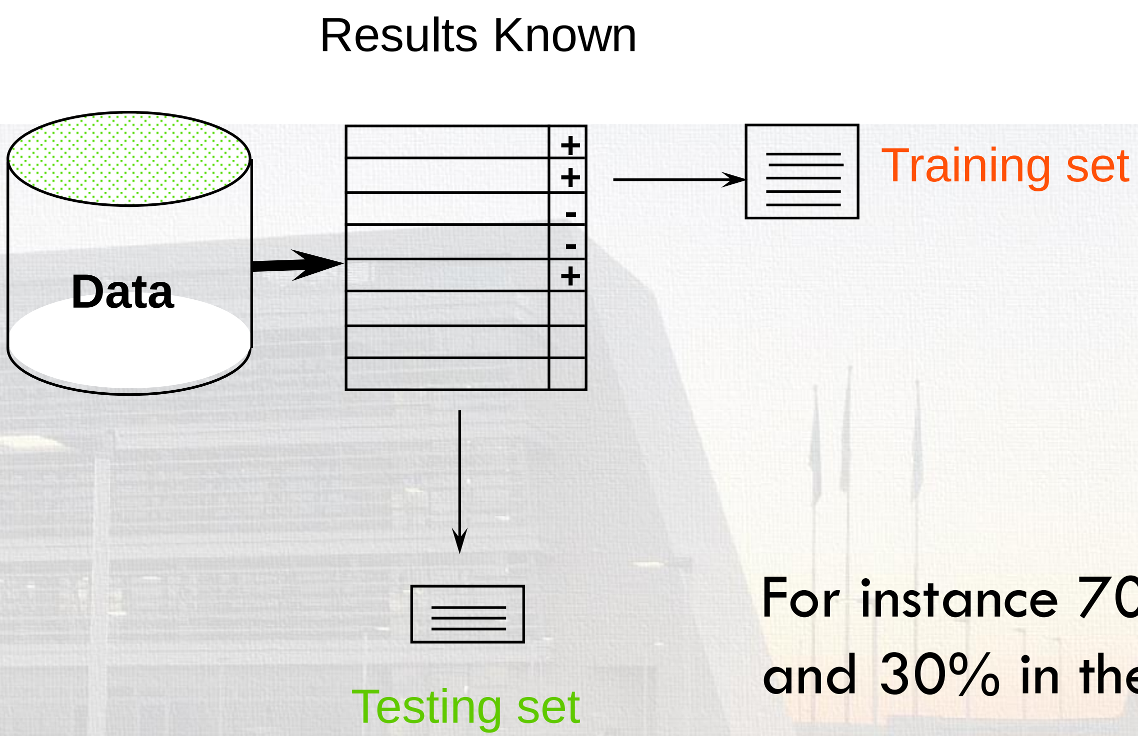
Overview

- Performance Evaluation Metrics
  - Classifier Evaluation Metrics
  - Information Retrieval Systems Evaluation Metrics
- **Tuning and Evaluation Methods**
- Error Diagnostics

Testing Data

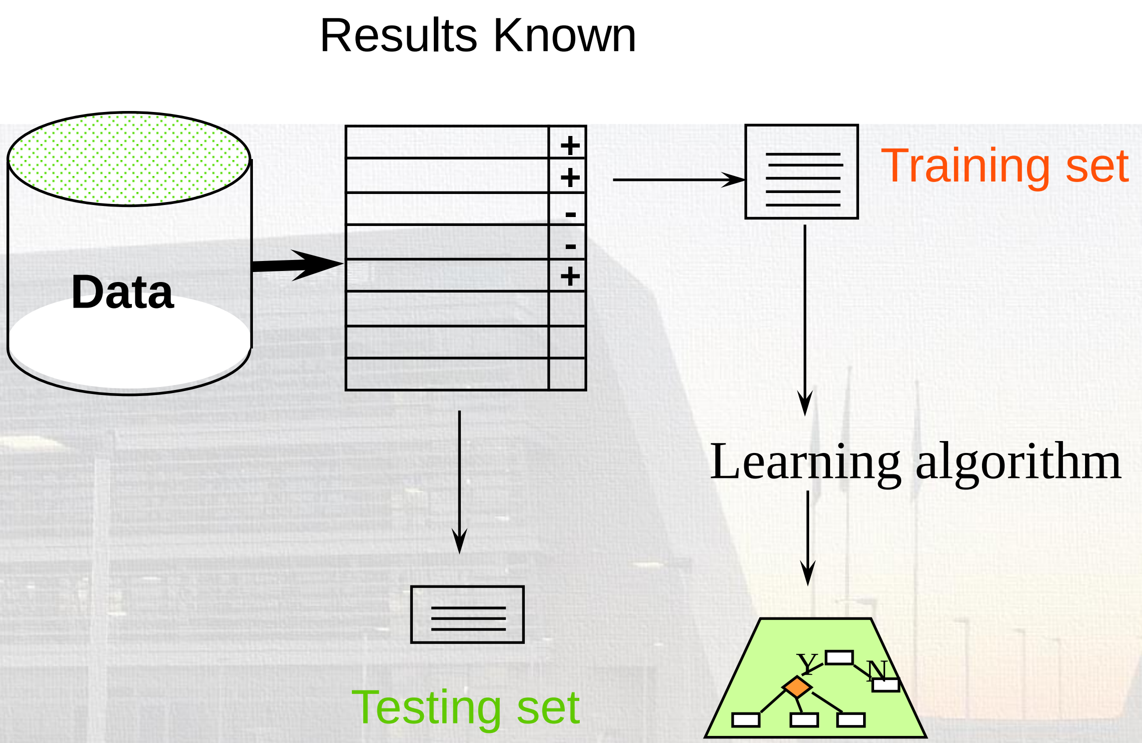
- To obtain a reliable estimation, test data **must be instances NOT employed for the training** step:
  - Error on the training data is *not* a good indicator of performance on future data, because new data will probably not be **exactly** the same as the training data!
  - *Overfitting* – fitting the training data too precisely - usually leads to poor results on new data
  - We want to evaluate how much accurate predictions of the model we learned are, and not other computational aspects (e.g. its memorization capability)

Step 1: dataset splitting

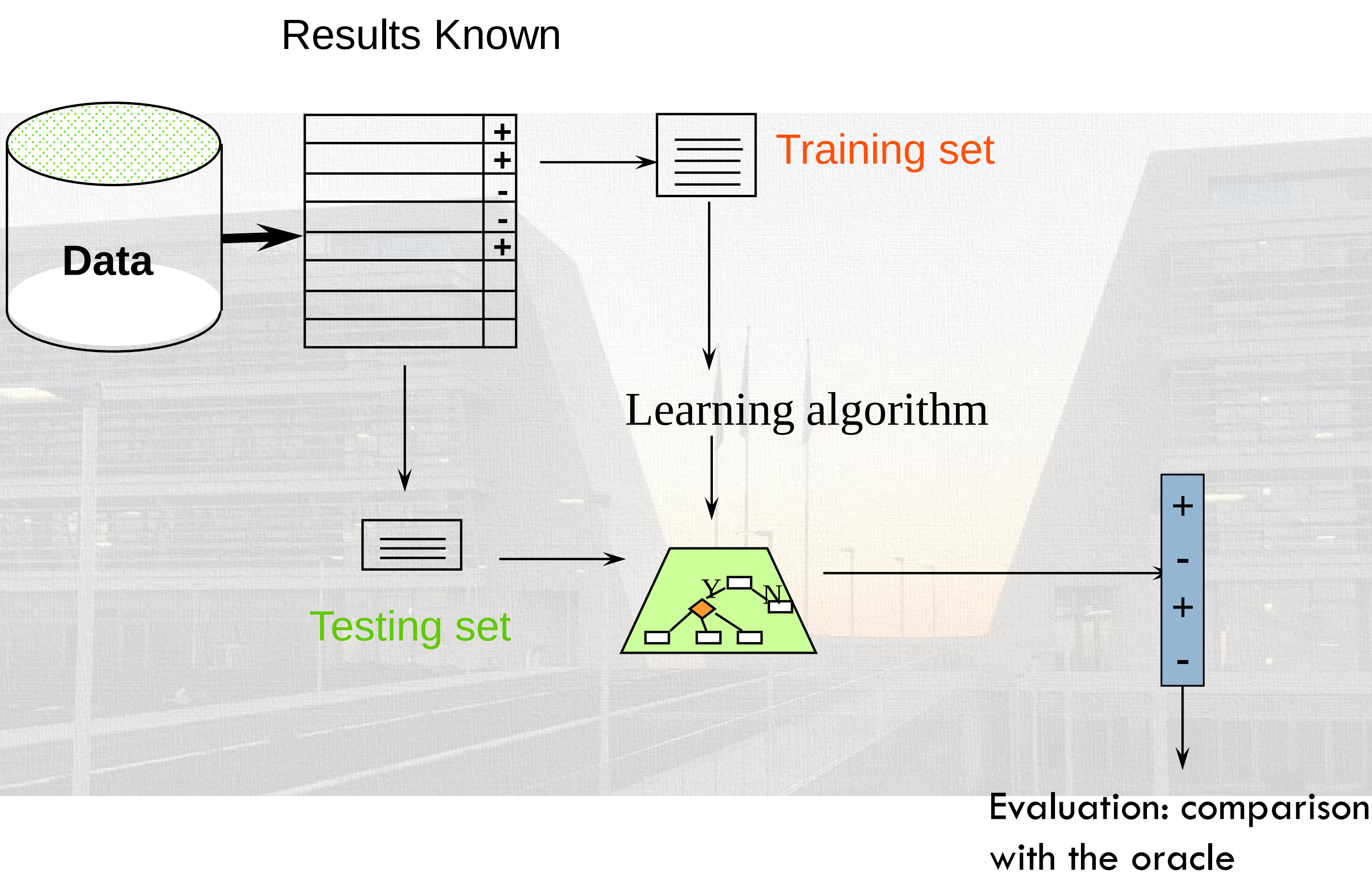


For instance 70% in the training set and 30% in the test set

Step 2: learning phase



Step 3: testing the model

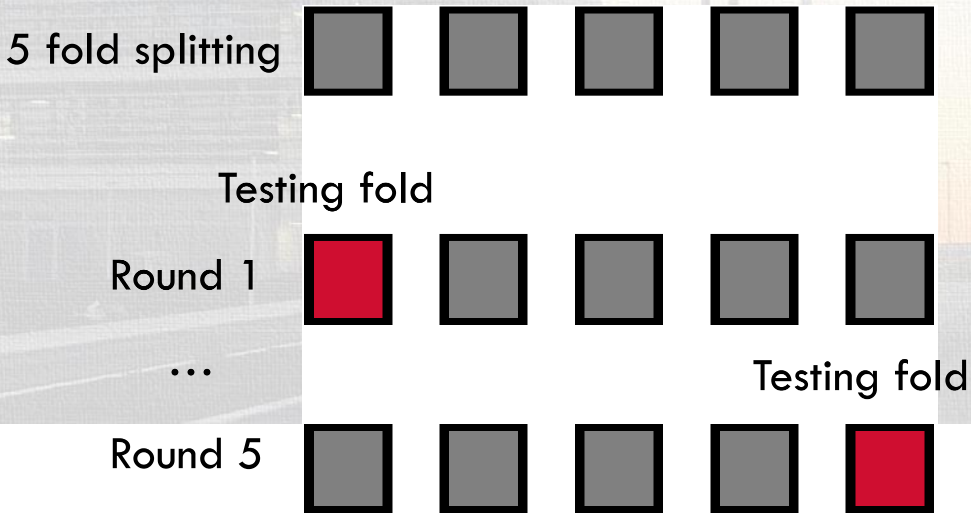


Evaluation on Few Data

- When data is scarce (totally or for a single class), a single evaluation process could not be enough representative
  - The testing set could contain too few instances to produce a reliable result
- SAMPLING: The evaluation process must be repeated with different splitting

N-Fold Cross Validation

- Data is split into *n* subsets of equal size
- Each subset in turn is used for testing and the remainders *n-1* for training
- The metrics estimated in each round are averaged



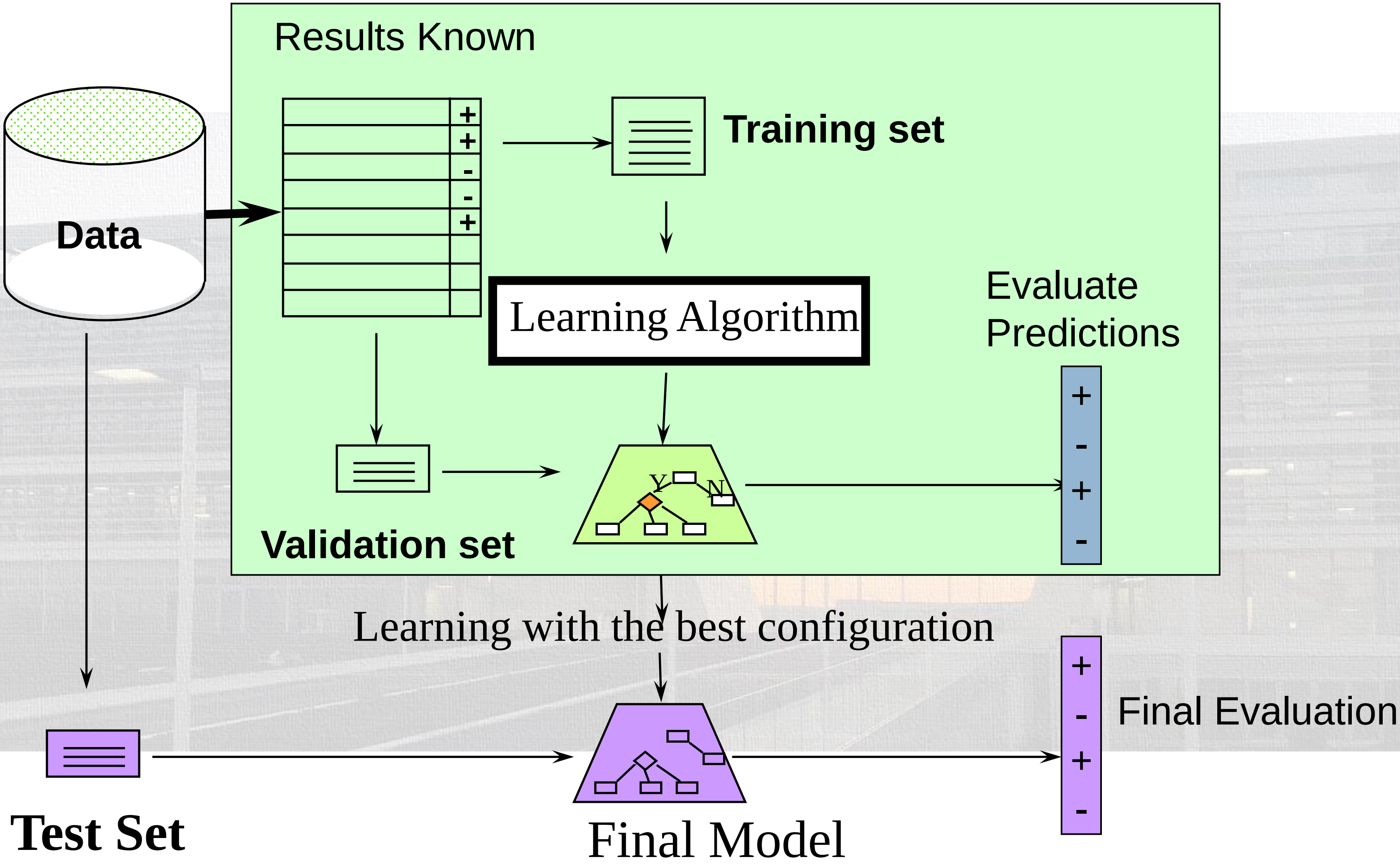
An example: Learning without learning.

LAZY LEARNING

Tuning a Classifier

- Most of ML algorithms depends on some parameters
  - Examples: k in KNN, w_i in Rocchio, $p(w_i \mid c_j)$ for NB
- The best configuration must be choosen after a proper tuning stage:
  - A set of configurations must be established (for instance, $k=1,2,5,10,\ldots,50$)
  - Each configuration must be evaluated on a validation (or tuning) set

Complete ML Process



Reuters text classification

- An example: the Reuters news text classification use case
  - Some well known classifiers (e.g. k-NN or SVM) are compared with a parametrized version of Rocchio
  - In the next slides, the parametrization procedure is presented and its evaluation is discussed

Feature Selection in Parametrized Rocchio

(Basili et al., IJCAI 2001)

- Literature work uses a bunch of values for β and γ
- Interpretation of positive (β) vs. negative (γ) information

  ⇒ value of β > γ > 0 (e.g. 16, 4)

- IJAIT interpretation: Parametrized Rocchio [IJAIT 2002, ECIR 2003]:

  Remove one parameter s (i.e. β) and let the remaining parameter to depend on the *i*-th class C^i

$$C_f^i = \max\left\{0, \frac{1}{|T_i|}\sum_{d\in T_i} d_f - \frac{\rho_i}{|\bar{T}_i|}\sum_{d\in \bar{T}_i} d_f\right\}$$

- C_f^i expresses the weight that a feature *f* brings in favour of the class *i*
- 0-weighted features *f* do not affect similarity estimation
- increasing ρ causes many feature to be set to 0 ⇒ they are removed
- Different values ρ_i of the parameter are used for different classes C^i

Experiments

- Reuters Collection 21578 Apté split (Apté94)
  - 90 classes (12,902 docs)
  - A fixed splitting between training and test set
  - 9603 vs 3299 documents
- Tokens
  - about 30,000 different
- Other different versions have been used but …

  most of TC results relate to the 21578 Apté
  - [Joachims 1998], [Lam and Ho 1998], [Dumais et al. 1998], [Li Yamanishi 1999], [Weiss et al. 1999],

    [Cohen and Singer 1999]…

A Reuters document- Acquisition Category

CRA SOLD FORREST GOLD FOR 76 MLN DLRS - WHIM CREEK

SYDNEY, April 8 - <Whim Creek Consolidated NL> said the consortium it is leading will pay 76.55 mln dlrs for the acquisition of CRA Ltd's <CRAA.S> <Forrest Gold Pty Ltd> unit, reported yesterday.

CRA and Whim Creek did not disclose the price yesterday. Whim Creek will hold 44 pct of the consortium, while <Austwhim Resources NL> will hold 27 pct and <Croesus Mining NL> 29 pct, it said in a statement.

As reported, Forrest Gold owns two mines in Western

Australia producing a combined 37,000 ounces of gold a year. It also owns an undeveloped gold project.

A Reuters document- Crude-Oil Category

FTC URGES VETO OF GEORGIA GASOLINE STATION BILL

WASHINGTON, March 20 - The Federal Trade Commission said its staff has urged the governor of Georgia to veto a bill that would prohibit petroleum refiners from owning and operating retail gasoline stations.

The proposed legislation is aimed at preventing large oil refiners and marketers from using predatory or monopolistic practices against franchised dealers.

But the FTC said fears of refiner-owned stations as part of a scheme of predatory or monopolistic practices are unfounded. It called the bill anticompetitive and warned that it would force higher gasoline prices for Georgia motorists.

Precision and Recall of C_i

- a_i, corrects
- b_i, mistakes
- c_i, not retrieved

The *Precision* and *Recall* are defined by the above counts:

$$Precision_i = \frac{a_i}{a_i + b_i}$$

$$Recall_i = \frac{a_i}{a_i + c_i}$$

F-measure e MicroAverages

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

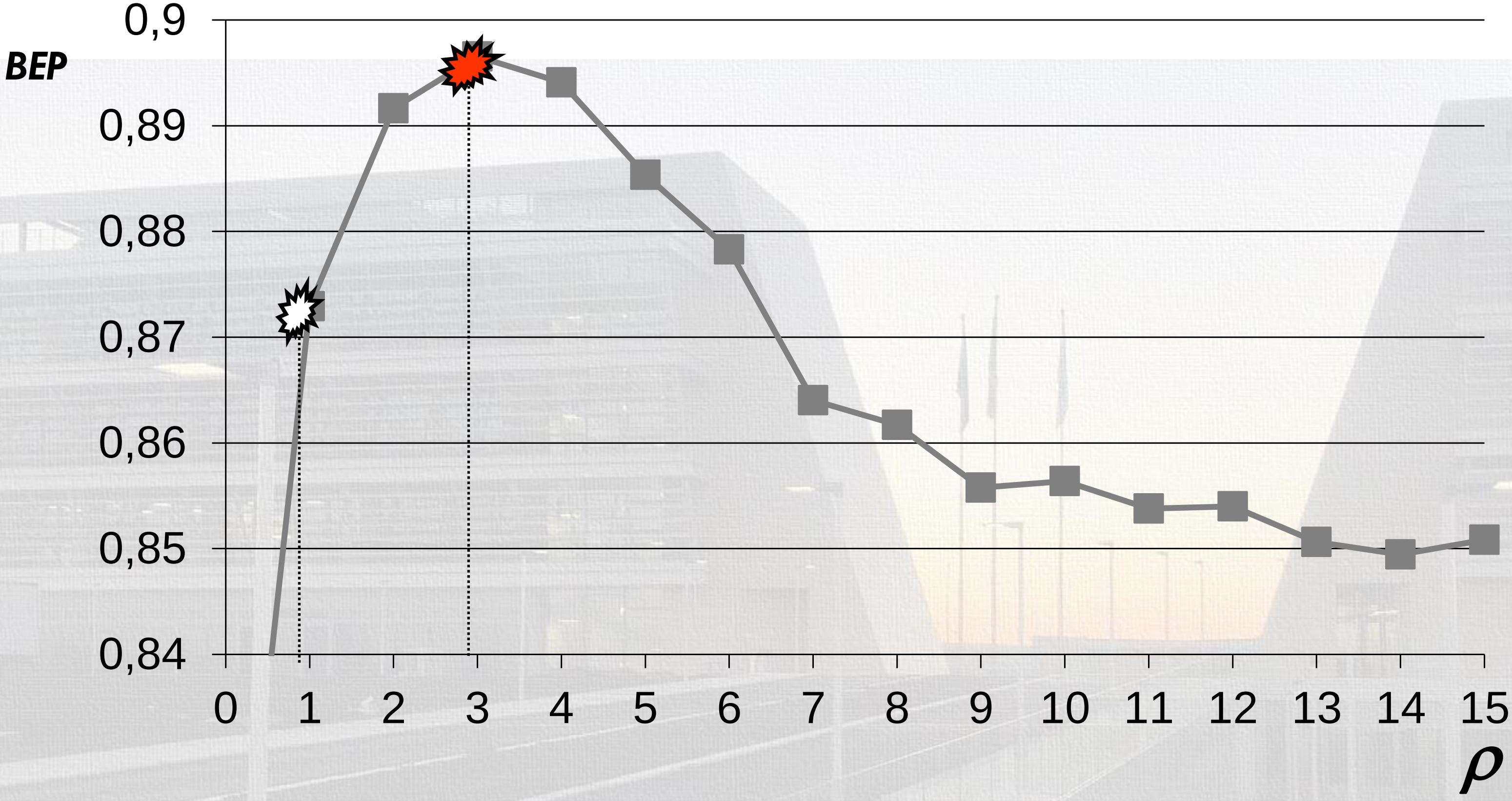
$$\mu Precision = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + b_i}$$

$$\mu Recall = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + c_i}$$

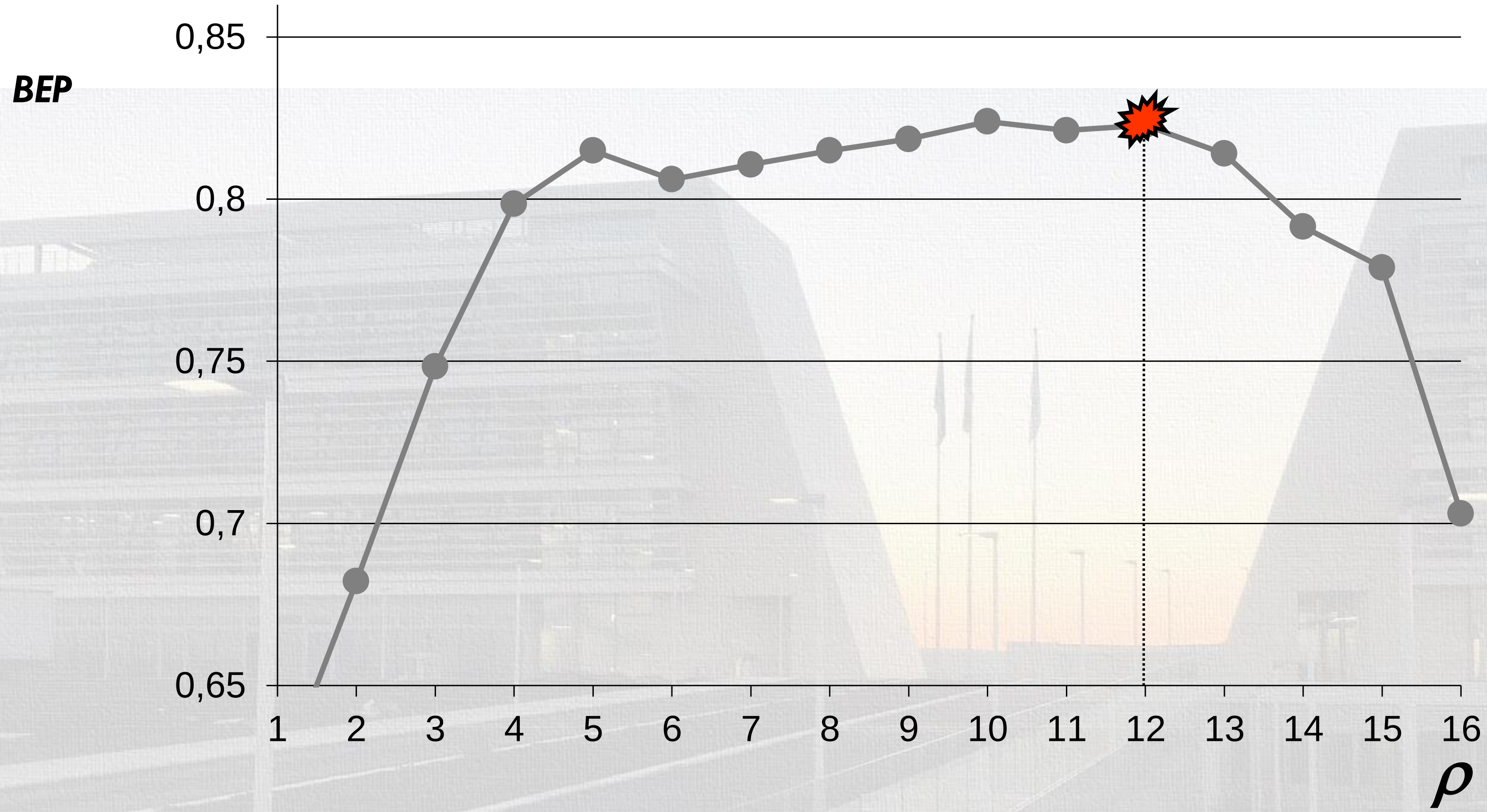
$$\mu BEP = \frac{\mu Precision + \mu Recall}{2}$$

$$\mu f_1 = \frac{2 \times \mu Precision \times \mu Recall}{\mu Precision + \mu Recall}$$

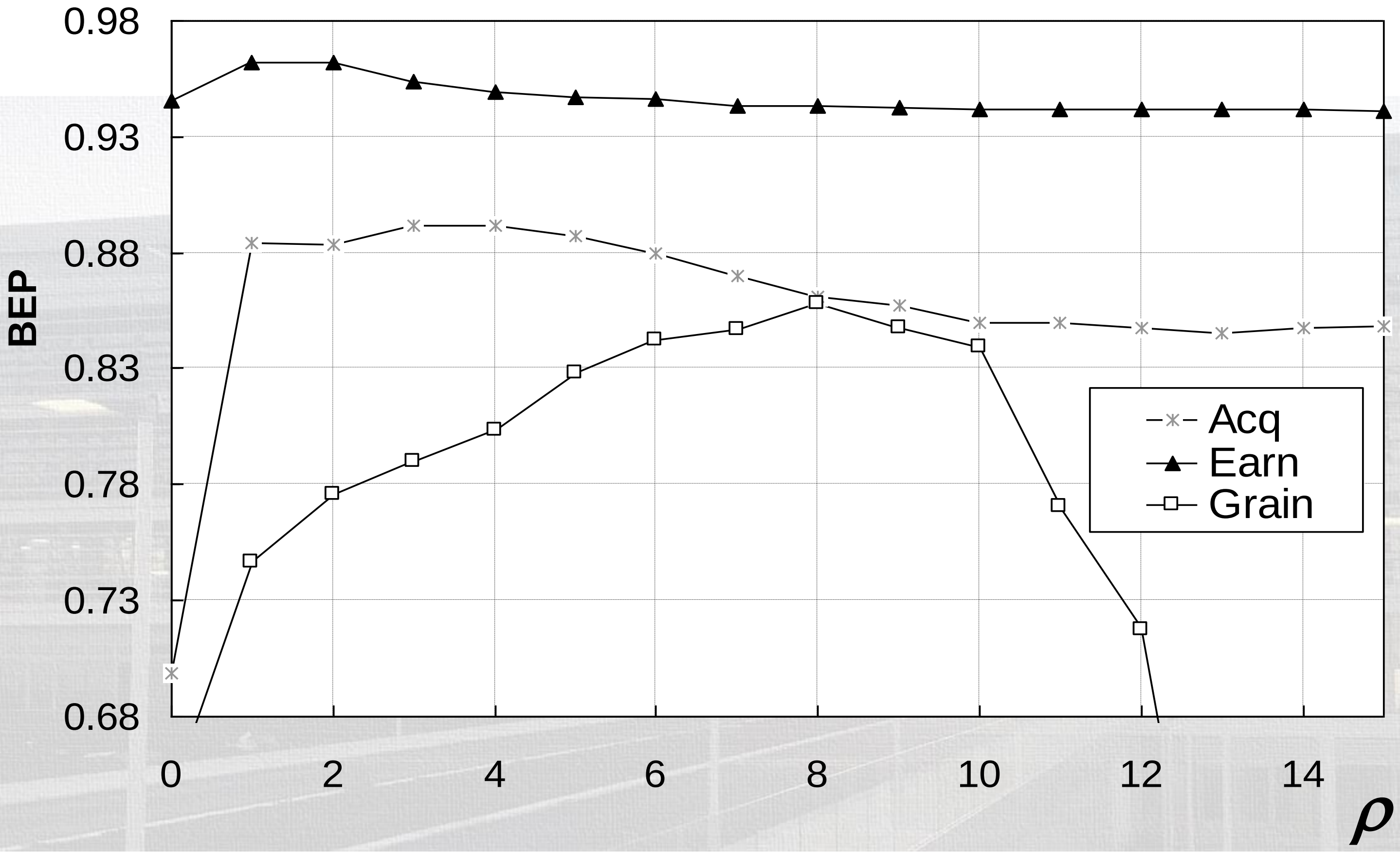
The Impact of ρ parameter on Acquisition category



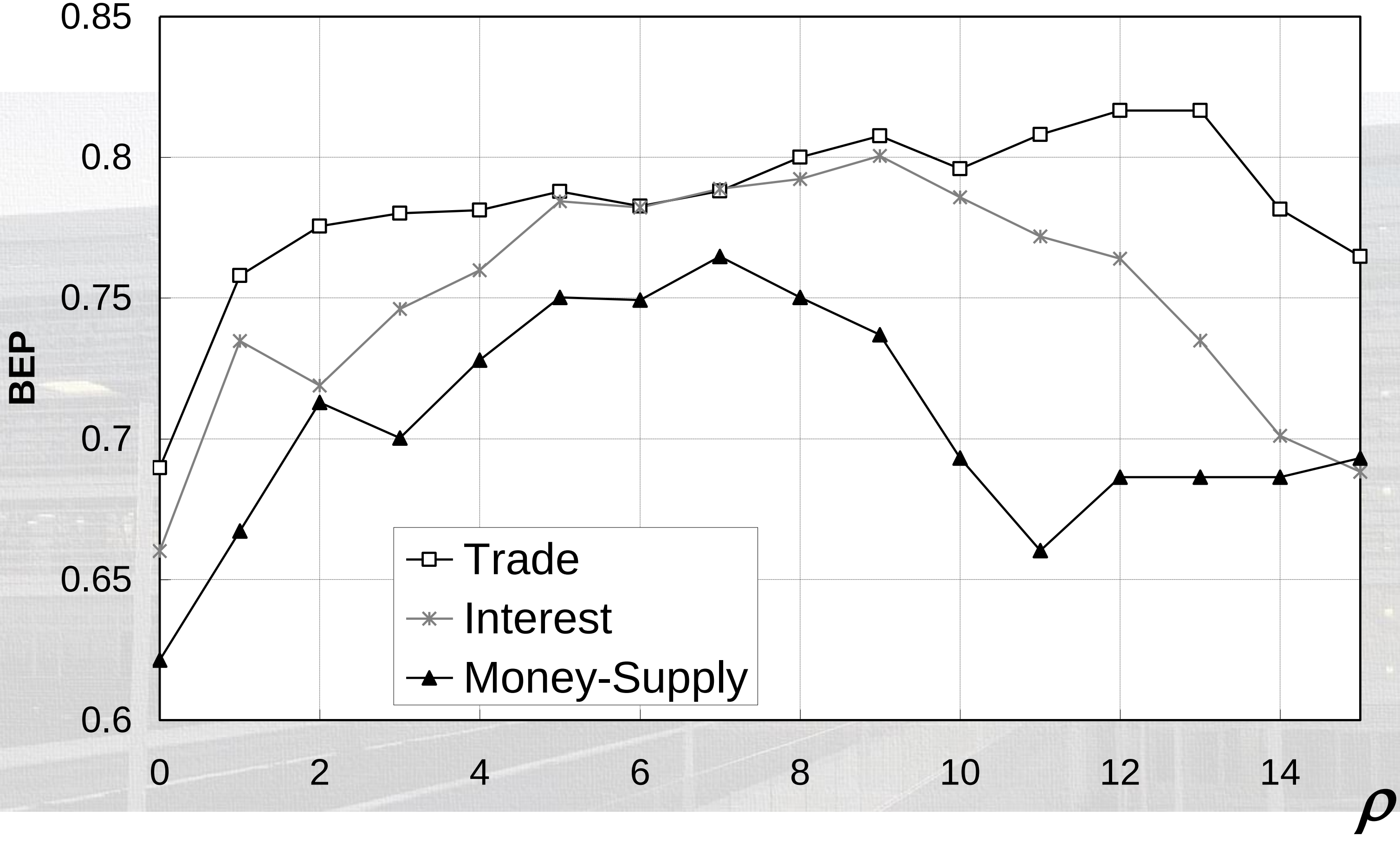
The impact of ρ parameter on Trade category



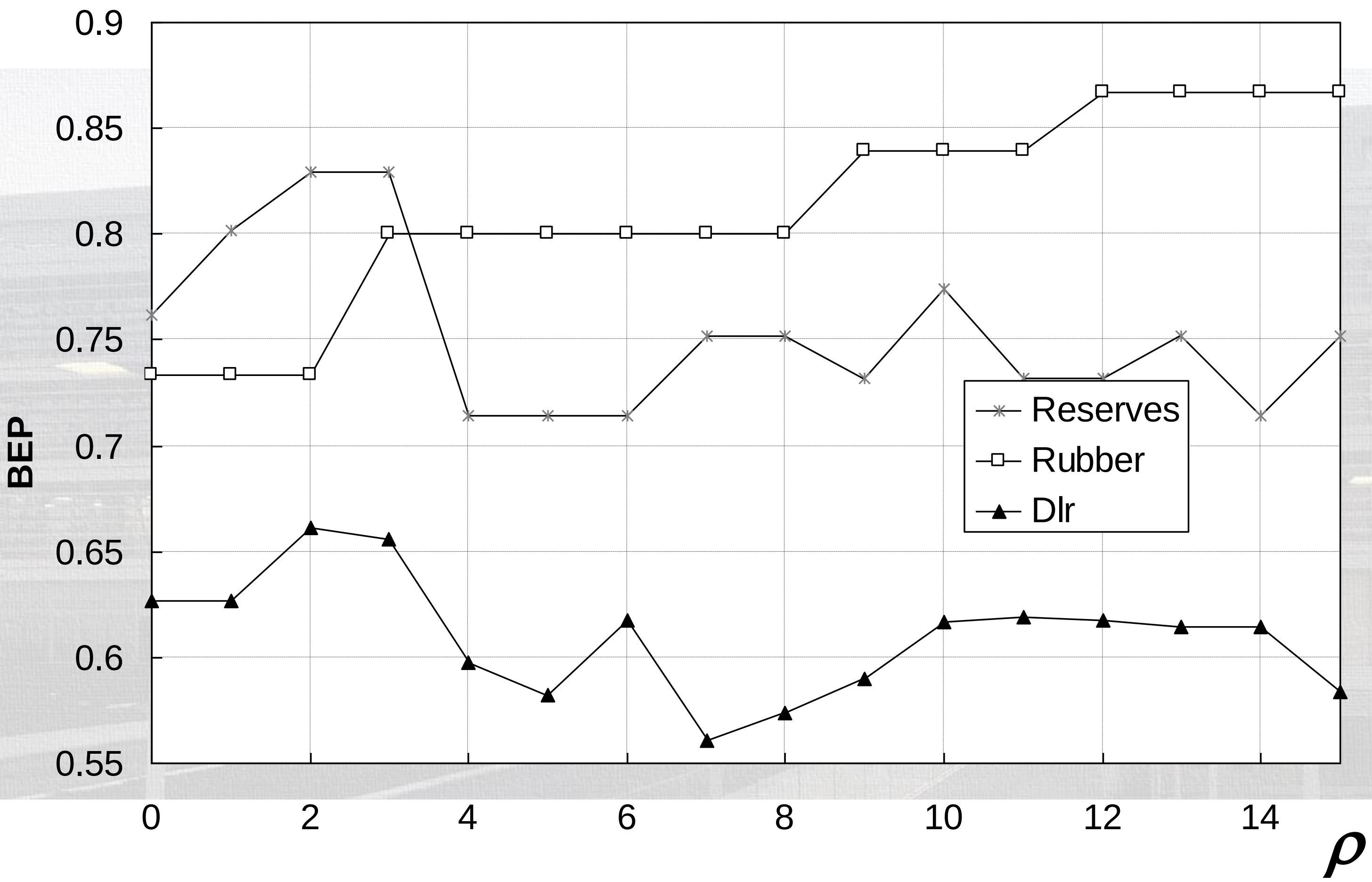
Mostly populated categories



Medium sized categories



Low size categories



Parameter Estimation Procedure

- Validation-set of about 30% of the training corpus
- for all $\rho \in [0,30]$
  - TRAIN the system on the remaining material
  - Measure the BEP on the validation-set
- Pick-up the ρ associated to the highest *BEP*
- re-TRAIN the system on the entire training-set
- TEST the system based on the obtained parameterized model
- For more reliable results:
  - 20 cross fold validation: 20 validation-sets and ρ as the average
- The Parameterized Rocchio Classifier will refer to as PRC

Comparative Analysis

- Rocchio literature parameterization
  - $\rho = 1$ ($\gamma = \beta = 1$) *and* $\rho = 1/4$ ($\gamma = 4$, $\beta = 16$)
- Reuters fixed test-set
  - Other literature results
- SVM
  - To better collocate our results
- Cross Validation (20 samples)
  - More reliable results
- Cross corpora/language validation
  - Reuters, Ohsumed (English) and ANSA (Italian)

Results on Reuters fixed split

Feature Set (~30.000)	PRC	Std Rocchio ($\gamma = \frac{1}{4}\beta$ or $\gamma = \beta$)	SVM
Tokens	82.83 %	72.71%-78.79%	85.34 %
Literature (stems)	-	75 % - 79.9%	84.2 %

- Rocchio literature results (Yang 99', Choen 98', Joachims98')
- SVM literature results (Joachims 98')

Breakeven points of widely known classifiers on the Reuters dataset

SVM	**PRC**	KNN	RIPPER	CLASSI*	Dtree
85.34%	82.83%	82.3%	82%	80.2%	79.4%

SWAP1*	CHARADE*	EXPERT	Rocchio	Naive Bayes
80.5%	78.3%	82.7%	72%-79.5%	75 % -

79.9%

* Evaluation on different Reuters versions

Cross-Validation

1. Generate n random splits of the corpus. For each split j, 70% of data can be used for training (LS^j) and 30% for testing (TS^j).

2. For each split j
   (a) Generate m validation sets, ES^j_k of about 10/30% of LS^j.
   (b) Learn the classifiers on $LS^j - ES^j{}_k$ and for each $ES^j{}_k$ evaluate: (i) the threshold associated to the BEP and (ii) the optimal parameter ρ.
   (c) Learn the classifiers Rocchio, $SVMs$ and PRC on LS^j: in case of PRC use the estimated $\bar{\rho}$.
   (d) Evaluate f_1 on TS_j (use the estimated thresholds for Rocchio and PRC) for each category and account data for the final processing of the global μf_1.

3. For each classifier evaluate the mean and the Standard Deviation for f_1 and μf_1 over the TS_j sets.

Cross-Validation on Reuters (20 samples)

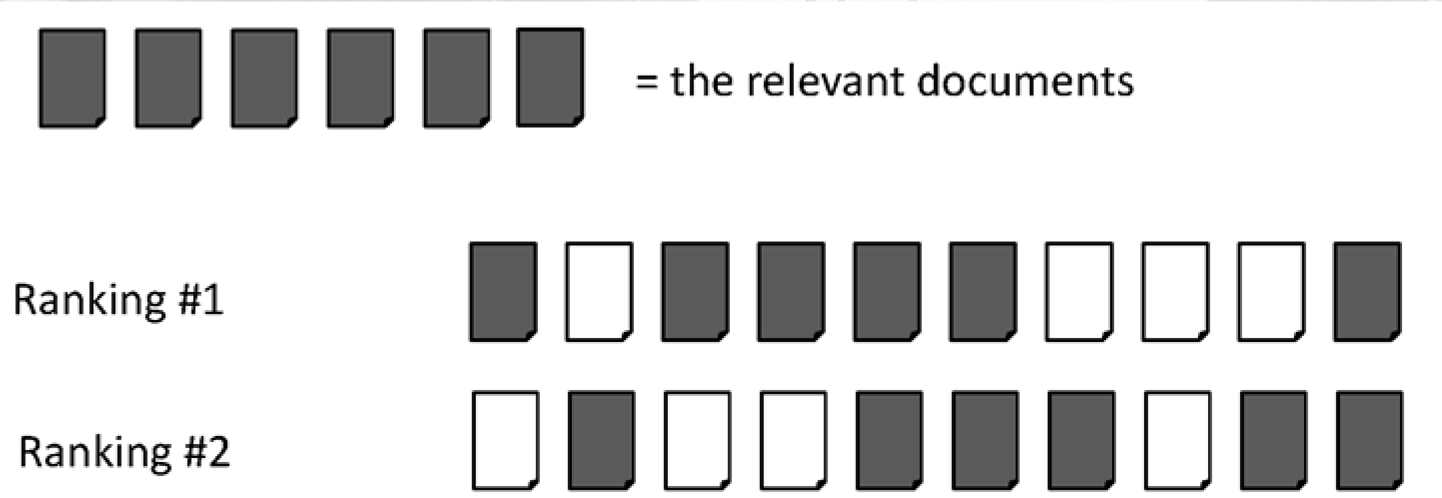
	Rocchio				PRC		SVM	
	RTS		TS^{σ}		RTS	TS^{σ}	RTS	TS^{σ}
	ρ=.25	ρ=1	ρ=.25	ρ=1				
earn	95.69	95.61	92.57±0.51	93.71±0.42	95.31	94.01±0.33	98.29	97.70±0.31
acq	59.85	82.71	60.02±1.22	77.69±1.15	85.95	83.92±1.01	95.10	94.14±0.57
money-fx	53.74	57.76	67.38±2.84	71.60±2.78	62.31	77.65±2.72	75.96	84.68±2.42
grain	73.64	80.69	70.76±2.05	77.54±1.61	89.12	91.46±1.26	92.47	93.43±1.38
crude	73.58	80.45	75.91±2.54	81.56±1.97	81.54	81.18±2.20	87.09	86.77±1.65
trade	53.00	69.26	61.41±3.21	71.76±2.73	80.33	79.61±2.28	80.18	80.57±1.90
interest	51.02	58.25	59.12±3.44	64.05±3.81	70.22	69.02±3.40	71.82	75.74±2.27
ship	69.86	84.04	65.93±4.69	75.33±4.41	86.77	81.86±2.95	84.15	85.97±2.83
wheat	70.23	74.48	76.13±3.53	78.93±3.00	84.29	89.19±1.98	84.44	87.61±2.39
corn	64.81	66.12	66.04±4.80	68.21±4.82	89.91	88.32±2.39	89.53	85.73±3.79
MicroAvg. 90 cat.	72.61	78.79	73.87±0.51	78.92±0.47	82.83	83.51±0.44	85.42	87.64±0.55

Overview

- **Performance Evaluation Metrics**
  - Classifier Evaluation Metrics
  - Information Retrieval System Evaluation Metrics
- Tuning and Evaluation Methods
- Error Diagnostics

Evaluating ranked results

- IR systems usually outputs the retrieved documents in a ranked list
  - A proper evaluating should mainly consider elements in the top of the list



Recall/Precision Points

- Compute a recall/precision pair for each position in the ranked list that contains a relevant document.

n	doc #	relevant
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

Total number of relevant docs = 6
Check each new recall point:

R=1/6=0.167; P=1/1=1

R=2/6=0.333; P=2/2=1

R=3/6=0.5; P=3/4=0.75

R=4/6=0.667; P=4/6=0.667

R=5/6=0.833; P=5/13=0.38

Missing one relevant document
Never reach 100% recall

Averaging over Queries

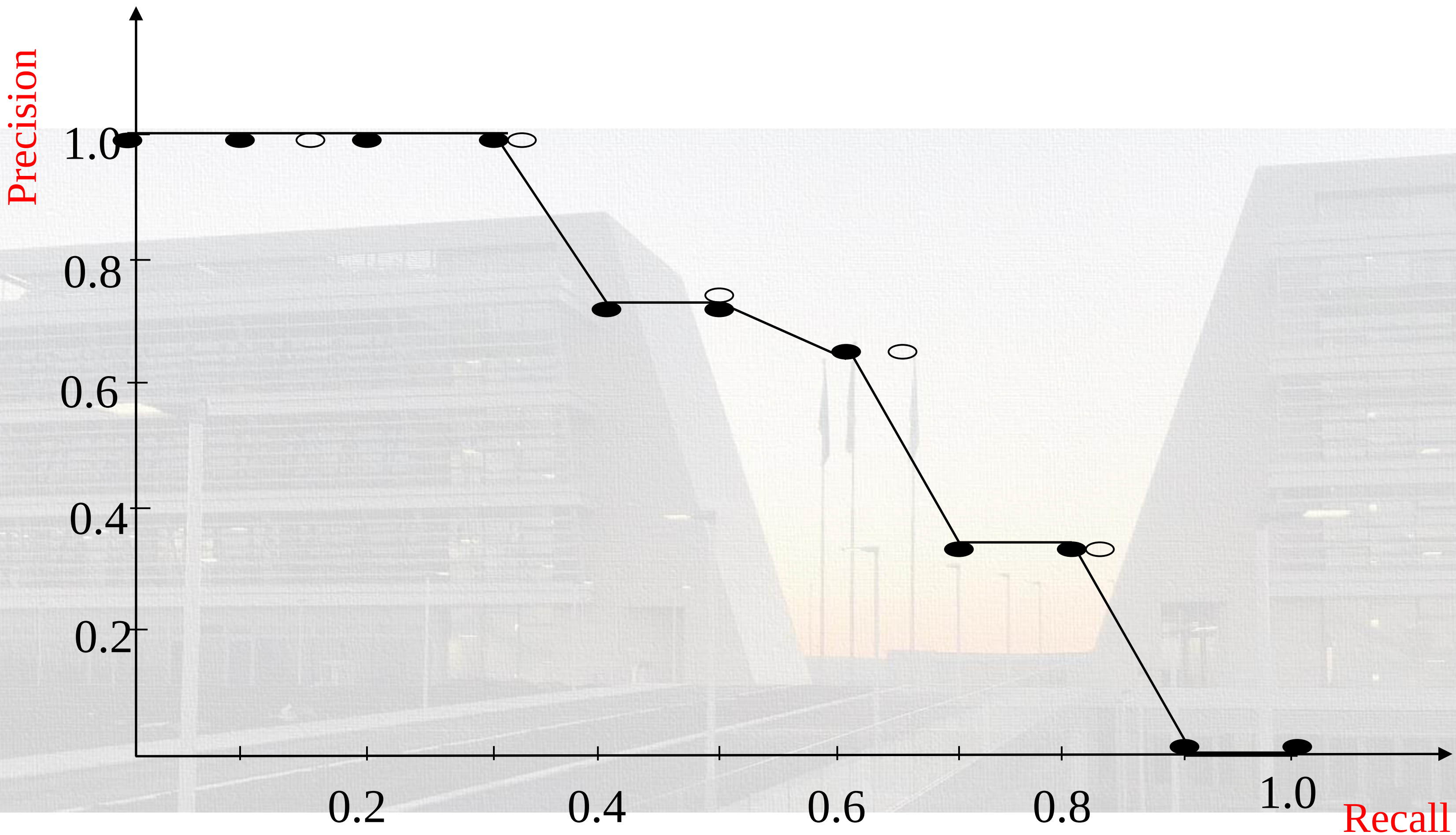
- A precision-recall graph for one query isn't a very sensible thing to look at
  - You need to average performance over a whole bunch of queries
- Some standard recall levels r_j are set. Typically:

  $r_0 = 0.0$, $r_1 = 0.1$, …, $r_{10}=1.0$ (11-point interpolated average precision)
- For each query the precision corresponding to each standard recall levels are estimated via interpolation:

$$P_{interp}(r_j) = \max_{r \geq r_j} P(r)$$

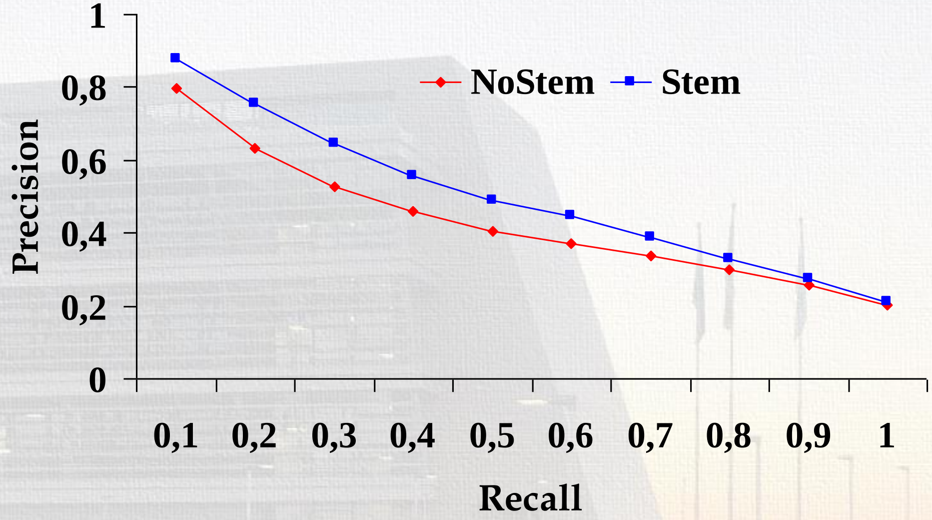
- Plot average precision/recall curves to evaluate overall system performance on a document/query corpus.

Interpolating a Recall/Precision Curve



Compare two or more Systems

- The curve closest to the upper right-hand corner of the graph indicates the best performance



- Graphs are good, but people may want a summary measure….

Ranking metrics

- Precision at fixed retrieval level
  - Precision-at-*k* *(P@k)*: Precision of top *k* results
  - Perhaps appropriate for most of web search: all people want are good matches on the first one or two result pages
- Mean Average Precision (MAP)

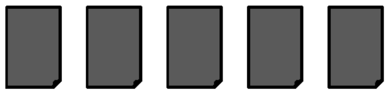
$$MAP(Q) = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{|R_q|} \sum_{d \in R_q} P@k_{q,d}$$

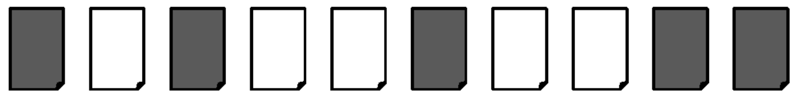
Q = set of queries

R_q=set of relevant documents for the query *q*

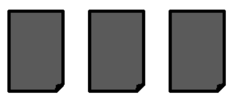
$K_{q,d}$=ranking of the document *d* retrieved throught the query *q*

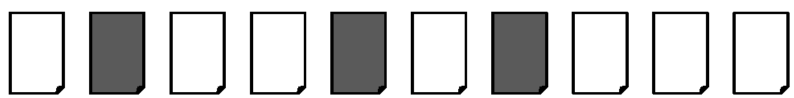
Mean Average Precision

 = relevant documents for query 1

Ranking #1 

Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

 = relevant documents for query 2

Ranking #2 

Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

$$\textit{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$
$$\textit{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$\textit{mean average precision} = (0.62 + 0.44)/2 = 0.53$$

Overview

- Performance Evaluation Metrics
  - Classifier Evaluation Metrics
  - Information Retrieval Systems Evaluation Metrics
- Tuning and Evaluation Methods
- **Error Diagnostics**

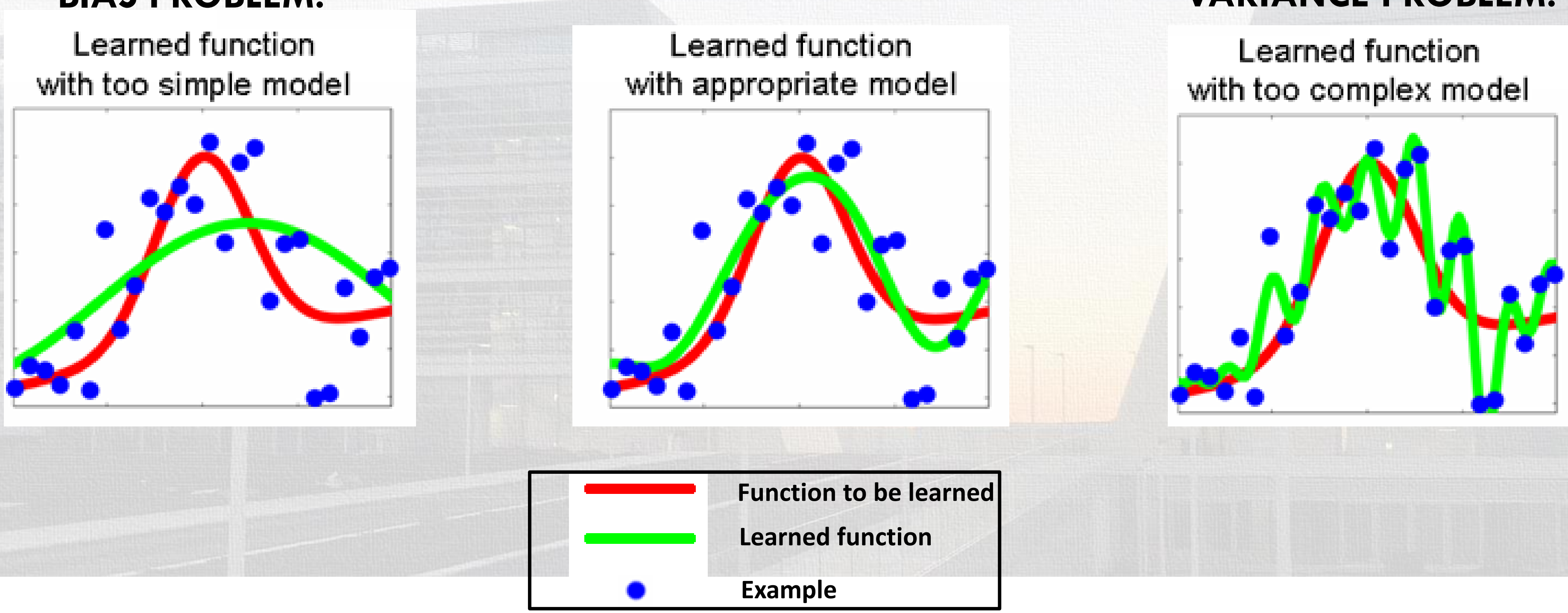
Error Diagnostics

- Error Diagnostics helps in identifying what problem is affecting an ML systems that performs poorly
- Understanding the problem is useful in coming up with promising solutions for improving the system
- Two opposite issues:
  - **Bias Problem**
  - **Variance Problem**

Bias Versus Variance

- Example in Regression

BIAS PROBLEM: **VARIANCE PROBLEM:**



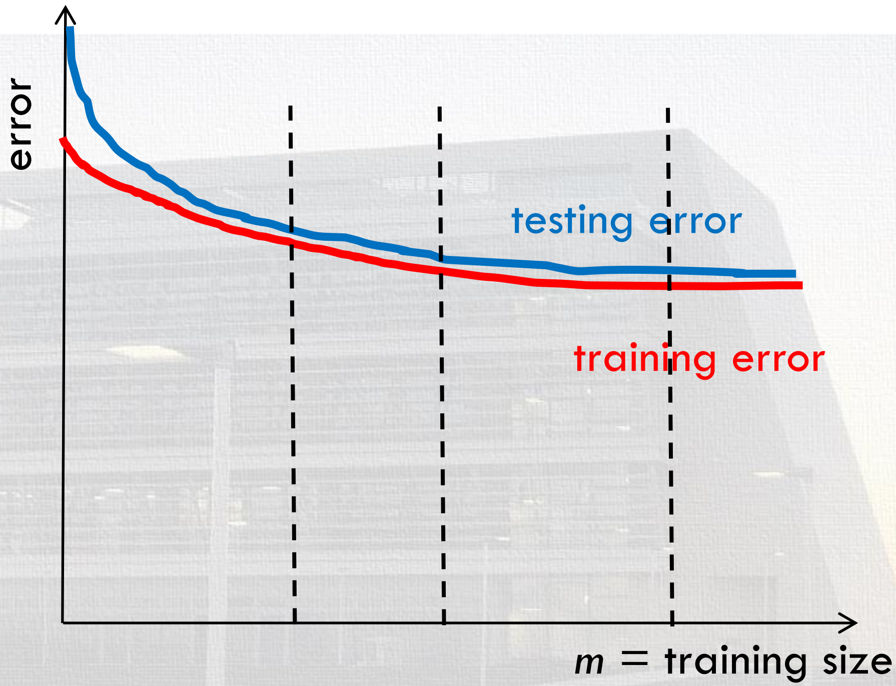
Diagnosing Bias vs Variance

- Bias
  - *Underfitting*: the model is not enough expressive to fit the complexity of the underlying concept to be learned
  - A high error is observed both in training and testing

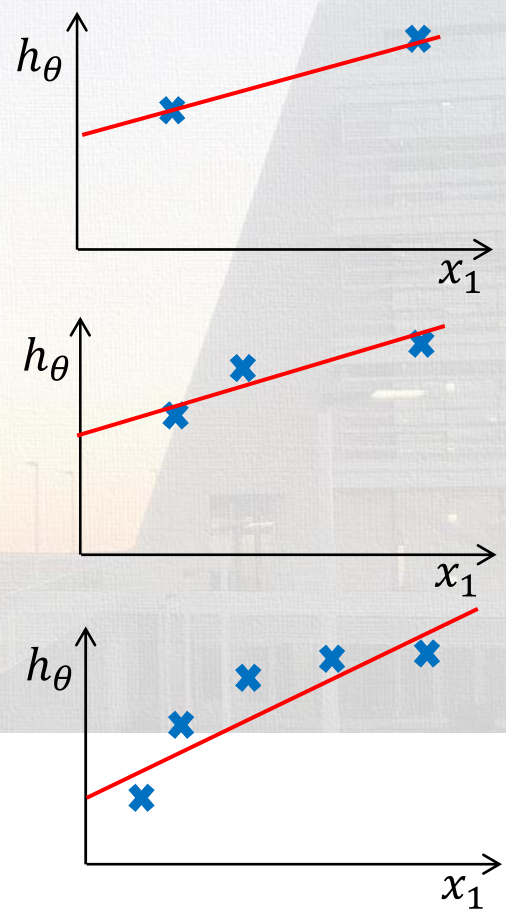
- Variance
  - *Overfitting*: the model perfectly fits training data but is too complex (example: an extremely deep decision tree) and does not generalize well on new data
  - A high difference between the training error and the testing error

Diagnosing High Bias via Learning Curve

Example in regression: we want to fit a 2D data distribution with a straight line



$$h_\theta(x) = \theta_0 + \theta_1 x_1$$

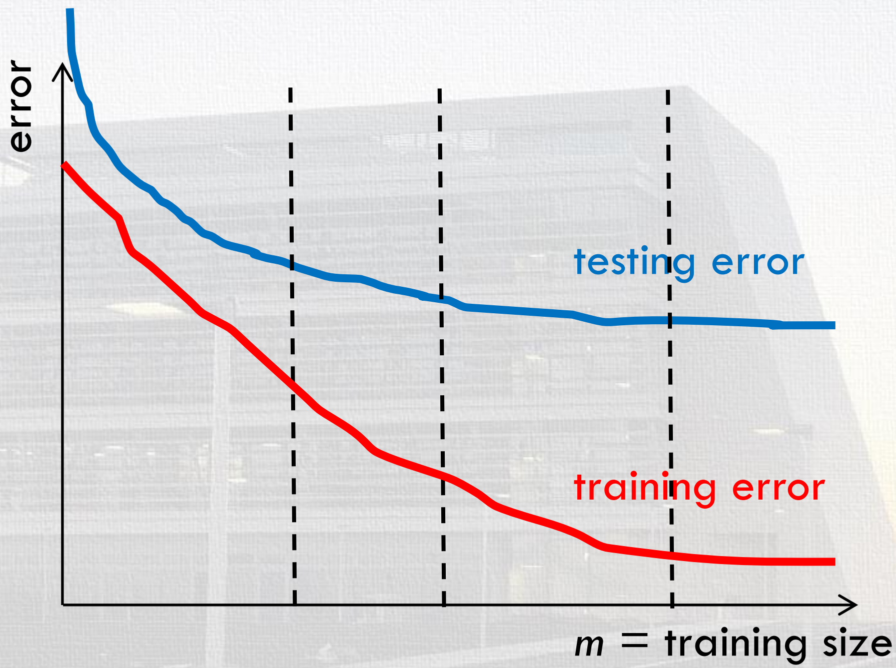


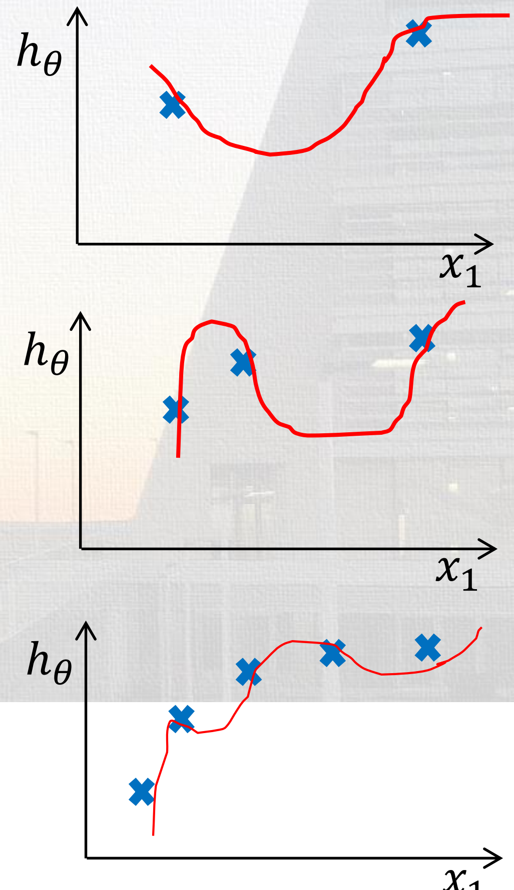
After a certain value of m, the learning process saturates and the testing error becomes similar to the training error → getting more example will not help too much

Diagnosing High Variance via Learning Curve

Example in regression: we want to fit a 2D data distribution with 10-th degree polynomial function

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \ \dots + \theta_{10} {x_1}^{10}$$





A large gap between the training error and the testing error is observed. The saturation point is still not reached → new examples should help

Solutions for Bias and Variance

- Bias
  - A different feature space may be needed. Add new informative features
  - Adopt a more sophisticated algorithm (or same learning policy but a more complex parameterization)

- Variance
  - More training data may be needed. Add new examples or adopt a data augmentation schema
  - Try to determine irrelevant and noisy features and remove them
  - Adopt a less complicated parameterization (e.g., a simpler polynomial function for regression)

Summary

- The effectiveness of ML or IR systems can be assessed with different evaluation metrics
  - we saw just the most popular, but a lot of other metrics exist!!!
- A reliable evaluation should follow some guideline
- Error diagnostics is useful for understanding how improving the system performance