Quantification

Using Supervised Learning to Estimate Class Prevalence

Fabrizio Sebastiani

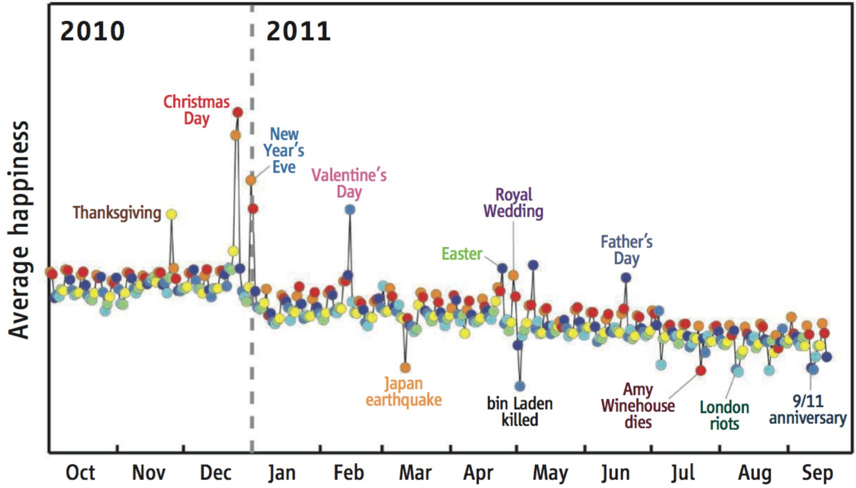
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, IT
E-mail: fabrizio.sebastiani@isti.cnr.it

December 12, 2017 @ UniPI

Download these slides at http://bit.ly/2nMMFQU

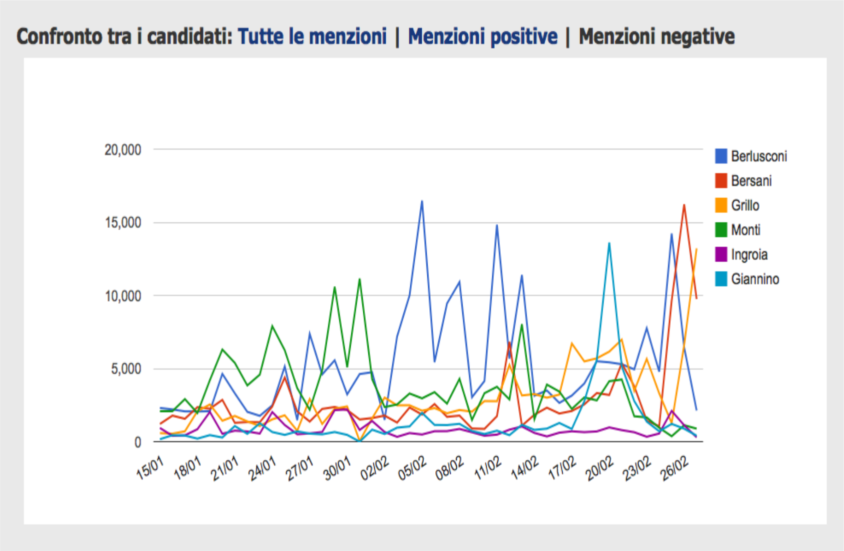
What is quantification?

[1]



[1] Dodds, Peter et al. Temporal Patterns of Happiness and Information in a Global Social Network: Hedonometrics and Twitter. *PLoS ONE*, 6(12), 2011.

What is quantification? (cont'd)



What is quantification? (cont'd)

- In many applications of classification, the real goal is determining the relative frequency (or: prevalence) of each class in the unlabelled data (quantification, a.k.a. supervised prevalence estimation)
- E.g.
  - Among the tweets about the next presidential elections, what is the fraction of pro-Democrat ones?
  - Among the posts about the Apple Watch 3 posted on forums, what is the fraction of "very negative" ones?
  - How have these percentages evolved over time?
- Quantification has been studied within IR, ML, DM, NLP, and has given rise to learning methods and evaluation measures specific to it
- We will mostly deal with text quantification

1 Introduction

2 Applications of Quantification in IR, ML, DM, NLP

3 Evaluation Measures for Quantification

4 Supervised Learning Methods for Prevalence Estimation

5 Resources and Shared Tasks

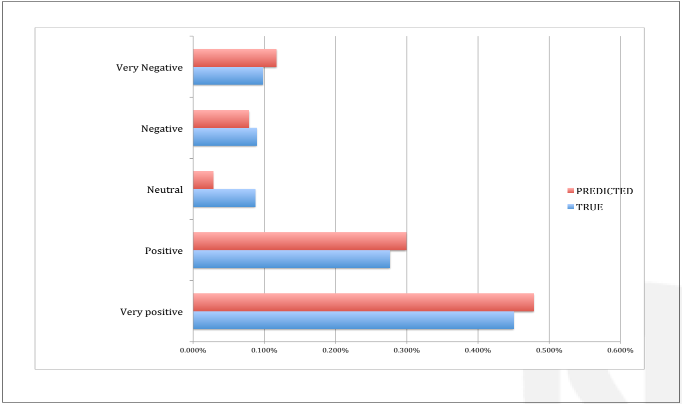
6 Conclusions

Outline

1. Introduction
2. Applications of Quantification in IR, ML, DM, NLP
3. Evaluation Measures for Quantification
4. Supervised Learning Methods for Prevalence Estimation
5. Resources and Shared Tasks
6. Conclusions

What is quantification? (cont'd)

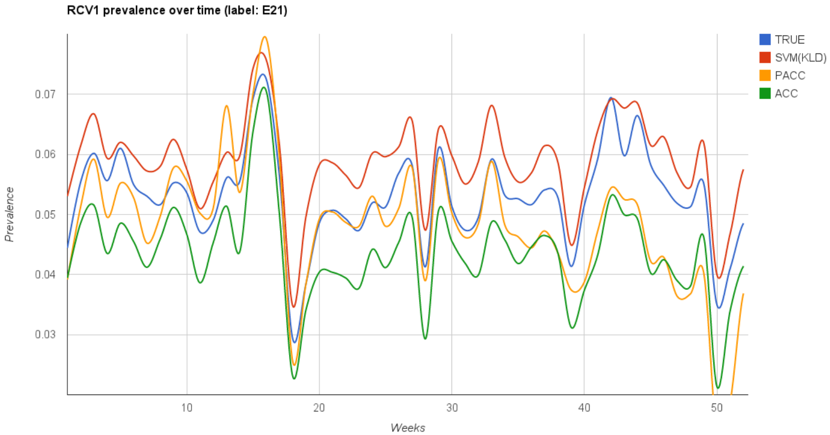
- Quantification may be also defined as the task of approximating a true distribution by a predicted distribution



- As a result, evaluation measures for quantification are divergences, which evaluate how much a predicted distribution diverges from the true distribution

Distribution drift

- The need to perform quantification arises because of distribution drift, i.e., the presence of a discrepancy between the class distribution of *Tr* and that of *Te*.



Distribution drift (cont'd)

- Distribution drift may derive when
  - the environment is not stationary across time and/or space and/or other variables, and the testing conditions are irreproducible at training time
  - the process of labelling training data is class-dependent (e.g., "stratified" training sets)
  - the labelling process introduces bias in the training set (e.g., if active learning is used)
- Distribution drift clashes with the IID assumption, on which standard ML algorithms are instead based.

The "paradox of quantification"

- Is "classify and count" the optimal quantification strategy?

The "paradox of quantification"

- Is "classify and count" the optimal quantification strategy? No!
- A perfect classifier is also a perfect "quantifier" (i.e., estimator of class prevalence), but ...
- ... a good classifier is not necessarily a good quantifier (and vice versa) :

The "paradox of quantification"

- Is "classify and count" the optimal quantification strategy? No!
- A perfect classifier is also a perfect "quantifier" (i.e., estimator of class prevalence), but ...
- ... a good classifier is not necessarily a good quantifier (and vice versa) :

	FP	FN
Classifier A	18	20
Classifier B	20	20

The "paradox of quantification"

- Is "classify and count" the optimal quantification strategy? No!
- A perfect classifier is also a perfect "quantifier" (i.e., estimator of class prevalence), but ...
- ... a good classifier is not necessarily a good quantifier (and vice versa) :

	FP	FN
Classifier A	18	20
Classifier B	20	20

- Paradoxically, we should prefer quantifier B to quantifier A, since A is biased
- This means that quantification should be studied as a task in its own right

Vapnik's Principle

- Key observation: classification is a more general problem than quantification
- Vapnik's principle:

  > *"If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem."*

- This suggests solving quantification directly (without solving classification as an intermediate step) with the goal of achieving higher quantification accuracy than if we opted for the indirect solution

Outline

1. Introduction
2. Applications of Quantification in IR, ML, DM, NLP
3. Evaluation Measures for Quantification
4. Supervised Learning Methods for Prevalence Estimation
5. Resources and Shared Tasks
6. Conclusions

Applications of quantification

A number of fields where classification is used are not interested in individual data, but in data aggregated across spatio-temporal contexts and according to other variables (e.g., gender, age group, religion, job type, ...); e.g.,

Applications of quantification

A number of fields where classification is used are not interested in individual data, but in data aggregated across spatio-temporal contexts and according to other variables (e.g., gender, age group, religion, job type, ...); e.g.,

- Social sciences : studying indicators concerning society and the relationships among individuals within it [2]

  > *[Others] may be interested in finding the needle in the haystack, but social scientists are more commonly interested in characterizing the haystack.*
  > *(Hopkins and King, 2010)*

  "Computational social science" is the big new paradigm spurred by the availability of "big data" from social networks

[2] D. Hopkins and G. King, A Method of Automated Nonparametric Content Analysis for Social Science. *American Journal of Political Science* 54(1), 2010.

Applications of quantification

A number of fields where classification is used are not interested in individual data, but in data aggregated across spatio-temporal contexts and according to other variables (e.g., gender, age group, religion, job type, ...); e.g.,

- Social sciences : studying indicators concerning society and the relationships among individuals within it [2]

  > *[Others] may be interested in finding the needle in the haystack, but social scientists are more commonly interested in characterizing the haystack.*
  >
  > *(Hopkins and King, 2010)*

  "Computational social science" is the big new paradigm spurred by the availability of "big data" from social networks
- Political science : e.g., predicting election results by estimating the prevalence of blog posts (or tweets) supporting a given candidate or party

[2] D. Hopkins and G. King, A Method of Automated Nonparametric Content Analysis for Social Science. *American Journal of Political Science* 54(1), 2010.

Applications of quantification (cont'd)

- Epidemiology : tracking the incidence and the spread of diseases; e.g.,
  - estimate pathology prevalence from clinical reports where pathologies are diagnosed
  - estimate the prevalence of different causes of death from "verbal autopsies", i.e., from verbal accounts of symptoms

Applications of quantification (cont'd)

- Epidemiology : tracking the incidence and the spread of diseases; e.g.,
  - estimate pathology prevalence from clinical reports where pathologies are diagnosed
  - estimate the prevalence of different causes of death from "verbal autopsies", i.e., from verbal accounts of symptoms
- Market Research : estimating the distribution of consumers' attitudes about products, product features, or marketing strategies; e.g.,
  - quantifying customers' attitudes from verbal responses to open-ended questions[3]

[3] Esuli, A. and F. Sebastiani: 2010, Machines that Learn how to Code Open-Ended Survey Data. *International Journal of Market Research* 52(6), 775–800.

Applications of quantification (cont'd)

- Natural Language Processing : e.g., tuning a word sense disambiguator to a domain characterized by sense priors different from those of the training set

Applications of quantification (cont'd)

- Natural Language Processing : e.g., tuning a word sense disambiguator to a domain characterized by sense priors different from those of the training set
- Machine Learning : e.g., estimating the class prevalence of the test set in order to improve the performance of classifiers trained on data with different class prevalence

Applications of quantification (cont'd)

- Natural Language Processing : e.g., tuning a word sense disambiguator to a domain characterized by sense priors different from those of the training set
- Machine Learning : e.g., estimating the class prevalence of the test set in order to improve the performance of classifiers trained on data with different class prevalence
- Others : e.g.,
  - estimating the proportion of no-shows within a set of bookings
  - estimating the proportions of different types of cells in blood samples

Dimensions of quantification

- Text quantification, like text classification, may be performed across various dimensions (i.e., criteria):
  - by topic : applications to the social sciences, epidemiology, market research, resource allocation, word sense disambiguation
  - by sentiment ("sentiment classification"): applications to the social sciences, political sciences, market research, ...
  - by language ("language identification"): e.g., estimating language diversity
- Applications of quantification found in the literature may be distinguished into
  - those that apply methods especially designed for quantification
  - those that, unaware of the existence of specific methods for quantification, apply standard classification methods with "classify and count"

Outline

1. Introduction
2. Applications of Quantification in IR, ML, DM, NLP
3. Evaluation Measures for Quantification
4. Supervised Learning Methods for Prevalence Estimation
5. Resources and Shared Tasks
6. Conclusions

Notation and terminology

- Domain $\mathcal{X}$ of items (documents), set $\mathcal{C}$ of classes
- Different brands of classification :
  - Binary classification: each item has exactly one of $\mathcal{C} = \{c_1, c_2\}$
  - Single-label multi-class classification (SLMC): each item has exactly one of $\mathcal{C} = \{c_1, ..., c_n\}$, with $n > 2$
  - Multi-label multi-class classification(MLMC) : each item may have zero, one, or several among $\mathcal{C} = \{c_1, ..., c_n\}$, with $n > 1$
    - MLMC is usually reduced to binary by solving n independent binary classification problems
  - Ordinal classification (aka "ordinal regression"): each item has exactly one of $\mathcal{C} = (c_1 \preceq ... \preceq c_n)$, where $\preceq$ is a total order and $n > 2$
  - (Metric regression): each item has a real-valued score from the range $[\alpha, \beta]$
- For each such brand of classification we will be interested in its "quantification equivalent" (Q-equivalent), i.e., in solving and evaluating that classification task at the aggregate level.

Notation and terminology (cont'd)

$\mathbf{x}$	vectorial representation of item x
$\mathcal{C} = \{c_1, ..., c_n\}$	set of classes
$p_S(c_j)$	true prevalence (aka "prior probability") of c_j in set S
$\hat{p}_S(c_j)$	estimated prevalence of c_j in set S
$\hat{p}_S^M(c_j)$	estimate $\hat{p}_S(c_j)$ obtained via method M
$p(c_j \vert \mathbf{x})$	posterior probability of c_j returned by the classifier
$p(\delta_j)$	probability that classifier attributes c_j to a random item
$p_S(\delta_j)$	fraction of items in S labelled as c_j by the classifier

How do we evaluate quantification methods?

- Evaluating quantification means measuring how well a predicted probabilistic distribution $\hat{p}(c)$ fits a true distribution $p(c)$
- The goodness of fit between two distributions can be computed via divergence functions, which enjoy

1. $D(p, \hat{p}) = 0$ only if $p = \hat{p}$ (identity of indiscernibles)
2. $D(p, \hat{p}) \geq 0$ (non-negativity)

and may enjoy (as exemplified in the binary case)

3. If $\hat{p}'(c_1) = p(c_1) - a$ and $\hat{p}''(c_1) = p(c_1) + a$, then $D(p, \hat{p}') = D(p, \hat{p}'')$ (impartiality)
4. If $\hat{p}'(c_1) = p'(c_1) \pm a$ and $\hat{p}''(c_1) = p''(c_1) \pm a$, with $p'(c_1) < p''(c_1) \leq 0.5$, then $D(p, \hat{p}') > D(p, \hat{p}'')$ (relativity)

How do we evaluate quantification methods? (cont'd)

Divergences frequently used for evaluating (multiclass) quantification are

- $\text{MAE}(p,\hat{p}) = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}|\hat{p}(c)-p(c)|$ (Mean Absolute Error)
- $\text{MRAE}(p,\hat{p}) = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{|\hat{p}(c)-p(c)|}{p(c)}$ (Mean Relative Absolute Error)
- $\text{KLD}(p,\hat{p}) = \sum_{c\in\mathcal{C}}p(c)\log\frac{p(c)}{\hat{p}(c)}$ (Kullback-Leibler Divergence)

	Impartiality	Relativity
Mean Absolute Error	Yes	No
Mean Relative Absolute Error	Yes	Yes
Kullback-Leibler Divergence	No	Yes

How do we evaluate quantification methods? (cont'd)

- MRAE and KLD may sometimes be undefined due to the presence of zero denominators.
- To solve this we can smooth $p(c)$ and $\hat{p}(c)$ via additive smoothing; the smoothed version of $p(c)$ is

$$p_s(c) = \frac{\epsilon + p(c)}{\epsilon|\mathcal{C}| + \sum_{c \in \mathcal{C}} p(c)} \tag{1}$$

- $\epsilon = \dfrac{1}{2|Te|}$ is often used as a smoothing factor

Multi-objective measures

- The "paradox of quantification":
  1. Classifier A : $CT_1 = (TP = 0, FP = 1000, FN = 1000, TN = 0)$
  2. Classifier B : $CT_2 = (TP = 990, FP = 0, FN = 10, TN = 1000)$

  A yields better KLD than B!, but we intuitively prefer A to B
- It is difficult to trust a quantifier if it is not also a good enough classifier ...
- The multi-objective measure[4] MOM strives to keep both classification and quantification error low

$$
\begin{aligned}
MOM(p, \hat{p}) &= \sum_{c_j \in \mathcal{C}} |FP_j^2 - FN_j^2| \\
&= \sum_{c_j \in \mathcal{C}} (FN_j + FP_j) \cdot |FN_j - FP_j|
\end{aligned}
$$

since

- $|FN_j - FP_j|$ is a measure of quantification error
- $(FN_j + FP_j)$ is a measure of classification error

[4] Milli, L., A. Monreale, G. Rossetti, F. Giannotti, D. Pedreschi, F. Sebastiani, Quantification Trees. In: ICDM 2013, pp. 528–536.

Outline

1. Introduction
2. Applications of Quantification in IR, ML, DM, NLP
3. Evaluation Measures for Quantification
4. Supervised Learning Methods for Prevalence Estimation
5. Resources and Shared Tasks
6. Conclusions

Quantification methods

- Quantification methods belong to two classes
  - 1. Aggregative : they require the classification of individual items as a basic step
  - 2. Non-aggregative : quantification is performed without performing classification
- Aggregative methods may be further subdivided into
  - 1a. Methods using general-purpose learners (i.e., originally devised for classification); can use any supervised learning algorithm that returns posterior probabilities
  - 1b. Methods using special-purpose learners (i.e., especially devised for quantification)

Quantification methods: CC

- Classify and Count (CC) consists of
  1. generating a classifier from Tr
  2. classifying the items in Te
  3. estimating $p_{Te}(c_j)$ by counting the items predicted to be in c_j, i.e.,

$$\hat{p}^{CC}_{Te}(c_j) = p_{Te}(\delta_j)$$

- But a good classifier is not necessarily a good quantifier ...
- CC suffers from the problem that "standard" classifiers are usually tuned to minimize $(FP + FN)$ or a proxy of it, but not $|FP - FN|$
  - E.g., in recent experiments of ours, out of 5148 binary test sets averaging 15,000+ items each, standard (linear) SVM brought about an average FP/FN ratio of 0.109.

Quantification methods: PCC

- Probabilistic Classify and Count (PCC) estimates p_{Te} by simply counting the expected fraction of items predicted to be in the class, i.e.,

$$\hat{p}^{PCC}_{Te}(c_j) = E_{Te}[c_j] = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} p(c_j|\mathbf{x})$$

- The rationale is that posterior probabilities contain richer information than binary decisions, which are obtained from posterior probabilities by thresholding.
- Shown to perform very well in (Gao and Sebastiani, 2016)[5].

[5] W. Gao and F. Sebastiani. From Classification to Quantification in Tweet Sentiment Analysis. *Social Network Analysis and Mining*, 6(19), 1–22, 2016.

Quantification methods: ACC

- Adjusted Classify and Count (ACC) is based on the observation that, after we have classified the test documents in Te, for all $c_j \in \mathcal{C}$ it holds that

$$p_{Te}(\delta_j) = \sum_{c_i \in \mathcal{C}} p_{Te}(\delta_j | c_i) \cdot p_{Te}(c_i)$$

Quantification methods: ACC

- Adjusted Classify and Count (ACC) is based on the observation that, after we have classified the test documents in Te, for all $c_j \in \mathcal{C}$ it holds that

$$p_{Te}(\delta_j) = \sum_{c_i \in \mathcal{C}} p_{Te}(\delta_j | c_i) \cdot p_{Te}(c_i)$$

- The $p_{Te}(\delta_j)$'s are observed

Quantification methods: ACC

- Adjusted Classify and Count (ACC) is based on the observation that, after we have classified the test documents in Te, for all $c_j \in \mathcal{C}$ it holds that

$$p_{Te}(\delta_j) = \sum_{c_i \in \mathcal{C}} p_{Te}(\delta_j | c_i) \cdot p_{Te}(c_i)$$

- The $p_{Te}(\delta_j)$'s are observed
- The $p_{Te}(\delta_j | c_i)$'s can be estimated on Tr via k-fold cross-validation (these latter represent the system's bias).

Quantification methods: ACC

- Adjusted Classify and Count (ACC) is based on the observation that, after we have classified the test documents in Te, for all $c_j \in \mathcal{C}$ it holds that

$$p_{Te}(\delta_j) = \sum_{c_i \in \mathcal{C}} p_{Te}(\delta_j|c_i) \cdot p_{Te}(c_i)$$

- The $p_{Te}(\delta_j)$'s are observed
- The $p_{Te}(\delta_j|c_i)$'s can be estimated on Tr via k-fold cross-validation (these latter represent the system's bias).
- This results in a system of $|\mathcal{C}|$ linear equations (one for each c_j) with $|\mathcal{C}|$ unknowns (the $p_{Te}(c_i)$'s).
- ACC consists of solving this system, i.e., of correcting the class prevalence estimates $p_{Te}(\delta_j)$ obtained by CC according to the estimated system's bias.

Quantification methods: EMQ

- Accurate quantification may improve classification accuracy since, in the presence of distribution drift, classification accuracy may suffer
- E.g., in a Naïve Bayesian classifier

$$p(c|\mathbf{x}) = \frac{p(\mathbf{x}|c)p(c)}{p(\mathbf{x})}$$

  posterior probabilities have been "calibrated" for *Tr*
- Probabilities are calibrated for a set S when

$$p_S(c) = E_S[c] = \frac{1}{|S|}\sum_{\mathbf{x}\in S} p(c|\mathbf{x})$$

  which means that in the presence of distribution drift they cannot be calibrated for both *Tr* and *Te*

Quantification methods: EMQ (cont'd)

- By estimating class prevalence in Te we can adjust the classifier itself so as to yield better classification accuracy
- EMQ : an iterative, EM-based "quantification" method for improving classification accuracy[6]
- EMQ consists of an iterative recalibration of the posterior probabilities $p(c|\mathbf{x})$ for the test set Te, until convergence
- The class prevalences $p_{Te}(c)$ are the "byproducts" of this process

[6] Saerens, M., P. Latinne, and C. Decaestecker: 2002, Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation* 14(1), 21–41.

Quantification methods: EMQ (cont'd)

- We apply EM in the following way until convergence of the $\hat{p}^{(s)}(c)$:
  - **Step 0**: For each $c \in \mathcal{C}$ initialize $\hat{p}^{(0)}(c) \leftarrow p_{Tr}(c)$
    For each $\mathbf{x} \in Te$ initialize $p^{(0)}(c|\mathbf{x}) \leftarrow p(c|\mathbf{x})$
  - **Step s**: Iterate:
    - **Step s(E)**: For each c compute:

$$\hat{p}^{(s+1)}(c) = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} p^{(s)}(c|\mathbf{x}) \tag{2}$$

    - **Step s(M)**: For each test item $\mathbf{x}$ and each c compute:

$$p^{(s+1)}(c|\mathbf{x}) = \frac{\dfrac{\hat{p}^{(s+1)}(c)}{p^{(s)}(c)} \cdot p^{(s)}(c|\mathbf{x})}{\displaystyle\sum_{c \in \mathcal{C}} \dfrac{\hat{p}^{(s+1)}(c)}{p^{(s)}(c)} \cdot p^{(s)}(c|\mathbf{x})} \tag{3}$$

- Step s(E) re-estimates the priors in terms of the new posterior probabilities
- Step s(M) re-calibrates the posterior probabilities by using the new priors

Quantification methods: EMQ (cont'd)

- We apply EM in the following way until convergence of the $\hat{p}^{(s)}(c)$:
  - **Step 0**: For each $c \in \mathcal{C}$ initialize $\hat{p}^{(0)}(c) \leftarrow p_{Tr}(c)$
    For each $\mathbf{x} \in Te$ initialize $p^{(0)}(c|\mathbf{x}) \leftarrow p(c|\mathbf{x})$
  - **Step s**: Iterate:
    - **Step s(E)**: For each c compute:

$$\hat{p}^{(s+1)}(c) = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} p^{(s)}(c|\mathbf{x}) \tag{2}$$

    - **Step s(M)**: For each test item $\mathbf{x}$ and each c compute:

$$p^{(s+1)}(c|\mathbf{x}) = \frac{\dfrac{\hat{p}^{(s+1)}(c)}{p^{(s)}(c)} \cdot p^{(s)}(c|\mathbf{x})}{\displaystyle\sum_{c \in \mathcal{C}} \dfrac{\hat{p}^{(s+1)}(c)}{p^{(s)}(c)} \cdot p^{(s)}(c|\mathbf{x})} \tag{3}$$

- Step s(E) re-estimates the priors in terms of the new posterior probabilities
- Step s(M) re-calibrates the posterior probabilities by using the new priors

Quantification methods: EMQ (cont'd)

- We apply EM in the following way until convergence of the $\hat{p}^{(s)}(c)$:
  - **Step 0**: For each $c \in \mathcal{C}$ initialize $\hat{p}^{(0)}(c) \leftarrow p_{Tr}(c)$
    For each $\mathbf{x} \in Te$ initialize $p^{(0)}(c|\mathbf{x}) \leftarrow p(c|\mathbf{x})$
  - **Step s**: Iterate:
    - **Step s(E)**: For each c compute:

$$\hat{p}^{(s+1)}(c) = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} p^{(s)}(c|\mathbf{x}) \tag{2}$$

    - **Step s(M)**: For each test item $\mathbf{x}$ and each c compute:

$$p^{(s+1)}(c|\mathbf{x}) = \frac{\dfrac{\hat{p}^{(s+1)}(c)}{p^{(s)}(c)} \cdot p^{(s)}(c|\mathbf{x})}{\displaystyle\sum_{c \in \mathcal{C}} \dfrac{\hat{p}^{(s+1)}(c)}{p^{(s)}(c)} \cdot p^{(s)}(c|\mathbf{x})} \tag{3}$$

- Step s(E) re-estimates the priors in terms of the new posterior probabilities
- Step s(M) re-calibrates the posterior probabilities by using the new priors

Quantification methods: EMQ (cont'd)

- We apply EM in the following way until convergence of the $\hat{p}^{(s)}(c)$:
  - **Step 0**: For each $c \in \mathcal{C}$ initialize $\hat{p}^{(0)}(c) \leftarrow p_{Tr}(c)$
    For each $\mathbf{x} \in Te$ initialize $p^{(0)}(c|\mathbf{x}) \leftarrow p(c|\mathbf{x})$
  - **Step s**: Iterate:
    - **Step s(E)**: For each c compute:

$$\hat{p}^{(s+1)}(c) = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} p^{(s)}(c|\mathbf{x}) \tag{2}$$

    - **Step s(M)**: For each test item $\mathbf{x}$ and each c compute:

$$p^{(s+1)}(c|\mathbf{x}) = \frac{\dfrac{\hat{p}^{(s+1)}(c)}{p^{(s)}(c)} \cdot p^{(s)}(c|\mathbf{x})}{\displaystyle\sum_{c \in \mathcal{C}} \dfrac{\hat{p}^{(s+1)}(c)}{p^{(s)}(c)} \cdot p^{(s)}(c|\mathbf{x})} \tag{3}$$

- Step s(E) re-estimates the priors in terms of the new posterior probabilities
- Step s(M) re-calibrates the posterior probabilities by using the new priors

Quantification methods: SVM(KLD)

- Most researchers using aggregative methods have used general-purpose learning algorithms optimized for classification;
- Alternative idea: use special-purpose learning algorithms optimized directly for quantification[7]
- SVM(KLD): use explicit loss minimization, i.e., use a learner which directly optimizes the evaluation measure ("loss") used for quantification

[7] A. Esuli and F. Sebastiani. Optimizing Text Quantifiers for Multivariate Loss Functions. *ACM Transactions on Knowledge Discovery and Data*, 9(4), Article 27, 2015.

Quantification methods: SVM(KLD) (cont'd)

- Problem:
    - The loss functions most learners (e.g., AdaBoost, SVMs) can be optimized for must be linear (i.e., the error on the test set is a linear combination of the error generated by each test example) / univariate (i.e., each test item can be taken into consideration in isolation)
    - Loss functions for quantification are nonlinear (the impact of the error on a test item depends on how the other test items have been classified) / multivariate (they must take in consideration all test items at once)
- SVM_{perf}, a structured output learning algorithm that can be optimized for arbitrary nonlinear / multivariate measures
- SVM(KLD) tailors SVM_{perf} to use KLD as a loss

Quantification methods: SVM(KLD) (cont'd)

- Quantification accuracy is often analysed by class prevalence ...

Table: Accuracy as measured in terms of KLD on the 5148 test sets of RCV1-v2 grouped by class prevalence in *Tr*

RCV1-v2	VLP	LP	HP	VHP	All
SVM(KLD)	**2.09E-03**	**4.92E-04**	7.19E-04	1.12E-03	**1.32E-03**
PACC	2.16E-03	1.70E-03	**4.24E-04**	2.75E-04	1.74E-03
ACC	2.17E-03	1.98E-03	5.08E-04	6.79E-04	1.87E-03
MAX	2.16E-03	2.48E-03	6.70E-04	**9.03E-05**	2.03E-03
CC	2.55E-03	3.39E-03	1.29E-03	1.61E-03	2.71E-03
X	3.48E-03	8.45E-03	1.32E-03	2.43E-04	4.96E-03
PCC	1.04E-02	6.49E-03	3.87E-03	1.51E-03	7.86E-03
MM(PP)	1.76E-02	9.74E-03	2.73E-03	1.33E-03	1.24E-02
MS	1.98E-02	7.33E-03	3.70E-03	2.38E-03	1.27E-02
T50	1.35E-02	1.74E-02	7.20E-03	3.17E-03	1.38E-02
MM(KS)	2.00E-02	1.14E-02	9.56E-04	3.62E-04	1.40E-02

Quantification methods: SVM(KLD) (cont'd)

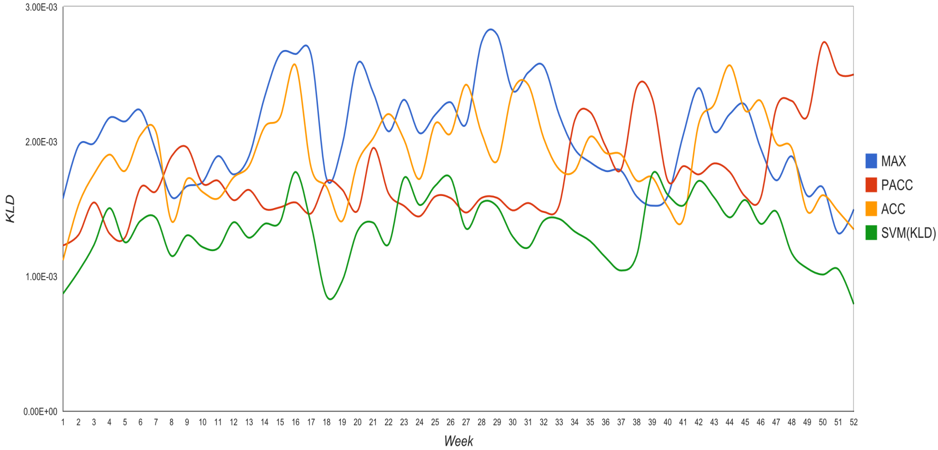
- ... or by amount of drift ...

Table: Accuracy as measured in terms of KLD on the 5148 test sets of RCV1-v2 grouped into quartiles homogeneous by distribution drift

RCV1-v2	VLD	LD	HD	VHD	All
SVM(KLD)	**1.17E-03**	**1.10E-03**	**1.38E-03**	1.67E-03	**1.32E-03**
PACC	1.92E-03	2.11E-03	1.74E-03	**1.20E-03**	1.74E-03
ACC	1.70E-03	1.74E-03	1.93E-03	2.14E-03	1.87E-03
MAX	2.20E-03	2.15E-03	2.25E-03	1.52E-03	2.03E-03
CC	2.43E-03	2.44E-03	2.79E-03	3.18E-03	2.71E-03
X	3.89E-03	4.18E-03	4.31E-03	7.46E-03	4.96E-03
PCC	8.92E-03	8.64E-03	7.75E-03	6.24E-03	7.86E-03
MM(PP)	1.26E-02	1.41E-02	1.32E-02	1.00E-02	1.24E-02
MS	1.37E-02	1.67E-02	1.20E-02	8.68E-03	1.27E-02
T50	1.17E-02	1.38E-02	1.49E-02	1.50E-02	1.38E-02
MM(KS)	1.41E-02	1.58E-02	1.53E-02	1.10E-02	1.40E-02

Quantification methods: SVM(KLD) (cont'd)

- ... or along the temporal dimension ...



Outline

1. Introduction
2. Applications of Quantification in IR, ML, DM, NLP
3. Evaluation Measures for Quantification
4. Supervised Learning Methods for Prevalence Estimation
5. Resources and Shared Tasks
6. Conclusions

Software resources for quantification

- A. Esuli and F. Sebastiani. Optimizing Text Quantifiers for Multivariate Loss Functions. *ACM Transactions on Knowledge Discovery from Data*, 9(4): Article 27, 2015. Contains links to quantification software & datasets.
- W. Gao and F. Sebastiani. From Classification to Quantification in Tweet Sentiment Analysis. *Social Network Analysis and Mining*, 6(19), 1–22, 2016. Contains links to quantification software & datasets.
- Hopkins, D.J. and G. King: 2010, A Method of Automated Nonparametric Content Analysis for Social Science. *American Journal of Political Science* 54(1), 229–247. Contains links to quantification software.

Shared tasks

- SemEval 2016 Task 4: "Sentiment Analysis in Twitter" (http://alt.qcri.org/semeval2016/task4/)
  - Subtask D: **Tweet quantification according to a two-point scale**:
    - Given a set of tweets about a given topic, estimate the distribution of the tweets across the "Positive" and "Negative" labels.
    - Evaluation measure is KLD
  - Subtask E: **Tweet quantification according to a five-point scale**:
    - Given a set of tweets about a given topic, estimate the distribution of the tweets across the five classes of a five-point scale.
    - Evaluation measure is Earth Mover's Distance
- Run again in 2017

Outline

1. Introduction
2. Applications of Quantification in IR, ML, DM, NLP
3. Evaluation Measures for Quantification
4. Supervised Learning Methods for Prevalence Estimation
5. Resources and Shared Tasks
6. Conclusions

Conclusion

- Quantification: a relatively (yet) unexplored new task, with many research problems still open
- Growing awareness that quantification is going to be more and more important; given the advent of big data, application contexts will spring up in which we will simply be happy with analysing data at the aggregate (rather than at the individual) level

Questions?

Thank you!

For any question, email me at

fabrizio.sebastiani@isti.cnr.it