Text Quantification

Fabrizio Sebastiani

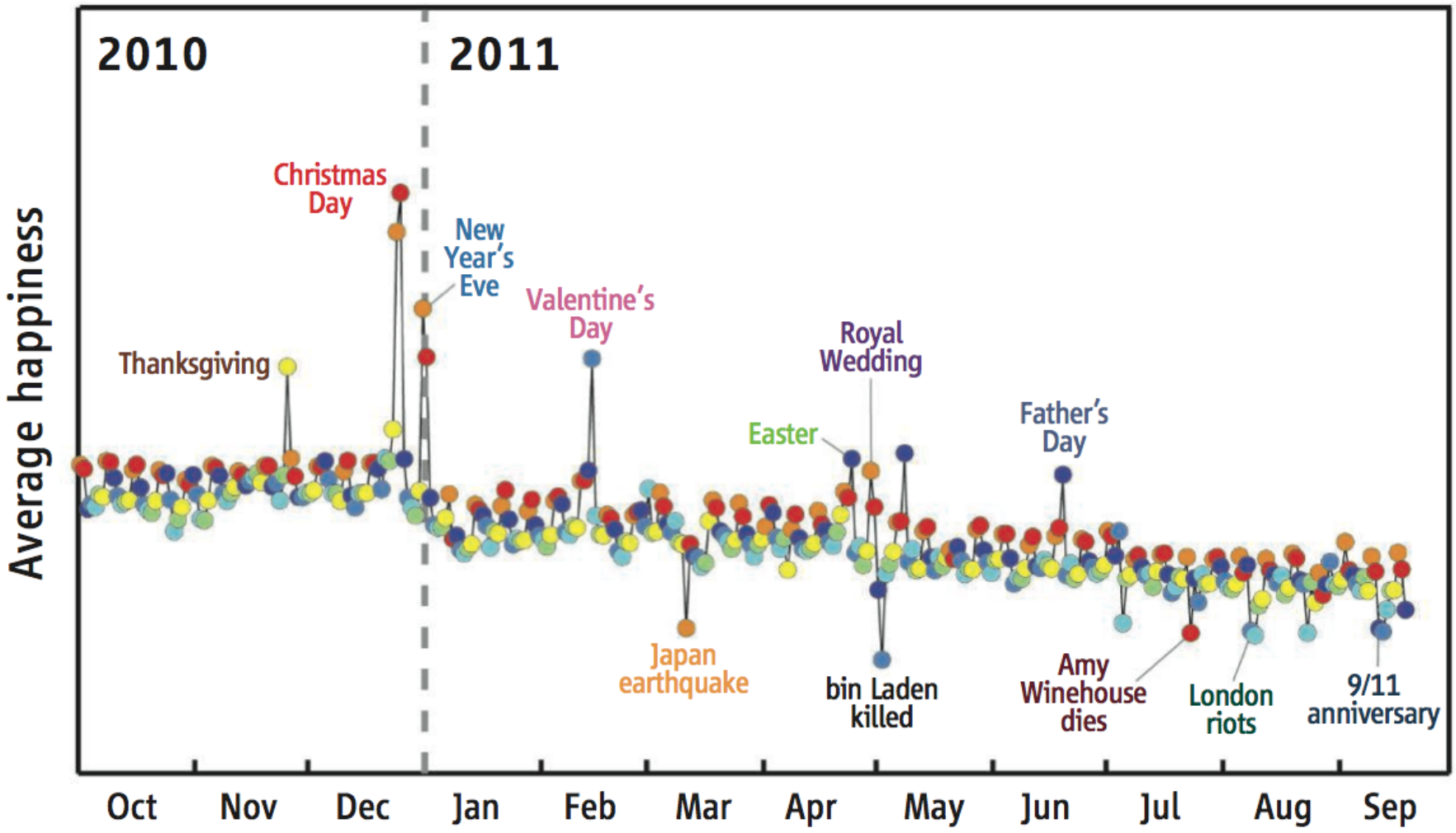
Qatar Computing Research Institute
Qatar Foundation
PO Box 5825 – Doha, Qatar
E-mail: fsebastiani@qf.org.qa
http://www.qcri.com/

RUSSIR 2015
St. Petersburg, RU – August 24–28, 2015

Download most recent version of these slides at
http://bit.ly/1PbcrBv

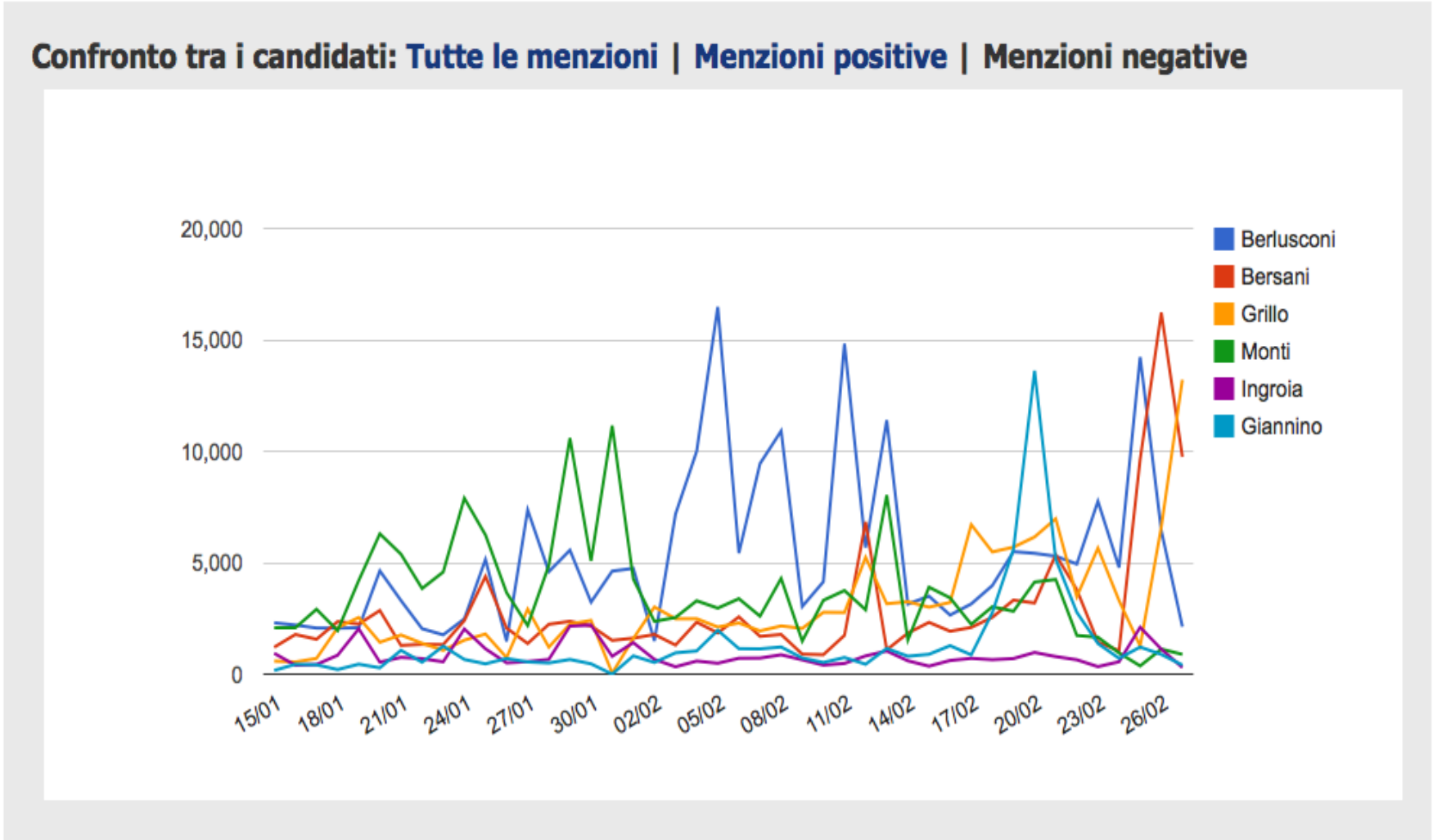
What is quantification?

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)

- Classification : a ubiquitous enabling technology in data science
- In many applications of classification, the real goal is determining the relative frequency of each class in the unlabelled data; this task is called quantification
- E.g.
  - Among the blog posts concerning the next presidential elections, what is the percentage of pro-Democrat posts?
  - Among the posts about the iPhone6 posted on forums, what is the percentage of "very positive" ones?
  - How do these percentages evolve over time?
- This task has applications in IR, ML, DM, NLP, and has given rise to learning methods and evaluation measures specific to it
- We will mostly be interested in quantification from text

Outline

Outline

Classification: A Primer

- Classification (aka "categorization") is the task of assigning data items to groups ("classes") whose existence is known in advance
- Examples :
  1. Assigning newspaper articles to one or more of `Home News`, `Politics`, `Economy`, `Lifestyles`, `Sports`
  2. Assigning email messages to exactly one of `Legitimate`, `Spam`
  3. Assigning product reviews to exactly one of `Disastrous`, `Poor`, `Average`, `Good`, `Excellent`
  4. Assigning one or more classes from the ACM Classification Scheme to a computer science paper
  5. Assigning photographs to one of `Still Life`, `Portrait`, `Landscape`, `Events`
  6. Predicting tomorrow's weather as one of `Sunny`, `Cloudy`, `Rainy`

Classification: A Primer (cont'd)

- Classification is different from clustering, since in the latter case the groups (and sometimes their number) are not known in advance
- Classification requires subjective judgment : assigning natural numbers to either `Prime` or `NonPrime` is #not# classification
- Classification is thus prone to error; we may experimentally evaluate the error made by a classifier against a set of manually classified objects

Classification: A Primer (cont'd)

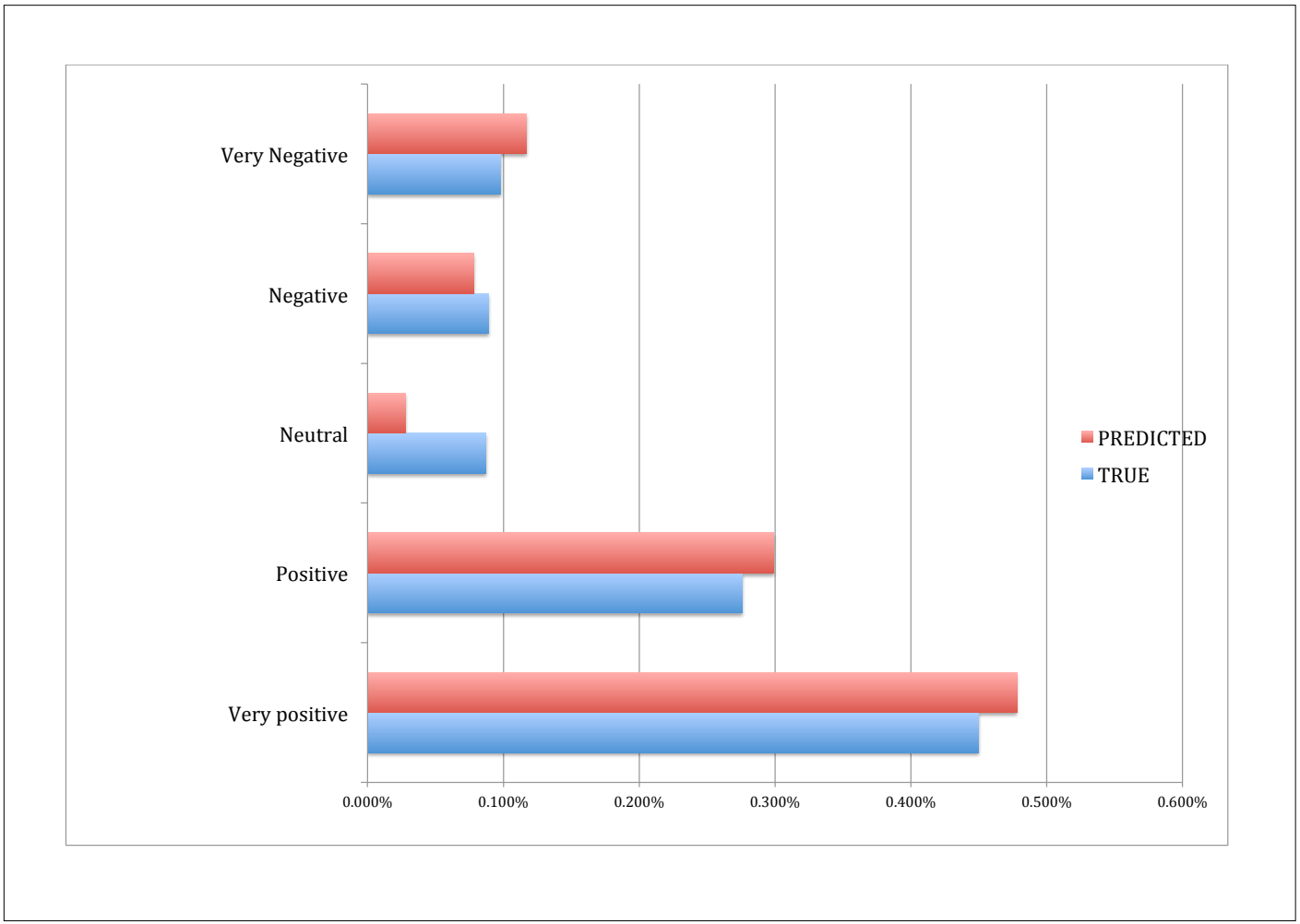
- (Automatic) Classification is usually tackled via supervised machine learning : a general-purpose learning algorithm trains (using a set of manually classified items) a classifier to recognize the characteristics an item should have in order to be attributed to a given class
- "Learning" metaphor: advantageous, since
  - no domain knowledge required to build a classifier (cheaper to manually classify some items for training than encoding domain knowledge by hand into the classifier)
  - easy to revise the classifier (a) if new training items become available, or (b) if new classes need to be considered
- Popular classes of supervised learning algorithms: SVMs, boosting, decision trees, k-NN, Naïve Bayes, genetic algorithms, neural networks, etc.

Introduction (cont'd)

- Quantification goes under different names in different fields
  1. "prevalence estimation" (in statistics and epidemiology)
  2. "class prior estimation" (in machine learning)
  3. "quantification" (in data mining)
- "relative frequency" ≡ "prevalence" ≡ "a priori probability" ≡ "prior (probability)"
- Slight differences among 1., 2., 3. are that
  - There are no training data and no classifiers in 1., while there are in 2. and 3.
  - The task is of independent interest in 1. and 3, while it is only ancillary (i.e., functional to generating better classifiers) in 2.

Introduction (cont'd)

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



Distribution drift

- Real applications may suffer from distribution drift (or "shift", or "mismatch"), defined as a discrepancy between the class distribution of Tr and that of Te
- Standard ML algorithms are instead based on the IID assumption, i.e., that training and test items are drawn from the same distribution. When using such algorithms in the presence of distribution drift, suboptimal quantification accuracy may derive.
- 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 (cont'd)

- Distribution drift is one type of concept drift, which may come in three forms:
  1. the prior probabilities $p(c_j)$ may change from training to test set
  2. the class-conditional distributions (aka "within-class densities") $p(\mathbf{x}|c_j)$ may change
  3. the posterior probabilities $p(c_j|\mathbf{x})$ may change
- It is 1. that poses a problem for quantification

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 choose B rather than A!, since A is biased
- This means that quantification should be studied as a task in its own right

Historical development

- The history of quantification research is highly non-linear, since the task has been discovered and re-discovered from within different disciplines.
- First stage : interest in the "estimation of class prevalence" from screening tests in epidemiology;
  - Earliest recorded method is (Gart & Buck, 1966)[2]
  - No training data (and no supervised learning) is involved, the role of the classifier is played by a clinical test that has imperfect "sensitivity" and "specificity"
  - Several papers appeared on epidemiology-related journals to this day

[2] Gart, J. J. & A. A. Buck: 1966, Comparison of a screening test and a reference test in epidemiologic studies: II. A probabilistic model for the comparison of diagnostic tests. *American Journal of Epidemiology* 83(3), 593–602.

Historical development (cont'd)

- Second stage : interest in the "estimation of class priors" in machine learning
  - Goal : building classifiers that are robust to the presence of distribution drift, and that are better attuned to the characteristics of the data to which they need to be applied
  - Earliest recorded method is (Vucetic & Obradovic, 2001), most influential one is (Saerens et al., 2002)
  - Several papers appeared on ML-related venues to this day
- Third stage : interest in "quantification" from data mining / text mining
  - Goal : estimating quantities and trends from unlabelled data
  - Earliest recorded work is (Forman, 2005), where the term "quantification" was coined
  - It is the applications from these fields that have provided the impetus behind the most recent wave of research in quantification

Related tasks: Prevalence Estimation from Screening Tests

- Quantification is similar to "prevalence estimation from screening tests" in epidemiology
- Screening test : a test that a patient undergoes in order to check if s/he has a given pathology
- Tests are often imperfect, i.e., they may yield
  - false positives (patient incorrectly diagnosed with the pathology)
  - false negatives (patient incorrectly diagnosed to be free from the pathology)
- Testing a patient is thus akin to classifying an item
- Main difference: a screening test typically has known and fairly constant "sensitivity" (recall) and "specificity" (1−fallout), while the same usually does not hold for a classifier

Related tasks: Density Estimation

- Quantification is similar to density estimation (e.g., estimating the prevalence of white balls in a large urn containing white balls and black balls).
- However, in traditional density estimation
  1. We can deterministically assess whether each item belongs to the class (variable c_j can be observed); in quantification this does not hold.
  2. It is impossible / economically not viable to assess class membership for each single item (e.g., we do not want to inspect every single ball in the urn); in quantification this does not hold
- Quantification is thus closely related to classification, where 1. and 2. also do not hold. However,
  - in classification the goal is correctly estimating the true class of each single individual;
  - classification is applied to individual items, and not to batches of such examples

Related tasks: Collective Classification

- A task seemingly related to quantification is collective classification (CoC), as in e.g., the classification of networked items. Similarly to quantification, in CoC the classification of an instance is not viewed in isolation of the other instances.
- However, the focus of CoC is on improving the accuracy of classification by exploiting relationships between the items to classify (e.g., hypertextual documents). CoC
  - assumes the existence of explicit relationships between the objects to classify (which quantification does not)
  - is evaluated at the individual level, rather than at the aggregate level as quantification.

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)

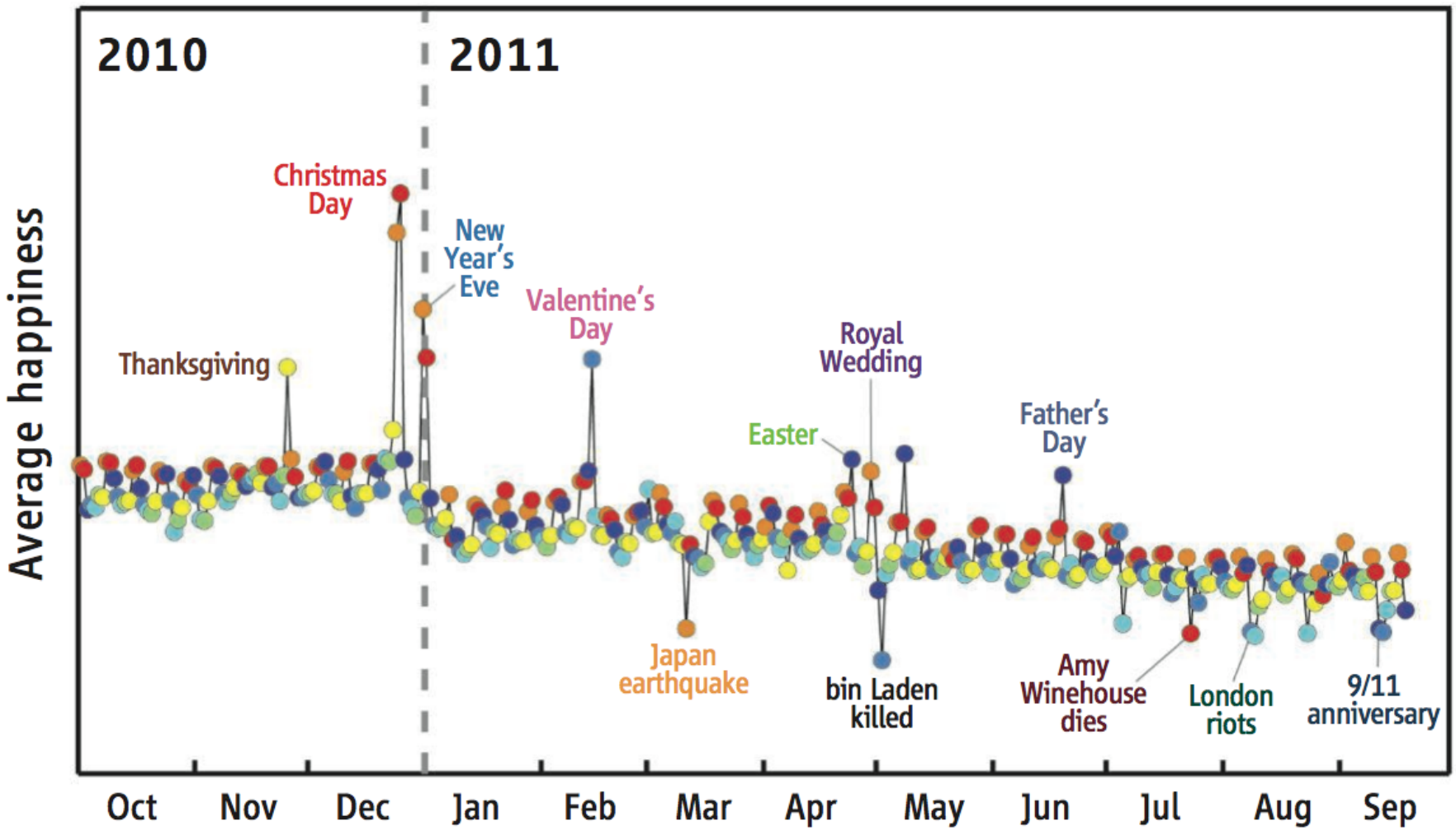
$\mathbf{x}$	vectorial representation of item x
$\mathcal{C} = \{c_1, ..., c_n\}$	set of classes
$p_S(c_j)$	true prevalence (aka "prior probability") of c_j in set S
$\hat{p}_S(c_j)$	estimated prevalence of c_j in set S
$\hat{p}_S^M(c_j)$	estimate $\hat{p}_S(c_j)$ obtained via method M
$p(c_j\|\mathbf{x})$	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

Outline

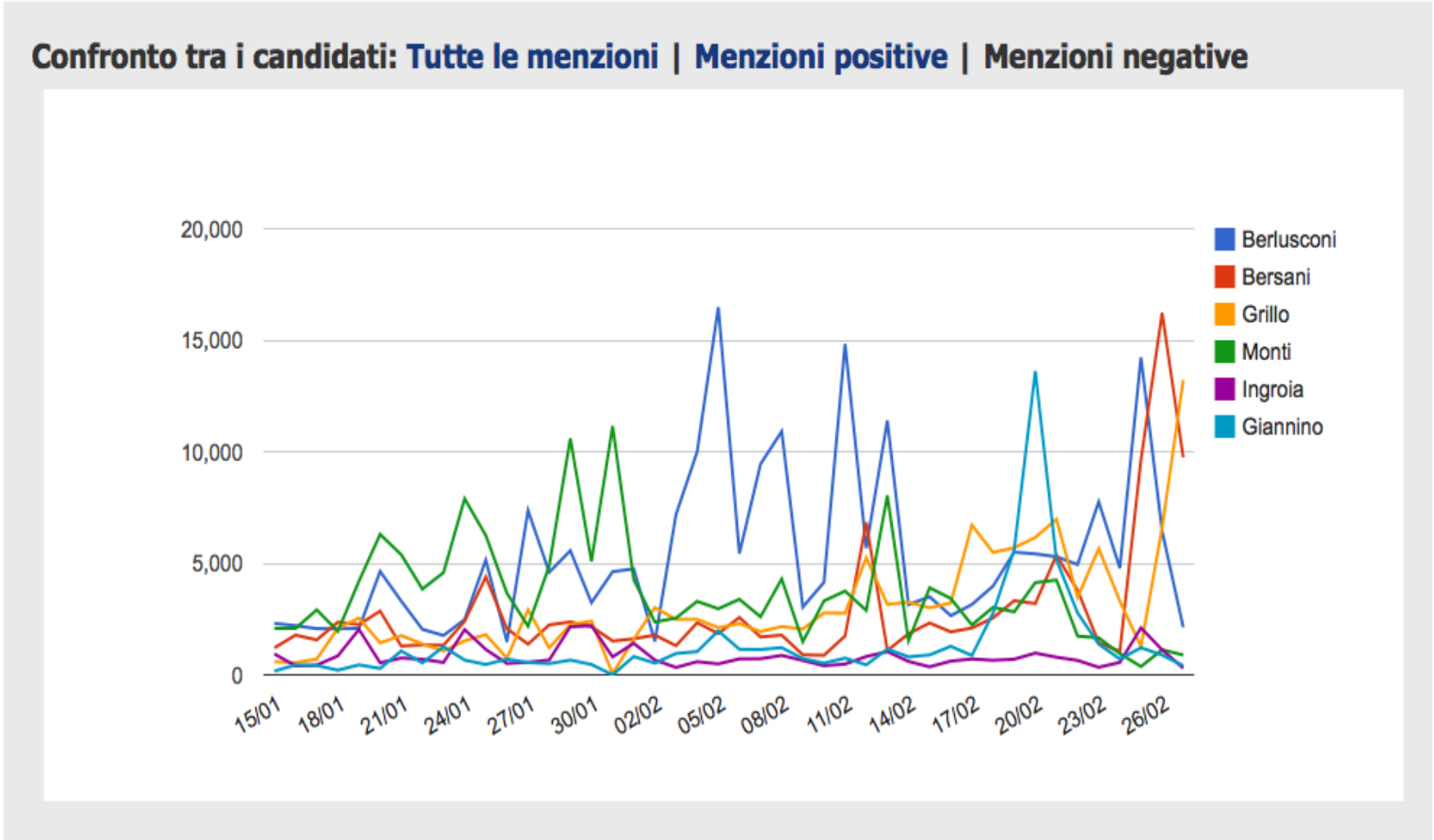
Dimensions of quantification

- 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"

Applications to the social / political sciences



Applications to the social / political sciences (cont'd)



Applications to the social / political sciences (cont'd)

- Social science is a discipline in which individual cases hardly matter, and where the goal is obtaining quantitative indicators about a population (e.g., by age group, gender, ethnic group, geographical region, time interval, ...)

  > *[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)*

- Further applications include
  - predicting election results by estimating the prevalence of blog posts (or tweets) supporting a given candidate or party[3]
  - estimate the emotional responses of the population to a natural disaster (Mandel et al., 2012)

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

[3] 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.

Applications to epidemiology

- Epidemiology : concerned with tracking the incidence of diseases across spatio-temporal contexts and across other variables (e.g., gender, age group, religion, job type, ...)
- Text quantification: Supporting epidemiological research by estimating the prevalence of clinical reports where a specific pathology is diagnosed [4]

[4] S. Baccianella, A. Esuli, F. Sebastiani: 2013, Variable-Constraint Classification and Quantification of Radiology Reports under the ACR Index. *Expert Systems and Applications* 40(9), 3441–3449.

Applications to epidemiology (cont'd)

- Quantification: Supporting epidemiology via verbal autopsies, i.e., tracking causes of death in populations where medical death certification is missing[5]
  - "verbal autopsy": estimating the cause of death from verbal symptom reports obtained from relatives (in the form of e.g., binary answers to questions about symptoms)
  - a supervised learning task: training data $(\mathbf{x}_i, y_i)$ are death records from nearby hospital in which both symptom reports obtained from caregivers $(\mathbf{x}_i)$ and medical death certification (y_i) are available
- Verbal autopsies:
  - cheaper and more effective than having physicians guess the cause of death from verbal reports
  - of crucial importance for international health policy-making and for channelling research efforts

[5] King, G. and Y. Lu: 2008, Verbal Autopsy Methods with Multiple Causes of Death. *Statistical Science* 23(1), 78–91.

Applications to market research

- Survey coding is the task of classifying natural language responses ("open-ends", aka "verbatims") elicited via open-ended questions in questionnaires[6]

- Main applications:

  1. Market Research
  2. Customer/Employee Relationship Management (CRM/ERM)
  3. Social Science
  4. Political Science (opinion polls)

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

Applications to market research (cont'd)

- Example 1 (CRM):

  "How satisfied are you with our mobile phone services?"

  Asked by: telecom company
  Class of interest: `MayDefectToCompetition`
  Goal: classification (at the individual level)

- Example 2 (MR):

  "What do you think of the recent ad for product X?"

  Asked by: MR agency
  Class of interest: `LovedTheCampaign`
  Goal: quantification (at the aggregate level)

Applications to resource allocation

- Customer support center: planning the amount of human resources to allocate to different types of issues
- Can be done by estimating the prevalence of customer calls related to a given issue [7]
- The same can be done for customer feedback obtained via email
- Important in order to improve customer support (determine priorities, track costs, plan product fixes / reengineering)

> *"rising problems can be identified before they become epidemics" (Forman, 2008)*

[7] Forman, G., E. Kirshenbaum, and J. Suermondt, Pragmatic text mining: Minimizing human effort to quantify many issues in call logs. KDD 2006, pp. 852–861.

Applications to word sense disambiguation

- Word Sense Disambiguation (WSD) is the task of determining, given the occurrence of a word w in a text, which sense of w is meant
- WSD is a text classification task, where
  - the linguistic context of the occurrence is the text
  - the different senses of the word are the classes
- Words have sense priors, i.e., different senses have different prevalences in language; WSD algorithms do exploit these priors
- The same word may have different priors in different domains; if the WSD algorithms has been trained on domain d_1, applying it on domain d_2 may yield suboptimal results
- Quantification may be used to estimate the word sense priors[8] of the new domain d_2, and use them to re-tune the classifier trained on domain d_1 (a case of domain adaptation)

[8] Chan, Y. S. and H. T. Ng, Estimating Class Priors in Domain Adaptation for Word Sense Disambiguation. ACL 2006, pp. 89–96.

(Meta-)applications to classification

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

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

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

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

  which means that in the presence of distribution drift they cannot be calibrated for both Tr and Te
- By estimating class prevalence in Te we can adjust the classifier itself so as to yield better classification accuracy[9]

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

(Meta-)applications to classification (cont'd)

- Posterior probabilities $p(c_j|\mathbf{x})$ can be re-calibrated as

$$p(c_j|\mathbf{x}) = \frac{\frac{\hat{p}_{Te}(c_j)}{p_{Tr}(c_j)} \cdot p_{Tr}(c_j|\mathbf{x})}{\sum_{c_j \in \mathcal{C}} \frac{\hat{p}_{Te}(c_j)}{p_{Tr}(c_j)} \cdot p_{Tr}(c_j|\mathbf{x})}$$

  where the $p_{Tr}(c_j|\mathbf{x})$ are the posteriors before calibration
- Also investigated for semi-supervised learning (Xue and Weiss, KDD 2009)
- Quantification is "ancillary" to classification, and not a goal in itself

Miscellaneous applications

- *Ante litteram*: In the late 1600s the Catholic Church tracked the proportion of printed texts which were non-religious
- Quantifying the proportion of damaged cells in biological samples, e.g., sperm for artificial insemination (González-Castro et al., 2013) or human tissues for oncology (Decaestecker et al., 1997)
- Measuring the prevalence of different types of pets' activity as detected by wearable devices (Weiss et al., 2013)
- Estimation of skeleton age distribution in *paleodemography*, the study of ancient human mortality, fertility, and migration (Hoppa and Vaupel, 2002)
- Real-time estimation of collective sentiment about TV shows from tweets (Amati et al., 2014)
- ...

Outline

Measures for evaluating binary + SLMC quantification

1. Absolute Error
2. Relative Absolute Error
3. Kullback-Leibler Divergence

Absolute Error

- Absolute Error (AE – sometimes called "Mean Absolute Error") is

$$AE(\hat{p}, p) = \frac{1}{|\mathcal{C}|} \sum_{c_j \in \mathcal{C}} |\hat{p}(c_j) - p(c_j)| \tag{1}$$

- Ranges between 0 (best) and

$$\frac{2(1 - \min_{c_j \in \mathcal{C}} p(c_j))}{|\mathcal{C}|}$$

(worst)

- A normalized version of AE that always ranges between 0 (best) and 1 (worst) can thus be obtained as

$$NAE(\hat{p}, p) = \frac{\sum_{c_j \in \mathcal{C}} |\hat{p}(c_j) - p(c_j)|}{2(1 - \min_{c_j \in \mathcal{C}} p(c_j))} \tag{2}$$

Absolute Error (cont'd)

- **Pros**:
  - enforces the notion that positive and negative bias are equally undesirable, and can thus be used as a general metric of Q-accuracy
  - intuitive, appealing to non-initiates too
- **Cons**:
  - predicting $\hat{p}_{Te}(c_j) = 0.01$ when $p_{Te}(c_j) = 0.02$ should be considered more serious than predicting $\hat{p}_{Te}(c_j) = 0.49$ when $p_{Te}(c_j) = 0.50$

Relative Absolute Error

- Relative Absolute Error (RAE) is

$$RAE(\hat{p}, p) = \frac{1}{|\mathcal{C}|} \sum_{c_j \in \mathcal{C}} \frac{|\hat{p}(c_j) - p(c_j)|}{p(c_j)} \tag{3}$$

- Ranges between 0 (best) and

$$\frac{|\mathcal{C}| - 1 + \frac{1 - \min_{c_j \in \mathcal{C}} p(c_j)}{\min_{c_j \in \mathcal{C}} p(c_j)}}{|\mathcal{C}|}$$

  (worst)
- A normalized version of RAE that always ranges between 0 (best) and 1 (worst) can thus be obtained as

$$NRAE(\hat{p}, p) = \frac{\sum_{c_j \in \mathcal{C}} \frac{|\hat{p}(c_j) - p(c_j)|}{p(c_j)}}{|\mathcal{C}| - 1 + \frac{1 - \min_{c_j \in \mathcal{C}} p(c_j)}{\min_{c_j \in \mathcal{C}} p(c_j)}} \tag{4}$$

Relative Absolute Error (cont'd)

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

$$p_s(c_j) = \frac{\epsilon + p(c_j)}{\epsilon|\mathcal{C}| + \sum_{c_j \in \mathcal{C}} p(c_j)} \tag{5}$$

- $\epsilon = \frac{1}{2|Te|}$ is often used as a smoothing factor.
- **Pros**:
  - all of the above, plus: relativizes to true class prevalence

Kullback-Leibler Divergence

- KLD (aka normalized cross-entropy) is [10]

$$KLD(\hat{p}, p) = \sum_{c_j \in \mathcal{C}} p(c_j) \log \frac{p(c_j)}{\hat{p}(c_j)} \tag{6}$$

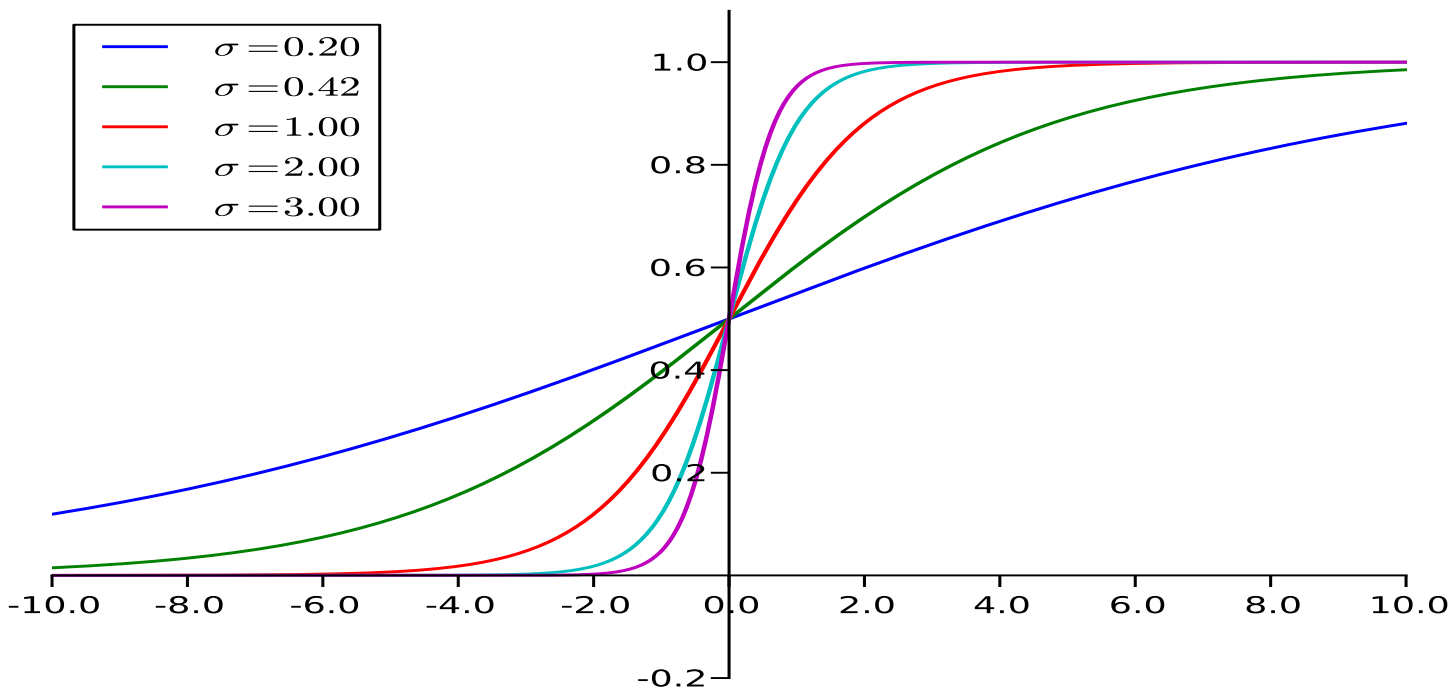
- An information-theoretic measure of the inefficiency incurred when estimating a true distribution p over a set $\mathcal{C}$ of classes by means of a predicted distribution $\hat{p}$.
- Not symmetric
- Ranges between 0 (best) and $+\infty$ (worst)

[10] Forman, G., Counting Positives Accurately Despite Inaccurate Classification. ECML 2005, pp. 564–575.

Kullback-Leibler Divergence (cont'd)

- A normalized version of KLD ranging between 0 (best) and 1 (worst) may be defined as

$$NKLD(\hat{p}, p) = \frac{e^{KLD(\hat{p},p)} - 1}{e^{KLD(\hat{p},p)}} \tag{7}$$



Kullback-Leibler Divergence (cont'd)

- **Pros**:
  - all of the above, plus: well studied within information theory and language modelling
- **Cons**:
  - hardly intuitive, difficult to explain to non-initiates ...
  - undefined when the $\hat{p}_{Te}$ is 0 for at least one class; smoothing thus needed with

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

Kullback-Leibler Divergence (cont'd)

- KLD has somehow become the "standard" measure for binary, MLMC, and SLMC quantification
- KLD is a member of the family of "f-divergences"; other such members might be appropriate measures for evaluating quantification; e.g., the Hellinger distance[11]

$$HD(\hat{p}, p) = \left(\sum_{c_j \in \mathcal{C}} \left(\hat{p}(c_j)^{\frac{1}{2}} - p(c_j)^{\frac{1}{2}}\right)^2\right)^{\frac{1}{2}}$$

[11] Víctor González-Castro, Rocío Alaiz-Rodríguez, Enrique Alegre: Class distribution estimation based on the Hellinger distance, *Information Sciences* 218 (2013), 146–164.

Measures for evaluating ordinal quantification

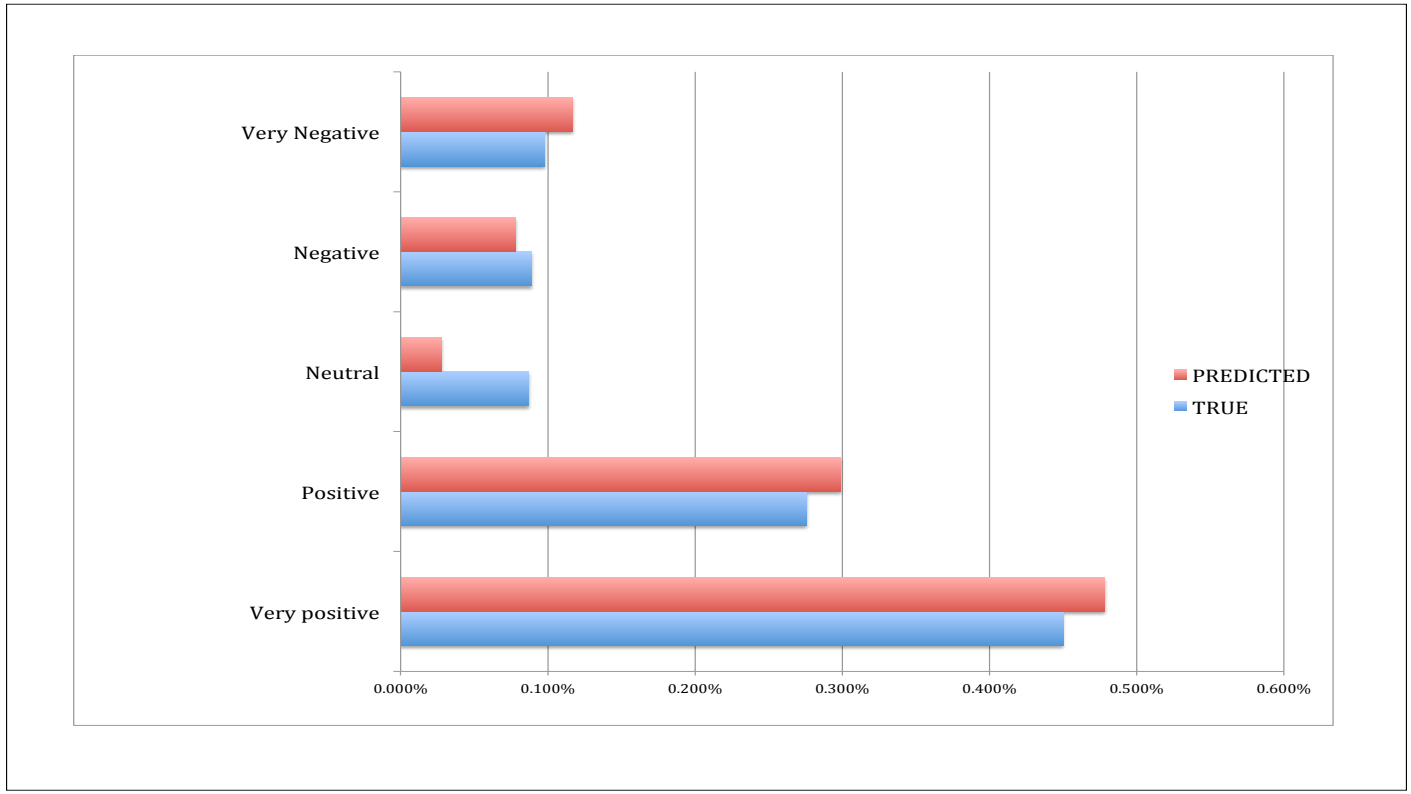
- Ordinal classification ≡ SLMC classification when there is a total order on the n classes
- Important in the social sciences, ordinal scales often used to elicit human evaluations (e.g., product reviews)
- The only known measure for ordinal quantification is the Earth Mover's Distance[12] (aka "Wasserstein metric")

$$EMD(\hat{p}, p) = \sum_{j=1}^{|\mathcal{C}|-1} |\sum_{i=1}^{j} \hat{p}(c_i) - \sum_{i=1}^{j} p(c_i)| \tag{8}$$

- The EMD may be seen as measuring the "minimum effort" to turn the predicted distribution into the true distribution, where the effort is measured by
  - the probability masses that need to be moved between one class and the other;
  - the "distance" traveled by these probability masses

[12] Esuli, A. and F. Sebastiani: 2010, Sentiment quantification. *IEEE Intelligent Systems* 25(4), 72–75.

Measures for evaluating ordinal quantification (cont'd)



- **Pros**: It works ...
- **Cons**: It is the "ordinal analogue" of absolute error ...
- Open problem: devising an "ordinal analogue" of KLD!

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
- The (Milli et al., 2013) method optimizes the multi-objective measure[13]

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

  since
  - $|FN_j - FP_j|$ is a measure of quantification error
  - $(FN_j + FP_j)$ is a measure of classification error
- By optimizing MOM we strive to keep both classification and quantification error low

  > *"it is difficult to trust a quantifier if it is not also a good enough classifier"*

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

Experimental protocols for evaluating quantification

- Standard classification datasets may be used for evaluating quantification
- Two different experimental protocols used in the literature
  - the artificial-prevalence approach (adopted by most works in the DM literature): take a standard dataset split into Tr and Te, conducting repeated experiments in which either $p_{Tr}(c_j)$ or $p_{Te}(c_j)$ are artificially varied via subsampling
    - **Pros**: class prevalence and drift may be varied at will
    - **Cons**: non-realistic experimental settings may result
  - the natural-prevalence approach (adopted in (Esuli & Sebastiani, 2015)[14]): pick one or more standard datasets that represent a wide array of class prevalences and drifts
    - **Pros**: experimental settings being tested are realistic
    - **Cons**: class prevalence and drift may not be varied at will

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

The natural prevalence approach: An example

		RCV1-v2	OHSUMED-S
ALL	Total # of docs	804,414	15,643
	# of classes (i.e., binary tasks)	99	88
	Time unit used for split	week	year
TRAINING	# of docs	12,807	2,510
	# of features	53,204	11,286
	Min # of positive docs per class	2	1
	Max # of positive docs per class	5,581	782
	Avg # of positive docs per class	397	55
	Min prevalence of the positive class	0.0001	0.0004
	Max prevalence of the positive class	0.4375	0.3116
	Avg prevalence of the positive class	0.0315	0.0218
TEST	# of docs	791,607	13,133
	# of test sets per class	52	4
	Total # of test sets	5,148	352
	Avg # of test docs per set	15,212	3,283
	Min # of positive docs per class	0	0
	Max # of positive docs per class	9,775	1,250
	Avg # of positive docs per class	494	69
	Min prevalence of the positive class	0.0000	0.0000
	Max prevalence of the positive class	0.5344	0.3532
	Avg prevalence of the positive class	0.0323	0.0209

The natural prevalence approach: An example (cont'd)

- Breaking down by drift

RCV1-v2	VLD	LD	HD	VHD	All
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
CC	2.43E-03	2.44E-03	2.79E-03	3.18E-03	2.71E-03
PCC	8.92E-03	8.64E-03	7.75E-03	6.24E-03	7.86E-03

- Breaking down by class prevalence

RCV1-v2	VLP	LP	HP	VHP	All
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
CC	2.55E-03	3.39E-03	1.29E-03	1.61E-03	2.71E-03
PCC	1.04E-02	6.49E-03	3.87E-03	1.51E-03	7.86E-03

Outline

Sampling

- The naïve method for quantification is sampling (Sa), which consists in computing $p_{Tr}(c_j)$ and taking it as an estimate of $p_{Te}(c_j)$; i.e.,

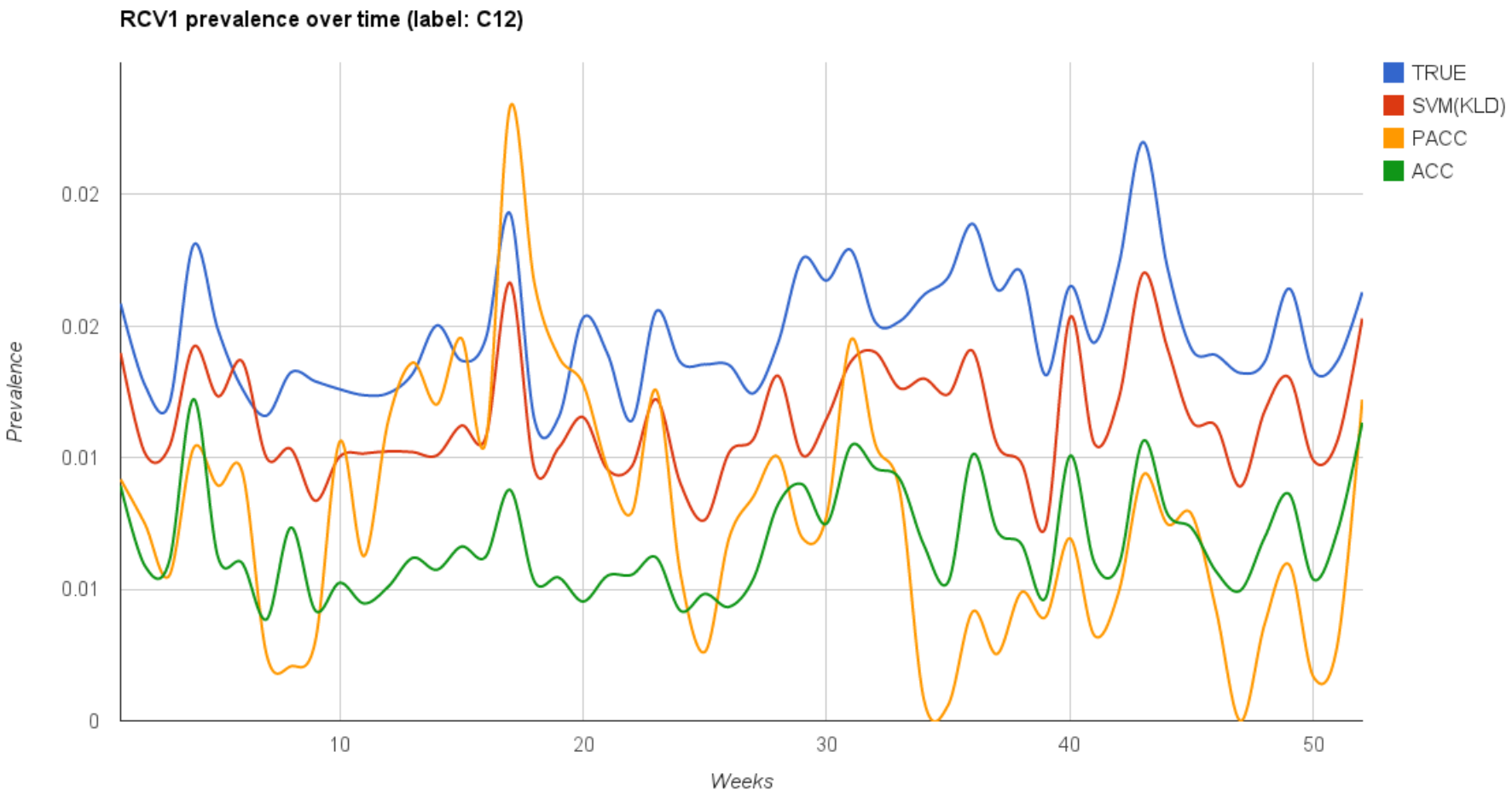
$$\hat{p}^{Sa}_{Te}(c_j) = p_{Tr}(c_j) \tag{9}$$

- Akin to always picking the majority class in classification
- Optimal in case of no distribution drift but "risky", since distribution drift is
  - ubiquitous in real-life applications
  - fundamental to application that track trends

Sampling is simply not an answer

> *"In quantification (...) you fundamentally need to assume shifts in the class priors; if the class distribution does not change, you don't need an automatic quantifier" (Forman, 2006)*

Sampling (cont'd)



Aggregative methods based on general-purpose learners

- 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)

Classify and Count

- 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.,

$$\hat{p}^{CC}_{Te}(c_j) = p_{Te}(\delta_j) \tag{10}$$

- 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) SVMs bring about an average FP/FN ratio of 0.109.

Probabilistic Classify and Count

- Probabilistic Classify and Count (PCC) is a variant of CC which estimates p_{Te} by simply counting the expected fraction of items predicted to be in the class, i.e.,[15]

$$\hat{p}_{Te}^{PCC}(c_j) = E_{Te}[c_j] = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} p(c_j|\mathbf{x}) \tag{11}$$

- The rationale is that posterior probabilities contain richer information than binary decisions, which are obtained from posterior probabilities by thresholding.
- PCC is shown to perform better than CC in (Bella et al., 2010) and (Tang et al., 2010)

[15] Bella, A., C. Ferri, J. Hernańdez-Orallo, and M. J. Ramírez-Quintana, Quantification via Probability Estimators. ICDM 2010, pp. 737–742.

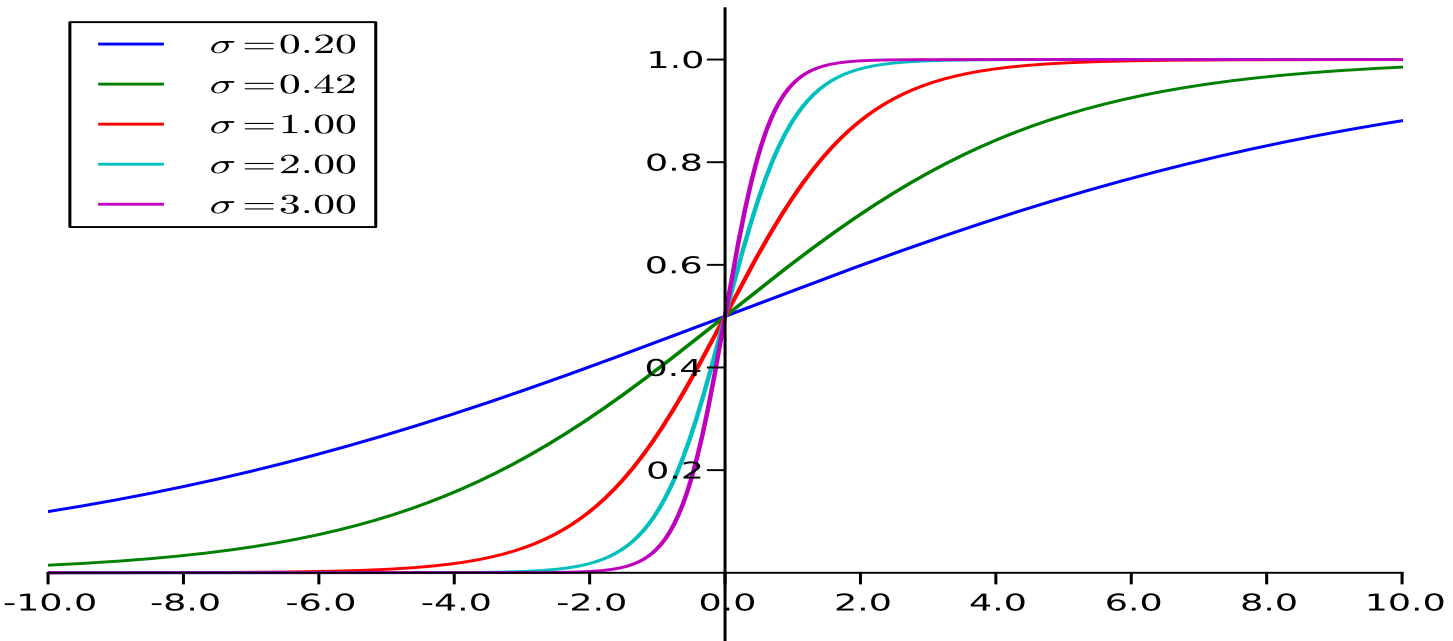
Probabilistic Classify and Count (cont'd)

- If the classifier only returns scores $s_j(\mathbf{x})$ that are not (calibrated) probabilities, the scores must be converted into calibrated probabilities, e.g., by applying a generalized logistic function

$$p(c_j|\mathbf{x}) = \frac{e^{\sigma s_j(\mathbf{x})}}{e^{\sigma s_j(\mathbf{x})} + 1} \tag{12}$$

- Calibration consists in tuning the σ parameter so that

$$p_{Tr}(c_j) = E_{Tr}[c_j] = \frac{1}{|Tr|} \sum_{\mathbf{x} \in Tr} p(c_j|\mathbf{x})$$



Probabilistic Classify and Count (cont'd)

- PCC is dismissed as unsuitable in (Forman, 2008) on the grounds that, if the $p(c_j|\mathbf{x})$ are calibrated on Tr, they can also be calibrated for Te only if there is no distribution drift ...
- Indeed, (Esuli & Sebastiani, 2015) find PCC to be worse than CC for all values of distribution drift

RCV1-v2	VLD	LD	HD	VHD	All
CC	**2.43E-03**	**2.44E-03**	**2.79E-03**	**3.18E-03**	**2.71E-03**
PCC	8.92E-03	8.64E-03	7.75E-03	6.24E-03	7.86E-03
OHSUMED-S	VLD	LD	HD	VHD	All
CC	**3.31E-03**	**3.87E-03**	**3.87E-03**	**5.43E-03**	**4.12E-03**
PCC	1.10E-01	1.07E-01	9.97E-02	9.88E-02	1.04E-01

Adjusted Classify and Count

- Adjusted Classify and Count (ACC – aka the "confusion matrix model")[16] is based on the observation that

$$p_{Te}(\delta_j) = \sum_{c_i \in \mathcal{C}} p_{Te}(\delta_j | c_i) \cdot p_{Te}(c_i) \tag{13}$$

- We do not know the $p_{Te}(\delta_j|c_i)$'s but we may estimate them on Tr via k-fold cross-validation.
- This results in a system of $|\mathcal{C}|$ linear equations with $(|\mathcal{C}| - 1)$ unknowns, i.e., the $p_{Te}(c_j)$'s. ACC consists in solving this system.

[16] Gart, J. J. and A. A. Buck: 1966, Comparison of a screening test and a reference test in epidemiologic studies: II. A probabilistic model for the comparison of diagnostic tests. *American Journal of Epidemiology* 83(3), 593–602.

Adjusted Classify and Count (cont'd)

- In the binary case this comes down to

$$\begin{aligned} p_{Te}(\delta_1) &= p_{Te}(\delta_1|c_1) \cdot p_{Te}(c_1) + p_{Te}(\delta_1|c_2) \cdot p_{Te}(c_2) \\ &= tpr_{Te} \cdot p_{Te}(c_1) + fpr_{Te} \cdot (1 - p_{Te}(c_1)) \end{aligned}$$

- If we equate c_1 with the "positive class" and c_2 with the "negative class", then

$$p_{Te}(c_1) = \frac{p_{Te}(\delta_1) - fpr_{Te}}{tpr_{Te} - fpr_{Te}} \quad (14)$$

$$\hat{p}_{Te}^{ACC}(c_1) = \frac{p_{Te}(\delta_1) - fpr_{Tr}}{tpr_{Tr} - fpr_{Tr}} \quad (15)$$

- **Cons**:
  - May return values outside [0,1], due to imperfect estimates of the $p_{Te}(\delta_j|c_i)$'s: this requires "clipping and rescaling", which is scarcely reassuring
  - Relies on the hypothesis that estimating the $p_{Te}(\delta_j|c_j)$'s via k-FCV can be done reliably, which is questionable in the presence of distribution drift

Probabilistic Adjusted Classify and Count

- Probabilistic Adjusted Classify and Count (PACC – aka "Scaled Probability Average")[17] stands to ACC like PCC stands to CC.
- It is based on the observation that (similarly to ACC)

$$E_{Te}[\delta_j] = \sum_{c_i \in \mathcal{C}} E_{Te,c_i}[\delta_j] \cdot p_{Te}(c_i) \tag{16}$$

where

$$E_{Te}[\delta_j] = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} p(c_j|\mathbf{x}) \tag{17}$$

$$E_{Te,c_i}[\delta_j] = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} p(c_j|\mathbf{x}, c_i) \tag{18}$$

The latter can be estimated via k-FCV from Tr, so PACC amounts to solving a system of $|\mathcal{C}|$ linear equations with $(|\mathcal{C}| - 1)$ unknowns

[17] Bella, A., C. Ferri, J. Hernańdez-Orallo, M. J. Ramírez-Quintana, Quantification via Probability Estimators. ICDM 2010, pp. 737–742.

Probabilistic Adjusted Classify and Count (cont'd)

- PACC is dismissed in (Forman, 2005) on the same grounds as PCC
- PACC is shown to be the best among CC, ACC, PCC, PACC in both (Bella et al., 2010) and (Tang et al., 2010)
- Results in (Esuli & Sebastiani, 2015) are more varied ...

RCV1-v2	VLD	LD	HD	VHD	All
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
CC	2.43E-03	2.44E-03	2.79E-03	3.18E-03	2.71E-03
PCC	8.92E-03	8.64E-03	7.75E-03	6.24E-03	7.86E-03

OHSUMED-S	VLD	LD	HD	VHD	All
PACC	1.90E-05	1.59E-04	1.01E-04	**4.16E-06**	7.12E-05
ACC	1.49E-05	**2.27E-05**	3.35E-05	9.16E-05	**4.08E-05**
CC	**1.23E-05**	2.88E-05	**2.60E-05**	1.15E-04	4.58E-05
PCC	5.86E-04	4.92E-04	9.21E-04	9.86E-04	7.63E-04

T50, Method X, Method Max, Median Sweep

- Forman[18] observes that ACC is very sensitive to the decision threshold of the classifier, and this may yield unreliable / unstable results for $\hat{p}^{CC}_{Te}(c_j)$; e.g., in the binary case
  - if c_1 is very infrequent, a classifier optimized for 0-1 loss may yield $tpr_{Te} \approx 0$ and $fpr_{Te} \approx 0$, which may bring to 0 the denominator of

$$\hat{p}^{ACC}_{Te}(c_1) = \frac{\hat{p}^{CC}_{Te}(c_1) - fpr_{Tr}}{tpr_{Tr} - fpr_{Tr}}$$

  - even if the denominator is not 0, it may be very small, making the result unstable
- Forman proposes to use ACC after picking, via a number of methods, a threshold different from the "natural" one and "that admits more true positives and more false positives"

[18] Forman, G., Quantifying trends accurately despite classifier error and class imbalance. KDD 2006, pp. 157–166.

T50, Method X, Method Max, Median Sweep (cont'd)

- Threshold@0.50 (T50): set the decision threshold t in such a way that tpr_{Tr} (as obtained via k-FCV) is equal to .50
  - Rationale: avoid the tail of the $1 - tpr_{Tr}(t)$ curve
- Method X (X): set the decision threshold in such a way that $fpr_{Tr} + tpr_{Tr} = 1$
  - Rationale: avoid the tails of the $fpr_{Tr}(t)$ and $1 - tpr_{Tr}(t)$ curves
- Method Max (MAX): set the decision threshold in such a way that $(tpr_{Tr} - fpr_{Tr})$ is maximized
  - Rationale: avoid small values in the denominator of (15)
- Median Sweep (MS): compute $\hat{p}_{Te}^{ACC}(c_1)$ for every threshold that gives rise (in k-FCV) to different tpr_{Tr} or fpr_{Tr} values, and take the median[19]
  - Rationale: ability of the median to avoid outliers

[19] Forman, G., Quantifying trends accurately despite classifier error and class imbalance. KDD 2006, pp. 157–166.

T50, Method X, Method Max, Median Sweep (cont'd)

- **Cons**:
  - Methods have hardly any theoretical foundation
  - Unclear that these choices return better estimates of tpr_{Te} and fpr_{Te}
  - Complete lack of relation to classification accuracy
- In the experiments of (Esuli & Sebastiani, 2015) all these methods are generally outperformed by plain ACC

RCV1-v2	VLP	LP	HP	VHP	All
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
X	3.48E-03	8.45E-03	1.32E-03	2.43E-04	4.96E-03
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

OHSUMED-S	VLP	LP	HP	VHP	All
ACC	2.37E-03	5.40E-03	**2.82E-04**	**2.57E-04**	**2.99E-03**
MAX	5.57E-03	2.33E-02	1.76E-01	3.78E-01	3.67E-02
X	**1.38E-03**	3.94E-03	3.35E-04	5.36E-03	4.44E-03
MS	3.80E-03	**1.79E-03**	1.45E-03	1.90E-02	1.18E-02
T50	7.53E-02	5.17E-02	2.71E-03	1.27E-02	2.74E-02

The Mixture Model

- The Mixture Model (MM) method consists of assuming that the distribution D^{Te} of the scores that the (binary) classifier assigns to the test examples is a mixture[20]

$$D^{Te} = p_{Te}(c_1) \cdot D^{Te}_{c_1} + (1 - p_{Te}(c_1)) \cdot D^{Te}_{c_2} \tag{19}$$

  where

  - $D^{Te}_{c_1}$ and $D^{Te}_{c_2}$ are the distributions of the scores that the classifier assigns to the examples of c_1 and c_2, respectively
  - $p_{Te}(c_1)$ is a parameter of this mixture

- MM consists of
  - estimating $D^{Te}_{c_1}$ and $D^{Te}_{c_2}$ via k-FCV
  - picking as value of $p_{Te}(c_1)$ the one that yields the best fit between the observed D^{Te} and the mixture

[20] Forman, G., Counting Positives Accurately Despite Inaccurate Classification. ECML 2005, pp. 564–575.

The Mixture Model (cont'd)

- Two variants of MM:
  - the Kolmogorov-Smirnov Mixture Model (MM(KS))
  - the PP-Area Mixture Model (MM(PP))

  differ in terms of how the goodness of fit between the left- and the right-hand side of (19) is estimated.
- **Cons**:
  - relies on the hypothesis that estimating $D_{c_1}^{Te}$ and $D_{c_2}^{Te}$ via k-FCV on Tr can be done reliably

The Mixture Model (cont'd)

- In the experiments of (Esuli & Sebastiani, 2015) both MM(PP) and MM(KS) are outperformed by plain ACC

RCV1-v2	VLP	LP	HP	VHP	All
ACC	**2.17E-03**	**1.98E-03**	**5.08E-04**	6.79E-04	**1.87E-03**
MM(PP)	1.76E-02	9.74E-03	2.73E-03	1.33E-03	1.24E-02
MM(KS)	2.00E-02	1.14E-02	9.56E-04	**3.62E-04**	1.40E-02
OHSUMED-S	VLP	LP	HP	VHP	All
ACC	**2.37E-03**	**5.40E-03**	**2.82E-04**	**2.57E-04**	**2.99E-03**
MM(PP)	4.90E-03	1.41E-02	9.72E-04	4.94E-03	7.63E-03
MM(KS)	1.37E-02	2.32E-02	8.42E-04	5.73E-03	1.14E-02

- A similar method (called "HDy") is proposed in (González-Castro et al., 2013), where the Hellinger distance is used to measure the goodness of fit.)

Iterative methods

- (Saerens et al., 2002) propose an iterative, EM-based "quantification" method for improving classification accuracy[21]
- The likelihood of the test set $Te = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$ is

$$\begin{aligned} L(Te) &= \prod_{\mathbf{x}_k \in Te} p(\mathbf{x}_k) \\ &= \prod_{\mathbf{x}_k \in Te} \sum_{c_j \in \mathcal{C}} p(\mathbf{x}_k|c_j)p(c_j) \end{aligned}$$

- Since the within-class densities $p(\mathbf{x}_k|c_j)$ are assumed constant, the idea is to determine the estimates of $p(c_j)$ that maximize $L(Te)$; these are determined via EM
- EM is a well-known iterative algorithm for finding maximum-likelihood estimates of parameters (in our case: the class priors) for models that depend on unobserved variables (in our case: the class labels)

[21] 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.

Iterative methods (cont'd)

- We apply EM in the following way until convergence of the $\hat{p}^{(s)}(c_j)$:
  - **Step 0**: For each c_j initialize $\hat{p}^{(0)}(c_j) = p_{Tr}(c_j)$
  - **Step s**: Iterate:
    - **Step s(E)**: For each test item $\mathbf{x}_k$ and each c_j compute:

$$p^{(s)}(c_j|\mathbf{x}_k) = \frac{\frac{\hat{p}^{(s)}(c_j)}{p_{Tr}(c_j)} \cdot p_{Tr}(c_j|\mathbf{x}_k)}{\sum_{c_j \in \mathcal{C}} \frac{\hat{p}^{(s)}(c_j)}{p_{Tr}(c_j)} \cdot p_{Tr}(c_j|\mathbf{x}_k)} \tag{20}$$

    - **Step s(M)**: For each c_j compute:

$$\hat{p}^{(s+1)}(c_j) = \frac{1}{|Te|} \sum_{\mathbf{x}_k \in Te} p^{(s)}(c_j|\mathbf{x}_k) \tag{21}$$

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

Iterative methods (cont'd)

- If we have an initial "guess" of the values of the $p(c_j)$'s, we can use these guesses in place of $p_{Tr}(c_j)$ in Step 0 to speed up convergence
- The method depends on the $p_{Tr}(c_j|\mathbf{x}_k)$, so these should be well "calibrated" before starting the iteration

Iterative methods (cont'd)

- (Xue & Weiss, 2009) propose a different iterative binary quantification method[22]
- Main idea : train a classifier at each iteration, where the iterations progressively improve the quantification accuracy of performing CC via the generated classifiers
  1. Initialize by training standard classifier on Tr
  2. Iterate:
     - 2.1 compute $\hat{p}_{Tr}^{CC}(c_j)$ via k-FCV;
     - 2.2 compute $\hat{p}_{Te}^{CC}(c_j)$;
     - 2.3 retrain classifier via a cost-sensitive learner
- The authors show that the cost ratio $C(fp)/C(fn)$ to be used by the cost-sensitive learner is the "distribution mismatch ratio", i.e.,

$$dmr = \frac{\frac{\hat{p}_{Tr}^{CC}(c_1)}{(1-\hat{p}_{Tr}^{CC}(c_1))}}{\frac{\hat{p}_{Te}^{CC}(c_1)}{(1-\hat{p}_{Te}^{CC}(c_1))}}$$

[22] Xue, J. C. & G. M. Weiss: 2009, Quantification and semi-supervised classification methods for handling changes in class distribution. KDD 2009, pp. 897–906.

Aggregative methods based on special-purpose learners

- Most researchers using aggregative methods have used general-purpose learning algorithms, i.e., ones optimized for classification; quantification is achieved by post-processing their results
- An alternative idea is that of using special-purpose learning algorithms optimized directly for quantification
- **Pros** :
  - Addressing quantification as a task in its own right
  - Direct optimization usually delivers better accuracy
- **Cons** :
  - Optimal classification and optimal quantification require two different learning processes which do not "mirror" each other

Methods based on explicit loss minimization

- The first aggregative method based on special-purpose learners is due to (Esuli and Sebastiani, 2015)
- The basic idea is using explicit loss minimization, i.e., using a learner which directly optimizes the evaluation measure ("loss") used for quantification
- The measures most learners (e.g., AdaBoost, SVMs) are optimized for are 0-1 loss or variants thereof.
- In case of imbalance (e.g., positives $\ll$ negatives) optimizing for 0-1 loss is suboptimal, since the classifiers tend to make negative predictions, which means $FN \gg FP$, to the detriment of quantification accuracy.
  - E.g., the experiments in the above paper report that, out of 5148 binary test sets, standard (linear) SVMs bring about an average FP/FN ratio of 0.109.

Methods based on explicit loss minimization (cont'd)

- Problem:
  - The measures 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 incurred by each test example) / univariate (i.e., each test item can be taken into consideration in isolation)
  - Evaluation measures 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)
- (Esuli and Sebastiani, 2015) thus adopt CC with the SVM for Multivariate Performance Measures (SVM_{perf}) algorithm of (Joachims, 2005)[23] tailored to optimize KLD

[23] Joachims, T. A support vector method for multivariate performance measures. ICML 2005, 377–384.

Methods based on explicit loss minimization (cont'd)

- SVM$_{perf}$ is a specialization to the problem of binary classification of SVMs for structured prediction[24], an algorithm designed for predicting multivariate, structured objects (e.g., trees, sequences, sets)
- SVM$_{perf}$ learns multivariate classifiers $h : \mathcal{X}^{|S|} \rightarrow \{-1,+1\}^{|S|}$ that classify entire sets S of instances in one shot
- **Pros**: SVM$_{perf}$ can generate classifiers optimized for any non-linear, multivariate loss function that can be computed from a contingency table (as KLD is)
- **Cons**: not SLMC-ready

[24] Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. 2004. Support vector machine learning for interdependent and structured output spaces. ICML 2004.

Methods based on explicit loss minimization (cont'd)

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

Methods based on explicit loss minimization (cont'd)

Table: Variance of KLD results across the 5148 test sets of RCV1-v2 grouped by class prevalence in Tr

RCV1-v2	VLP	LP	HP	VHP	All
SVM(KLD)	**7.52E-06**	**3.44E-06**	8.94E-07	1.56E-06	**5.68E-06**
PACC	7.58E-06	2.38E-05	1.50E-06	2.26E-07	1.29E-05
ACC	1.04E-05	7.43E-06	**4.25E-07**	4.26E-07	8.18E-06
MAX	8.61E-06	2.27E-05	1.06E-06	**1.66E-08**	1.32E-05
CC	1.79E-05	1.99E-05	1.96E-06	1.66E-06	1.68E-05
X	2.21E-05	6.57E-04	2.28E-06	1.06E-07	2.64E-04
PCC	1.75E-04	1.76E-04	3.56E-05	1.59E-04	9.38E-04
T50	2.65E-04	4.56E-04	2.43E-04	1.19E-05	3.33E-04
MM(KS)	3.65E-03	7.81E-04	1.46E-06	4.43E-07	2.10E-03
MM(PP)	4.07E-03	5.69E-04	6.35E-06	2.66E-06	2.21E-03
MS	9.36E-03	5.80E-05	1.31E-05	6.18E-06	4.61E-03

Methods based on explicit loss minimization (cont'd)

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 trees and quantification forests

- Quantification trees are special-purpose decisions trees optimized for quantification[25]; the basic idea is to use, in the learning phase, a measure of quantification as the splitting criterion at each node. Three different such measures are mentioned
  - (a proxy of) absolute error, i.e.,

$$D(\hat{p}, p) = \sum_{c_j \in \mathcal{C}} |FP - FN|$$

  - KLD
  - a "multiobjective" measure, i.e.,

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

- Quantification forests are "random forests" of quantification trees
  - Exploits the "wisdom of the crowds" effect

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

Quantification trees and quantification forests (cont'd)

- **Pros**:
  - SLMC-ready
  - Theoretically well-founded
- **Cons**:
  - Tree-based learning does not scale to large dimensionalities

Non-aggregative methods

- (King and Lu, 2008)'s method, later popularized in (Hopkins & King, 2010), does not require the classification of the individual items[26]
- The idea is to estimate class prevalences directly via

$$p_{Te}(\mathbf{x}) = \sum_{c_j \in \mathcal{C}} p_{Te}(\mathbf{x}|c_j) p_{Te}(c_j) \tag{22}$$

- If $p_{Te}(\mathbf{x})$ and $p_{Te}(\mathbf{x}|c_j)$ could be estimated, $p_{Te}(c_j)$ could be derived; but (at least in *text* quantification) $\mathbf{x}$ is too high-dimensional for the above to be reliably estimated

[26] King, G. and Y. Lu: 2008, Verbal Autopsy Methods with Multiple Causes of Death. *Statistical Science* 23(1), 78–91.

Non-aggregative methods (cont'd)

- (King and Lu, 2008) propose using subsets $s_k(\mathbf{x})$ in place of $\mathbf{x}$, i.e.,

$$p(s_k(\mathbf{x})) = \sum_{c_j \in \mathcal{C}} p(s_k(\mathbf{x})|c_j)p(c_j) \tag{23}$$

- When all individual words are chosen as subsets, this is clearly reminiscent of the independence assumption in NB classifiers
- $p(c_j)$ is estimated several times by estimating (e.g., via k-FCV) $p(s_k(\mathbf{x}))$ and $p(s_k(\mathbf{x})|c_j)$ for different choices of $s_k(\mathbf{x})$
- The average (or median) of these results is taken as the final value of $\hat{p}_{Te}(c_j)$
- A "query-biased" variant of this method is proposed in (Amati et al., 2014)

Which methods perform best?

- Different papers present different methods + use different datasets, baselines, and evaluation protocols; it is thus hard to have a precise view
- Largest experimentation to date is likely (Esuli & Sebastiani, 2015)
- No TREC-like evaluation campaign for quantification (yet?); but see SemEval 2016 Task 4 ...

	(Saerens et al. 2002)	(Forman 2005)	(Forman 2006)	(King & Lu 2008)	(Xue & Weiss 2009)	(Bella et al. 2010)	(Tang et al. 2010)	(Esuli & Sebastiani 2015)	(Barranquero et al. 2013)	(Milli et al. 2013)
CC		x	x	x		x	x	x	x	x
ACC	x	x	x		x	x	x	x	x	x
EM	o									
MM(KS)		o	x					x		
MM(PP)		o	x					x		
T50			o			x	x	x	x	
X			o					x	x	
MAX			o					x	x	
MS			o				x	x	x	
KL				o						
CDE-Iterate					o					
PCC						o	x	x		
PACC						o	x	x		
CC(SVM(KLD))								o		
CC(k-NN)									o	
ACC(k-NN)									o	
ACC(QT)										o
ACC(QF)										o

Efficiency

- Which methods are most efficient?
- Only large-scale comparison to date is (Esuli & Sebastiani, 2015), which compares 11 methods
- CC (with linear SVMs) is the fastest, SVM(KLD) is 2nd fastest
  - Reason for fast performance is the fact that they are parameter-free, no need to need to estimate parameters via k-FCV
- The other methods (PCC, ACC, PACC, T50, X, MAX, MS, MM(PP), MM(KS)) are slowed down by need to estimate parameters via k-FCV (cost becomes a function of k):
  - PCC and PACC need to estimate σ (for probability calibration)
  - ACC (and T50, X, MAX, MS) need to estimate the $p_{Te}(\delta_j|c_j)$'s
  - MM(PP) and MM(KS) need to estimate $D^{Te}_{c_1}$ and $D^{Te}_{c_2}$
- EM not tested: sense that this might be as fast as CC and SVM(KLD), although dependent on the speed of convergence

Outline

Single-label multi-class quantification

- Some of the algorithms (e.g., CC, PCC, ACC, PACC, EM, QT, QF) presented previously are SLMC-ready, since their underlying intuitions apply straightforwardly to SLMC classifiers
- Some other algorithms (e.g., T50, X, MAX, MS, MM, SVM(KLD)) are binary in nature; while this is suboptimal, SLMC can be done by
  - doing binary quantification ("one against the rest", i.e., c_j vs. $\mathcal{C}/c_j$) for each $c_j \in \mathcal{C}$
  - rescaling the results so that they sum up to 1
- Open problem: devise SLMC variant of SVM(KLD)

Ordinal quantification

- Some of the algorithms (e.g., CC, PCC, ACC, PACC) presented previously are OC-ready, since their underlying intuitions apply straightforwardly to OC classifiers (train an OC classifier, and adjust its CC class estimates)
- For some other algorithm (e.g., EM – let alone those algorithms for which not even a SLMC version is available, e.g., SVM(KLD)) "ordinal equivalents" are not trivial to devise

Networked data quantification

- Networked data quantification is quantification when the individual unlabelled items are linked to each other[27]
- This is the Q-equivalent of collective classification, which leverages both
  - endogenous features (e.g., textual content)
  - exogenous features (e.g., hyperlinks and/or labels of neighbouring items)
- For performing "collective quantification" we may use a collective classification algorithm and then correct the resulting prevalence estimates via ACC or other

[27] Tang, L., H. Gao, and H. Liu: 2010, Network Quantification Despite Biased Labels. MLG 2010, pp. 147–154.

Networked data quantification (cont'd)

- (Tang et al., 2010) propose a non-aggregative method for this correction based on observing that

$$p(i^k) \quad = \quad p(i^k|c_1) \cdot p_{Te}(c_1) + p(i^k|c_2) \cdot (1 - p_{Te}(c_1)) \quad (24)$$

  where $p(i^k)$ is the probability that a node has a directed path of length k into node i
- It follows that

$$p_{Te}(c_1) \quad = \quad \frac{p(i^k) - p(i^k|c_2)}{p(i^k|c_1) - p(i^k|c_2)} \quad (25)$$

- $p(i^k)$ can be observed in the data, while $p(i^k|c_1)$ and $p(i^k|c_2)$ can be estimated from a training set
- A value $\hat{p}_{Te}^{(i,k)}(c)$ is obtained for each (i, k). All estimates for $k \in [1, k_{max}]$ are computed and the median is used as the final estimate $\hat{p}_{Te}(c)$

Quantification in data streams

- One of the major application modes for quantification: important for real-time monitoring, early detection of epidemics, of market and ecosystem evolution, of endangered species, etc.
- Granularity is an important issue when quantifying across time[28]
- Need to bin the timestamped data and treat each bin as a separate test set
- Problem: select the best bin width
  - If too granular, the size of the sample may become too small, estimates may become unreliable, curves become jagged
  - If too coarse, changes over time become less apparent
- Solution: use a sliding window to aggregate cases from adjacent bins into each test set

[28] Forman, G.: 2008, Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery* 17(2), 164–206.

Outline

1. Introduction
2. Applications of quantification in IR, ML, DM, NLP
3. Evaluation of quantification algorithms
4. Supervised learning methods for quantification
5. Advanced topics
6. Resources + shared tasks
7. Conclusions

Software resources for quantification

- Andrea Esuli and Fabrizio 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.
- Wei Gao and Fabrizio Sebastiani. Tweet Sentiment: From Classification to Quantification. *Proceedings of the 6th ACM/IEEE International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2015)*, Paris, FR, 2015. Contains links to quantification software.
- 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
  - Register and participate!

Conclusion

- Quantification: a relatively (yet) unexplored new task, with lots of low-hanging fruits to pick
- 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
`fsebastiani@qf.org.qa`