Predicting Classifier Accuracy in the Wild

Fabrizio Sebastiani

Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
fabrizio.sebastiani@isti.cnr.it

Joint work with
Lorenzo Volpi, Andrea Esuli, Alejandro Moreo

Artificial Intelligence Doctoral Academy

Pisa, IT – March 26, 2024

Predicting classifier accuracy

- Classifier Accuracy Prediction (CAP)
  - Routinely performed via k-fold cross-validation (k-FCV)
  - Reliable only when the training data T and the unlabelled data U are IID
  - Still an open problem when T and U are not IID, i.e., when dataset shift (DS – aka "dataset drift") is present
- One of several tasks of interest that tackle DS, among which
  - Estimating the type of DS at play
  - Estimating the amount of DS at play
  - Adapting classifiers to DS
- Useful
  - "How is my old classifier going to perform on these new data?"
  - "Should I obtain new labels for retraining?"
  - Important for responsible use of AI
- QuAcc, a new method for CAP under DS

Dataset Shift

- $P_1(X, Y)$: source distribution (from which the training data T are sampled)
- $P_2(X, Y)$: target distribution (from which the unlabelled data U are sampled)

Independently and Identically Distributed Data	Dataset Shift
$P_1(X, Y) = P_2(X, Y)$	$P_1(X, Y) \neq P_2(X, Y)$

Causes of Dataset Shift

- Variations in the environment that the data represent (real shift); i.e. the environment is not stationary, and the operating ("test") conditions were not the same at training time;
  - E.g., prevalence of terrorism-related news before or after 9/11;
- Misrepresentation of the environment on the part of the data (virtual shift): i.e., the process of labelling training data may have introduced "sample selection bias":
  - intentionally (e.g., when oversampling the minority class)
  - unintentionally (e.g., if active learning is used)

Factorizing the Joint Probability Distribution

- $P(X, Y)$ can be written as $P(Y|X)P(X)$
  - Factorization useful in "$X \rightarrow Y$ problems" (causal learning, i.e., inferring phenomena Y from causes X)

Schölkopf B., Janzing D., Peters J., Sgouritsa E., Zhang K., Mooij JM. On causal and anticausal learning. In ICML 2012.

Factorizing the Joint Probability Distribution

- $P(X, Y)$ can be written as $P(Y|X)P(X)$
  - Factorization useful in "$X \rightarrow Y$ problems" (causal learning, i.e., inferring phenomena Y from causes X)
- $P(X, Y)$ can be written as $P(X|Y)P(Y)$
  - Factorization useful in "$Y \rightarrow X$ problems" (anticausal learning, i.e., inferring phenomena Y from symptoms X)

Schölkopf B., Janzing D., Peters J., Sgouritsa E., Zhang K., Mooij JM. On causal and anticausal learning. In ICML 2012.

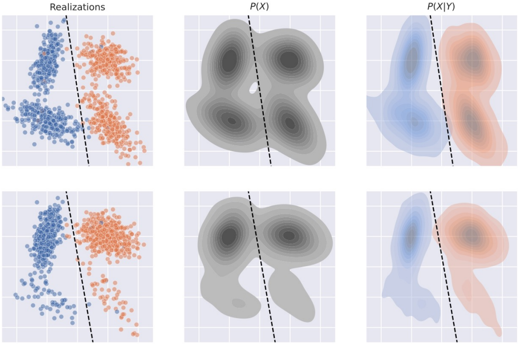
Factorizing the Joint Probability Distribution

- $P(X, Y)$ can be written as $P(Y|X)P(X)$
  - Factorization useful in "$X \rightarrow Y$ problems" (causal learning, i.e., inferring phenomena Y from causes X)
- $P(X, Y)$ can be written as $P(X|Y)P(Y)$
  - Factorization useful in "$Y \rightarrow X$ problems" (anticausal learning, i.e., inferring phenomena Y from symptoms X)
- Three major types of DS identified in the literature, depending on whether we are in the presence of causal learning or anticausal learning
  - Covariate shift (in $X \rightarrow Y$ problems)
  - Prior probability shift (in $Y \rightarrow X$ problems)
  - Concept shift (in both types of problems)

Schölkopf B., Janzing D., Peters J., Sgouritsa E., Zhang K., Mooij JM. On causal and anticausal learning. In ICML 2012.

Covariate Shift

- Context: "$X \to Y$ problems"
  - Causal learning, i.e., inferring phenomena Y from causes X
  - $P(X, Y)$ decomposed as $P(Y|X)P(X)$
- E.g., weather forecasting, avalanche forecasting

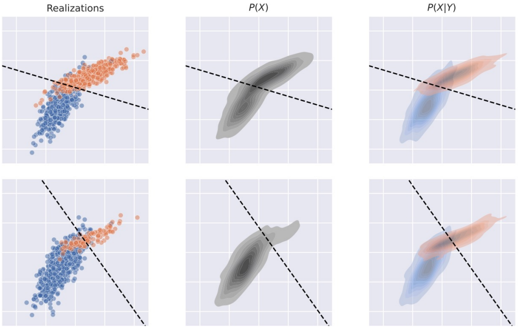


- Covariate shift defined as situation in which
  - $P_1(Y|X) = P_2(Y|X)$
  - $P_1(X) \neq P_2(X)$
- E.g., forecasting for different geographical areas

Prior Probability Shift

- Context: "$Y \to X$ problems"
  - Anticausal learning, i.e., inferring phenomena Y from symptoms X
  - $P(X, Y)$ decomposed as $P(X|Y)P(Y)$
- E.g., handwritten digit recognition, authorship attribution, predicting illnesses from symptoms

- Prior probability shift (aka "label shift") defined as situation in which
  - $P_1(X|Y) = P_2(X|Y)$
  - $P_1(Y) \neq P_2(Y)$
- E.g., digit recognition for binary digits only

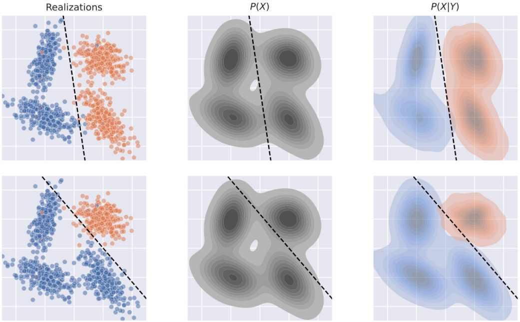


Concept Shift

- Context: either "$X \to Y$ problems" or "$Y \to X$ problems"

- Concept shift defined as situation in which one of
  - $P_1(Y|X) \neq P_2(Y|X)$
  - $P_1(X|Y) \neq P_2(X|Y)$

  holds
- E.g., perception of what counts as "positive" changes



Classifier Accuracy Prediction under DS

- A number of CAP methods emerged in the last few years, but SOTA still unsatisfactory
  - CAP error sometimes too high for practical applicability
  - Experimentation sometimes not thorough enough
  - Most methods devised for / tested on one CA measure only (vanilla accuracy)

Garg, S., Balakrishnan, S., Lipton, Z. C., Neyshabur, B., and Sedghi, H. (2022). Leveraging unlabeled data to predict out-of-distribution performance. In ICLR 2022.

Goel, K., Sohoni, N. S., Poms, F., Fatahalian, K., and Ré, C. (2021b). Mandoline: Model evaluation under distribution shift. In ICML 2021.

Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., and Schmidt, L. (2021). Predicting with confidence on unseen distributions. In ICCV 2021.

Elsahar, H. and Gallé, M. (2019). To annotate or not? Predicting performance drop under domain shift. In EMNLP-IJCNLP 2019.

QuAcc: A New Method for CAP under DS

- QuAcc: "Quantification for Accuracy Prediction"
  - Independent of the learning algorithm used for training the classifier
  - Independent of the CA measure chosen
- Standard setting for CAP:
  - Domain $\mathcal{X}$ of items, set $\mathcal{Y} = \{y_1, ..., y_n\}$ of classes
  - Training set $T \sim P_1(X, Y)$ assumed unavailable
  - Validation set $V \sim P_1(X, Y)$ available
  - Unlabelled set $U \sim P_2(X, Y)$ available
  - $P_1(X, Y) \neq P_2(X, Y)$
  - Goal: predict the accuracy that a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$ trained on T will have on U, according to an accuracy measure A
- Equivalently, one may assume that no validation set V is available but the original training set T is still available

QuAcc: A New Method for CAP under DS

- Observation #1: Any CA measure $A(h, U)$ can be computed from the contingency table C^U obtained by applying h to U, so we only need to estimate the values c_{ij}^U of each cell in C^U

		Predicted class				
		y_1	...	y_i	...	y_n
True class	y_1	c_{11}	...	c_{1j}	...	c_{1n}
	...	...	...	...	...	...
	y_i	c_{i1}	...	c_{ij}	...	c_{in}
	...	...	...	...	...	...
	y_n	c_{n1}	...	c_{nj}	...	c_{nn}

QuAcc: A New Method for CAP under DS

- $\Rightarrow$ Idea #1:
  1. View the cells of the contingency table C^U as classes
  2. Train on V a model that estimates the values c_{ij}^U
  3. Use the estimates $\hat{c}_{ij}^U$ to predict $A(h, U)$
- For Step 2 we represent the datapoints as pairs $(\ddot{\mathbf{x}}, \ddot{y})$, where
  - $\ddot{\mathbf{x}}$ is a vector

    $$\ddot{\mathbf{x}} = (\mathbf{x}, \Pr(y_1|\mathbf{x}), ..., \Pr(y_n|\mathbf{x}))$$

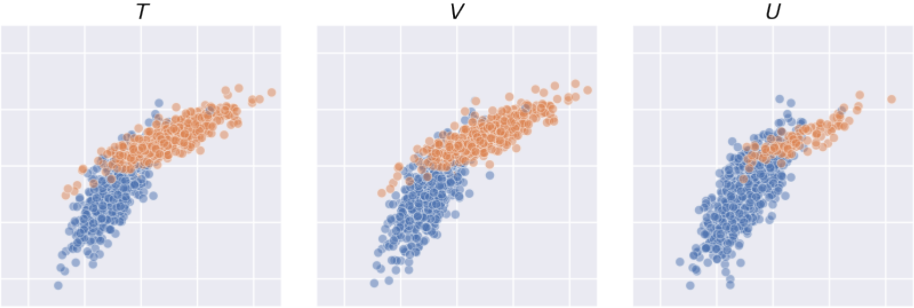
    which incorporates
    - the original representation $\mathbf{x}$ that classifier h has used
    - the posterior probabilities $\Pr(y_i|\mathbf{x})$ that h has returned for $\mathbf{x}$
  - $\ddot{y}$ is a label that ranges not on $\mathcal{Y}$ but on C^U
- For datapoints in V, $\ddot{y}$ encodes the pair $(y, h(\mathbf{x}))$ consisting of (i) the true class of $\mathbf{x}$ and (ii) the class that h has predicted for it
- For datapoints in U, $\ddot{y}$ is unknown, since we know $h(\mathbf{x})$ but not y

QuAcc: A New Method for CAP under DS

- Observation #2: We don't strictly need to predict the cell where each datapoint will end, we only need to predict the counts (or the frequencies) of datapoints that will end in each cell
- ⇒ Idea #2: Use quantification methods to estimate these frequencies
  - Quantifiers:
    - predictors of the fractions of datapoints that belong to each class
    - robust to DS "by design"
  - Goal: improving over simplistic "classify and count" via special-purpose learning techniques

QuAcc: A New Method for CAP under DS

- Focus of the present work:
  - Predicting the accuracy of binary classifiers
  - Prior probability shift



1st QuAcc variant: The 1×4 method

- The 1×4 method is based on training on V a single multiclass (4 classes) quantifier q that estimates the frequencies of the four cells

		Pred $\oplus$	Pred $\ominus$
True	$\oplus$	TP	FP
True	$\ominus$	FN	TN

- Basic process:
  1. Classify instances of V using h, to obtain $\ddot{V} = \{(\ddot{\mathbf{x}}_i, \ddot{y}_i) \mid (\mathbf{x}_i, y_i) \in V\}$
  2. Classify instances of U using h, to obtain $\ddot{U} = \{\ddot{\mathbf{x}}_i \mid \mathbf{x}_i \in U\}$
  3. Train a multiclass (4 classes) quantifier q on $\ddot{V}$
  4. Apply q to $\ddot{U}$ to obtain

$\hat{p}_U(\text{TP})$	$\hat{p}_U(\text{FP})$
$\hat{p}_U(\text{FN})$	$\hat{p}_U(\text{TN})$

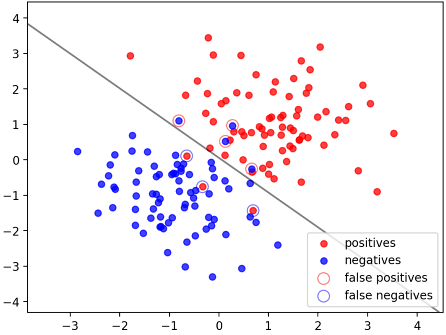
- Multiclass version: the $1\times n^2$ method

2nd QuAcc variant: The 2×2 method

- Observation: sets TP ∪ FN and TN ∪ FP are known
- The 2×2 method is based on training on V two binary quantifiers, i.e.,
  - one that discriminates between classes TP and FN
  - one that discriminates between classes TN and FP
- Advantageous over the 1×4 method since
  - Exploits additional knowledge
  - Implements "divide et impera"
- Multiclass version: the $n \times n$ method

3rd QuAcc variant: The 1×3 method

- Observation: The FP's and the FN's tend to lie in two regions that both flank (from opposite sides) the separating surface
- Since they are contiguous, they may be viewed as a single region FP ∪ FN



- The 1×3 method is based on training on V a single multiclass (3 classes) quantifier q that discriminates among TP, TN, and FP ∪ FN
- Caveat: Can only be used for CA measures that do not differentiate between FP and FN. E.g.,
  - Yes: vanilla accuracy, F_1
  - No: cost-sensitive CA measures

QuAcc: Additional covariates

- One can add to the $\ddot{\mathbf{x}} = (\mathbf{x}, \Pr(y_1|\mathbf{x}), ..., \Pr(y_n|\mathbf{x}))$ vector a number of covariates that make explicit information only implicitly present in the vector
- The vector thus becomes

$$\ddot{\mathbf{x}} = (\mathbf{x}, \Pr(y_1|\mathbf{x}), ..., \Pr(y_n|\mathbf{x}), \mathrm{MC}(\mathbf{p}), \mathrm{NE}(\mathbf{p}), \mathrm{MIS}(\mathbf{p}))$$

with

MaxConf $$\mathrm{MC}(\mathbf{p}) = \max_{y_i \in \mathcal{Y}} \Pr(y_i|\mathbf{x})$$

NegEnt $$\mathrm{NE}(\mathbf{p}) = \sum_{y_i \in \mathcal{Y}} \Pr(y_i|\mathbf{x}) \log \Pr(y_i|\mathbf{x})$$

Max Inverse Softmax $$\mathrm{MIS}(\mathbf{p}) = \max_{y_i \in \mathcal{Y}} \left[\log \Pr(y_i|\mathbf{x}) - \frac{1}{|\mathcal{Y}|} \sum_{y_j \in \mathcal{Y}} \log \Pr(y_j|\mathbf{x}) \right]$$

A few experiments

- Experiments aimed at simulating PPS by using the APP protocol
  - Meant to be a "stress test" for robustness to PPS, since it simulates a wide variety of amounts
    - of training data imbalance
    - of test data imbalance
    - of PPS
  - APP extracts from a dataset Ω
    - training samples
    - validation samples
    - test samples

    with class frequencies lying on a pre-specified grid
  - Implements the $P_1(Y) \neq P_2(Y)$ and $P_1(X|Y) = P_2(X|Y)$ conditions
- Experiments using two multiclass quantification methods (SLD and KDEy) that are SOTA for addressing PPS

Saerens, M., Latinne, P., and Decaestecker, C. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. Neural Computation, 2002.

Moreo, A., González, P., and del Coz, J. J. Kernel density estimation for multiclass quantification. arXiv:2401.00490 [cs.LG], 2023.

Experimental Setup

Datasets	• IMDB, CCAT, GCAT, MCAT
Classifier h	• Logistic Regression
Quantifier q	• (CC,) SLD, KDEy
$T \sim P_1(X, Y)$	• For training classifier h • Training samples extracted according to 9-point grid of class frequencies
$V \sim P_1(X, Y)$	• For training quantifier q • Validation samples with same frequencies as training samples
$U \sim P_2(X, Y)$	• Test samples extracted according to 21-point grid of class frequencies • 100 samples per frequency • Samples of 1000 datapoints each

Experimental Setup

- Baselines: we use all CAP methods published in the last 5 years that make their code available, i.e.,
  - ATC (Garg et al., ICLR 2022)
  - DoC (Guillory et al., ICCV 2021)
  - Mandoline (Chen et al., ICML 2021)
  - RCA and RCA* (Elsahar and Gallé, EMNLP-IJCNLP 2019)
- Classifier accuracy measure: we use
  - vanilla accuracy
  - F_1
- CAP error measure: we use

$$\mathrm{Err}(A(h,U),\hat{A}(h,U)) = |A(h,U) - \hat{A}(h,U)|$$

Optimisation

- Optimised QuAcc obtained via model selection
  - Via grid search over set of hyperparameters
  - By minimising $\mathrm{Err}(A(h, V), \hat{A}(h, V))$
- Optimise 1×4 method, 2×2 method, and 1×3 method, by exploring
  - C, rebalance ← classifier underlying quantification method
  - recalibrate (via BCTS) ← SLD
  - bandwidth ← KDEy
- Additional model selection to choose best optimised method

Overall Results for Vanilla Accuracy

		IMDB	CCAT	GCAT	MCAT
Baselines	Naïve	.1799 ± .208	.1274 ± .165	.1071 ± .150	.1183 ± .164
	RCA	.1085 ± .116	.0493 ± .056	.0519 ± .051	.0618 ± .076
	RCA*	.1099 ± .118	.0564 ± .073	.0504 ± .049	.0610 ± .076
	Mandoline	.3616 ± .280	.1920 ± .176	.1581 ± .161	.3599 ± .346
	DoC	.0226 ± .020	.0145 ± .013‡	.0112 ± .010	.0199 ± .022†
	ATC	.0613 ± .078	.0292 ± .036	.0151 ± .017	.0230 ± .030
QuAcc	QuAcc(CC)	.0474 ± .038	.0297 ± .024	.0201 ± .015	.0313 ± .042
	QuAcc(SLD)	**.0162 ± .013**	.0151 ± .014	.0097 ± .007	**.0136 ± .010**
	QuAcc(KDEy)	.0167 ± .018	**.0137 ± .012**	**.0090 ± .008**	.0143 ± .013‡
Error reduction		+28.32%	+5.52%	+19.64%	+31.66%

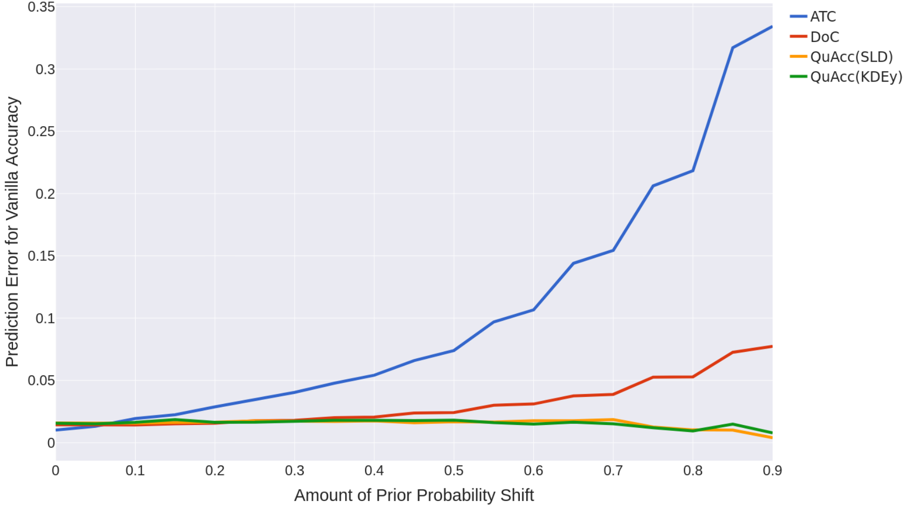
- Each figure is the average value of E across the $9 \times 21 \times 100 = 18{,}900$ combinations of a training sample T_i and a test sample U_j.

Overall Results for F_1

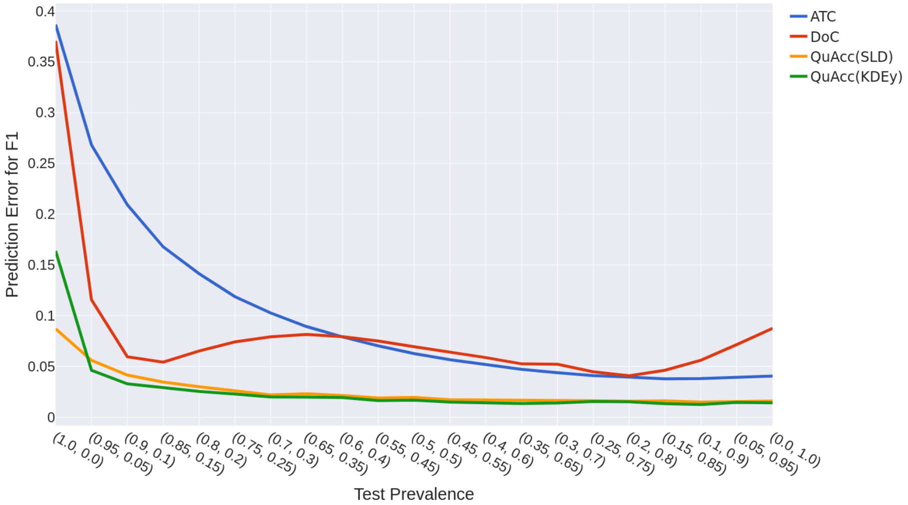
		IMDB	CCAT	GCAT	MCAT
Baselines	Naïve	.1512 ± .226	.1327 ± .229	.1288 ± .225	.1300 ± .223
	RCA	.1204 ± .148	.1008 ± .134	.1058 ± .147	.1147 ± .151
	RCA*	.2085 ± .237	.2593 ± .284	.2347 ± .238	.2486 ± .250
	Mandoline	—	—	—	—
	DoC	<u>.0809 ± .096</u>	.0951 ± .109	.0947 ± .126	.0911 ± .123
	ATC	.1015 ± .133	<u>.0798 ± .116</u>	<u>.0918 ± .141</u>	<u>.0765 ± .124</u>
QuAcc	QuAcc(CC)	.0640 ± .091	.0499 ± .082	.0470 ± .094	.0492 ± .092
	QuAcc(SLD)	**.0259** ± **.038**	**.0199** ± **.032**	**.0221** ± **.051**	**.0227** ± **.055**
	QuAcc(KDEy)	.0292 ± .066	.0201 ± .051	.0240 ± .067	.0302 ± .080
Error reduction		+67.99%	+75.06%	+75.93%	+75.08%

- Each figure is the average value of E across the $9\times21\times100$=18,900 combinations of a training sample T_i and a test sample U_j.

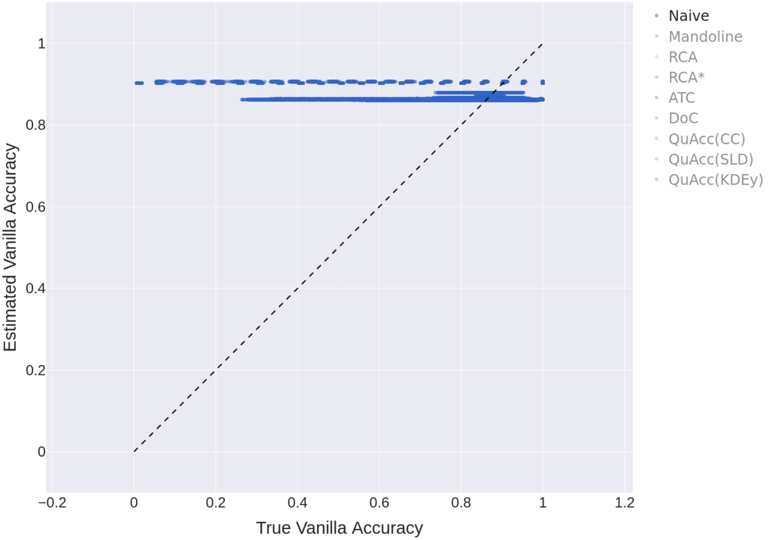
Vanilla Accuracy on IMDB (as a function of PPS)



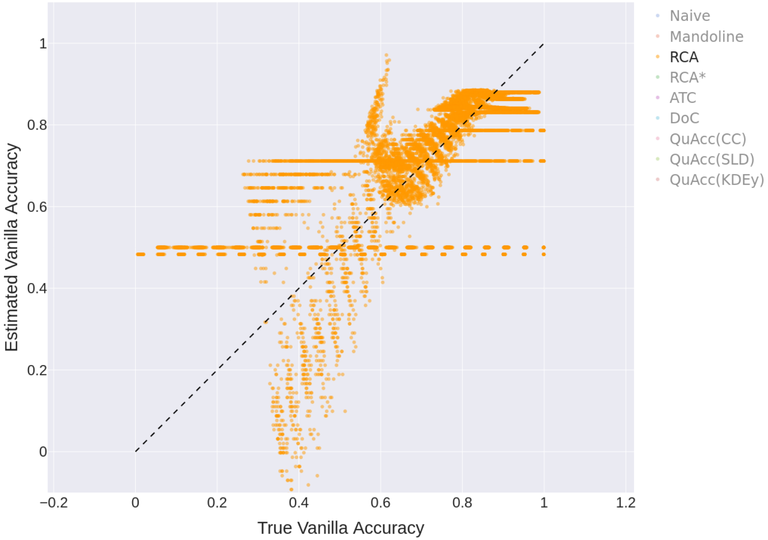
F_1 on IMDB (as a function of test prevalence)



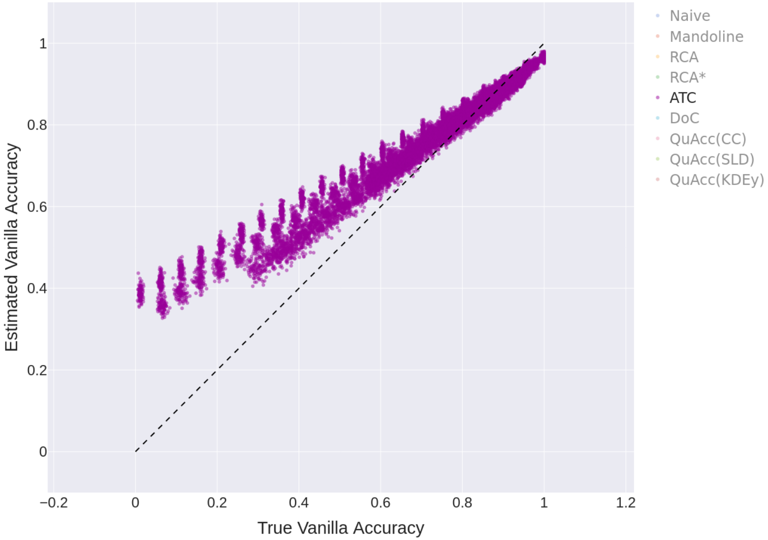
Vanilla Accuracy on IMDB: Naive



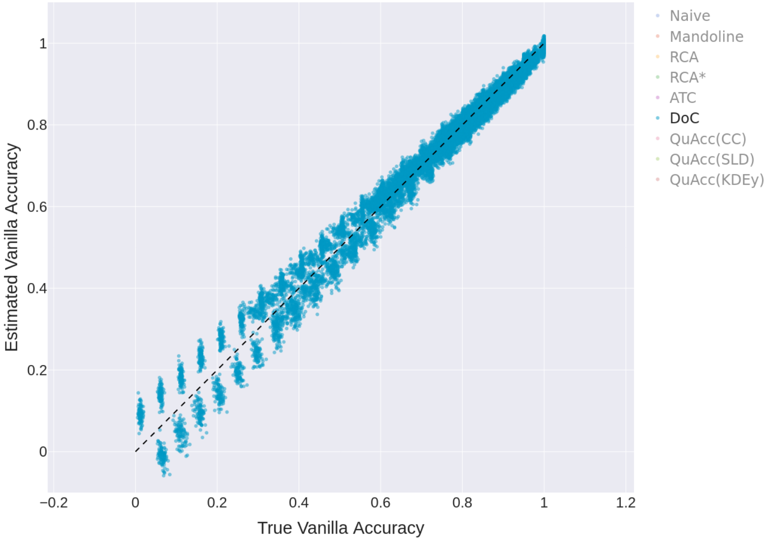
Vanilla Accuracy on IMDB: RCA



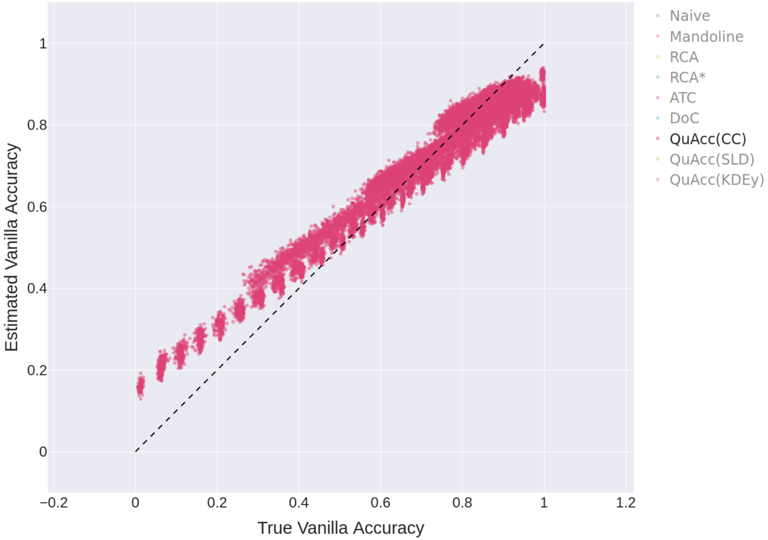
Vanilla Accuracy on IMDB: ATC



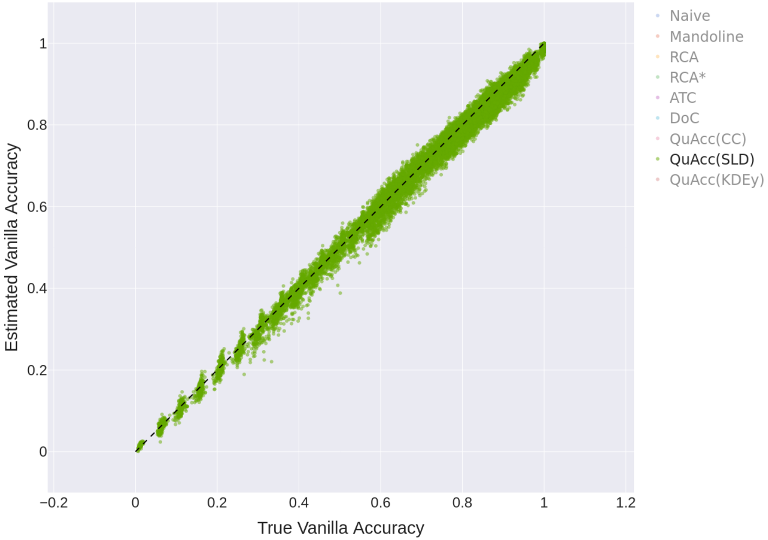
Vanilla Accuracy on IMDB: DoC



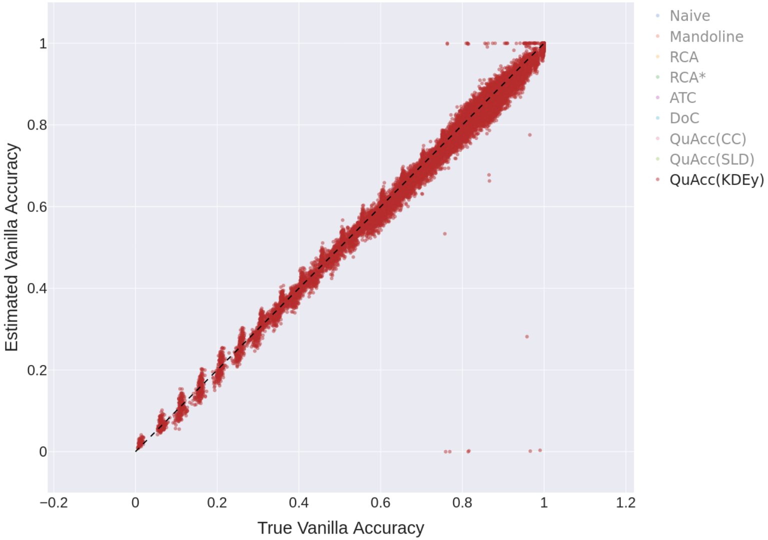
Vanilla Accuracy on IMDB: QuAcc(CC)



Vanilla Accuracy on IMDB: QuAcc(SLD)



Vanilla Accuracy on IMDB: QuAcc(KDEy)



Lessons learned

- QuAcc always outperforms all the baselines
  - Large error reduction with Acc, very large error reduction with F_1
  - Robust to a “difficult” accuracy measure such as F_1
- Even the simplistic QuAcc(CC) gives good results
  - $\Rightarrow$ Viewing contingency table cells as classes is a good idea
- Best performance obtained by QuAcc(SLD) or QuAcc(KDEy)
  - $\Rightarrow$ Using PPS-robust quantification algorithms for predicting the values of these cells is a good idea
- CAP under PPS can be performed reasonably well, with average CAP error
  - $< 2\%$ for Acc
  - $< 3\%$ for F_1

What remains to be done

- New variants
  - Generate ensemble out of the 3 variants via stacking
- Testing QuAcc on different classifier-learning techniques
  - E.g., deep learning methods
- Testing QuAcc on multiclass classification
  - Devise analogues of the 1×3 method
- Adapting QuAcc to PPS between training data and validation data
- Testing QuAcc on other types of dataset shift
  - Involves choosing quantification methods robust to DS types other than PPS

Thank you!

Email: fabrizio.sebastiani@isti.cnr.it