UNIVERSITY OF WARSAW

WYDZIAŁ MATEMATYKI, INFORMATYKI I MECHANIKI

MIM

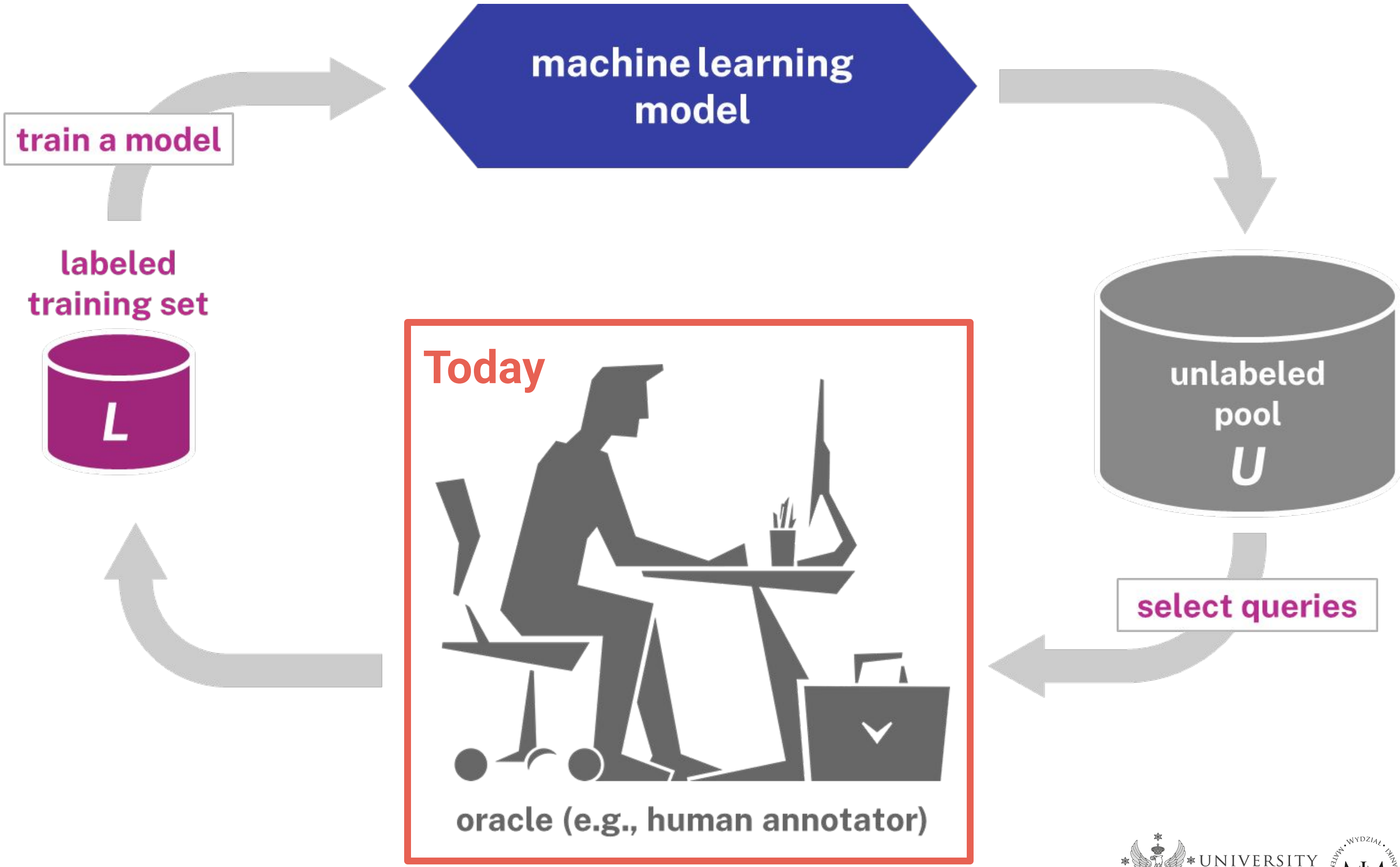
Active Learning - practical considerations and open problems

Andrzej Janusz
Daniel Kałuża

THE PLAN

- A recap of the previous lectures.
- Dealing with a faulty Oracle.
- Estimating usefulness of experts.
- Optimization of query assignments.
- Application examples and experiments.
- Summary.

The active learning cycle - the Oracle



The initial data batch

- The initial batch has huge impact on the active learning performance.
  - Random sampling.
  - Iterative sampling using the representativeness-diversity function.
  - Clustering-based sampling.
- Random samples are always needed for the evaluation!
- <u>We will focus on this problem in a different lecture!</u>

Faulty Oracle

- Some cases might be difficult to label even for domain experts.
  - Imagine a group of medical doctors discussing a difficult case.
  - People get distracted and tired over time.
- Some Oracles might not be "composed" of real experts.
  - Crowdsourcing the labeling task.
- How can we detect and deal with wrong or suspicious data labels?

Image: Freepik.com

What can we do?

- Redundancy in data labeling.
  - Each query is shown to a few <u>independent</u> experts.
  - The ground truth can be decided by voting.
  - It increases the total amount of labeling that needs to be done by experts…
- Data consistency checks.
  - Are there any similar cases from the same/other classes?
- The diagnostic of a trained model.
  - The use of XAI tools can help in flagging cases with “suspicious” labels.

Dealing with noisy data labels

- Strategies for the redundant labeling:
  - There is no “the best” method - the right approach strongly depends on a particular application.
  - Queries need to be selected in batches.
- The “query push” and “query pull” approaches:
  - Sets of queries are created and “pushed” for each expert.
    - It balances the workload.
    - A new iteration of AL cycle starts when all labeling is done.
    - The query sets may overlap.
  - One que of queries is created with duplicated entries.
    - Experts “pull” queries asynchronously from the que.
    - It makes sense to start a new AL cycle before the que depletes.

Reaching a consensus

- Majority voting.
  - The ground truth is decided by voting.
  - Each sample needs to be labeled by at least two (three) experts.
- Hierarchical verification of labels.
  - A part of labels is double-checked by “senior” annotators.
- Weighted voting.
  - Reliability scores are given to experts and used as weights.

Image: Freepik.com

Iterative algorithm for reaching a consensus

- Each expert is characterized by two vectors - true positive rates (TPR_ℓ) and true negative rates (TNR_ℓ) for each possible label ℓ.
  - Initially, all *TPR*s are set to *1/L* and *TNR*s to *(1 - 1/L)*, where *L* is the number of possible labels.

Step 1. *Temporary ground truths* are obtained by weighted voting. The weights are assigned using *TPR*s and *TNR*s.

Step 2. *TPR*s and *TNR*s are updated by comparing labels from each expert to the *temporary ground truths*.

- Steps 1 and 2 are repeated until the algorithm converges.
  - Output: the ground truths and expert reliability estimates

An experiment - reasoning from noisy labels

- Label counts per example: $S_1, \dots, S_N \sim Cat(s)$
- Label probabilities: $P_{i,c} \sim Uniform(0, 1)$
- True labels for each example: $L_{i,c} \mid P_{i,c}, S_i = 1 \text{ if } P_{i,c} \geq P_i^{S_i}$
- Probabilities of assigning a task to experts: $A_1, \dots, A_K \sim Beta(a_1, a_2)$
- Samples assigned to each expert: $U_{i,1}, \dots, U_{i,N} \sim Bernoulli(A_i)$
- Whether an expert is reliable: $G_1, \dots, G_K \sim Bernoulli(p_{good})$
- *TPR* (true positive rate) for each expert: $TP_i \mid G_i = g \sim Beta(\beta^g_{tpr}, \gamma^g_{tpr})$
- *TNR* (true negative rate) for each expert: $TN_i \mid G_i = g \sim Beta(\beta^g_{tnr}, \gamma^g_{tnr})$
- Finally, sample user labelings:

$$UL_{u,i,j} \mid L_{i,j} = 1, TP_u \sim Bernoulli(TP_u)$$

$$UL_{u,i,j} \mid L_{i,j} = 0, TN_u \sim Bernoulli(1 - TN_u)$$

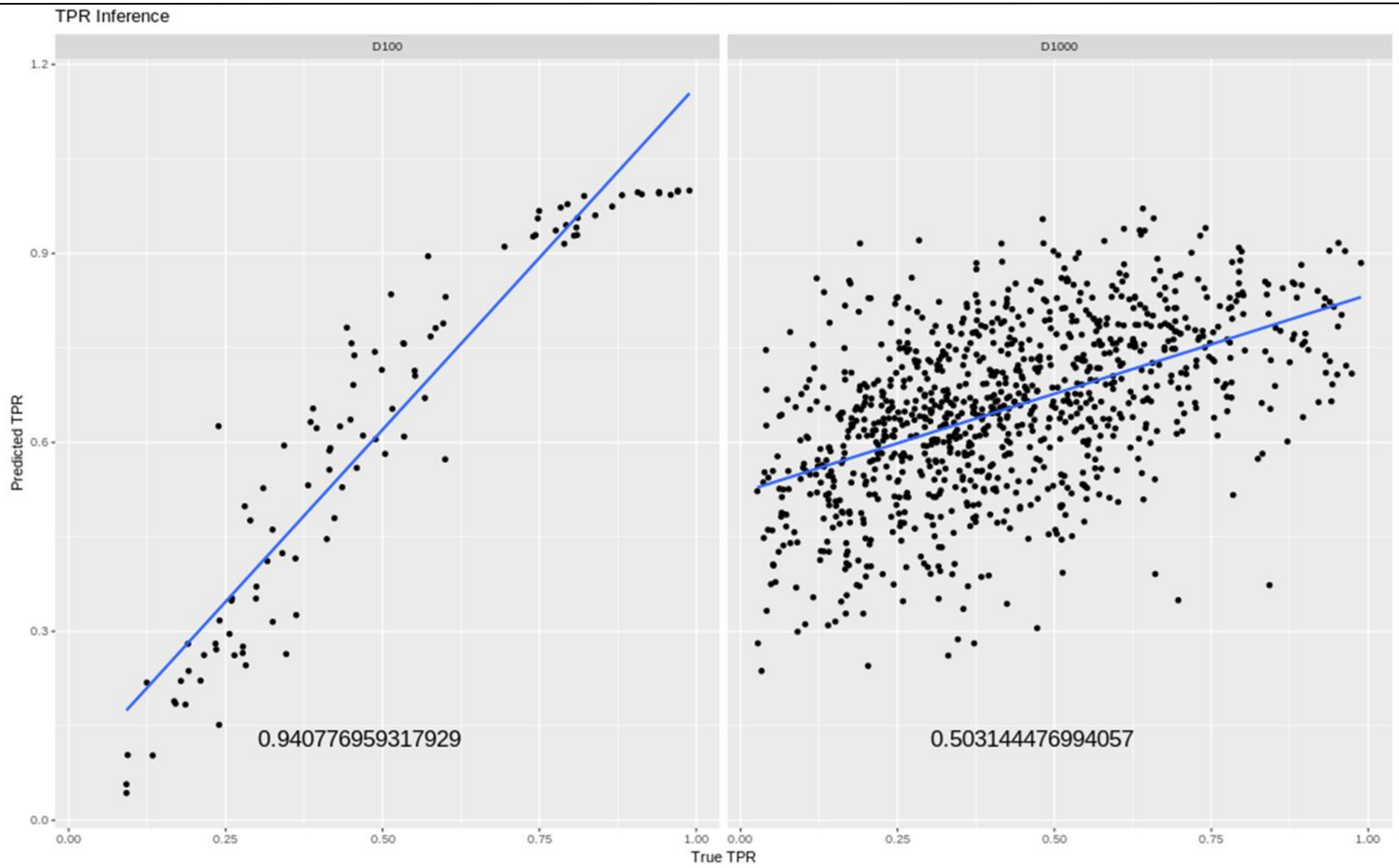
Experimental setup

Data Set	D100	D1000
Difficulty	Moderate	Hard
Number of Samples	10000	10000
Number of Experts	100	1000
Means Samples per User	400	40
Expected Good Users	25 (25%)	50 (5%)
Expected TPR	Good - 0.8 / Bad - 0.4	Good - 0.8 / Bad - 0.4
Expected TNR	Good - 0.9 / Bad - 0.7	Good - 0.9 / Bad - 0.7

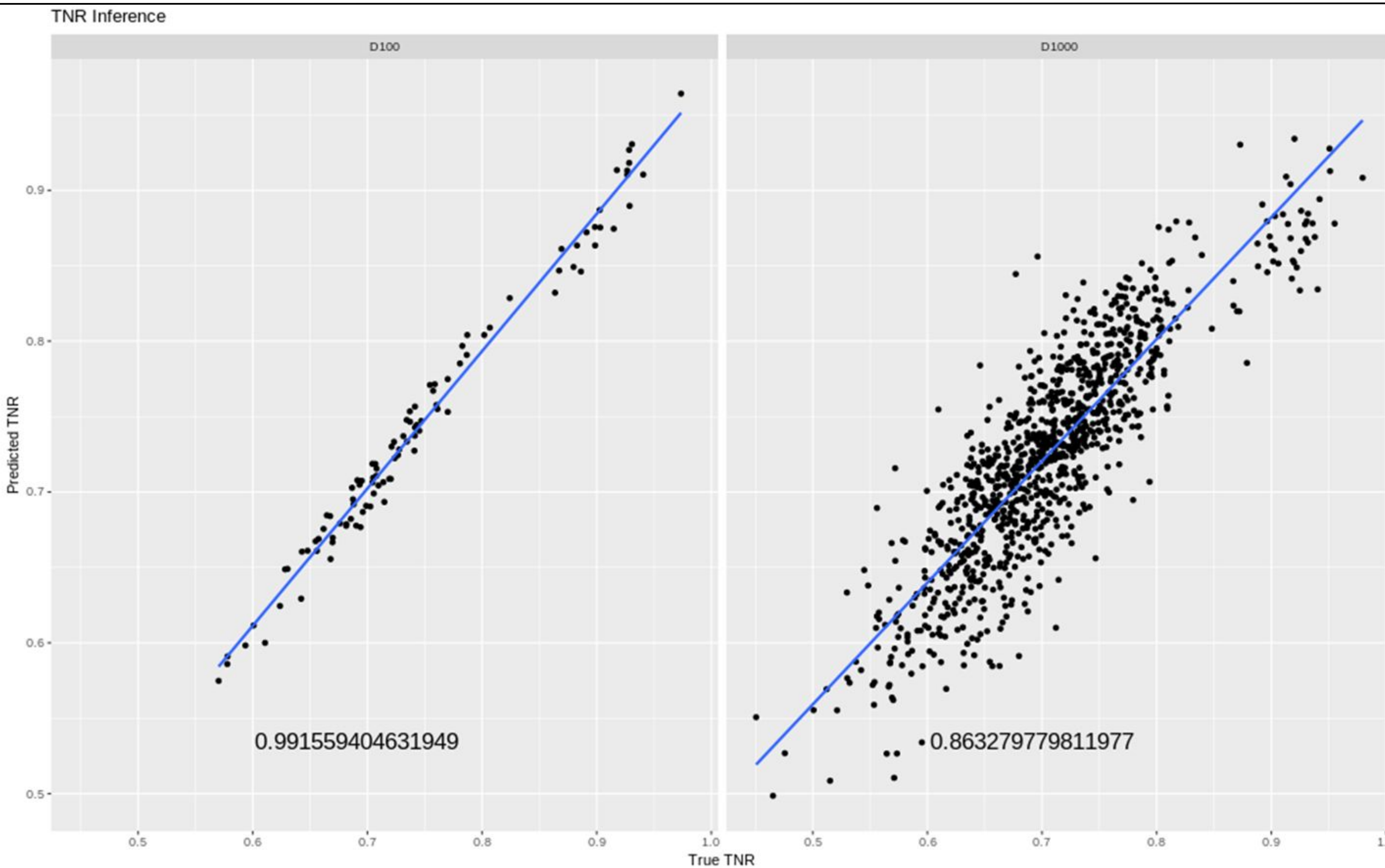
Results - the quality of ground truth estimation

Data set / Method	Accuracy	Precision	Recall	F1 Score
D100 / MV	0.834	0.434	**0.398**	0.415
D100 / EM	**0.889**	**0.731**	0.393	**0.511**
D1000 / MV	0.796	0.307	**0.296**	0.301
D1000 / EM	**0.825**	**0.359**	0.278	**0.313**

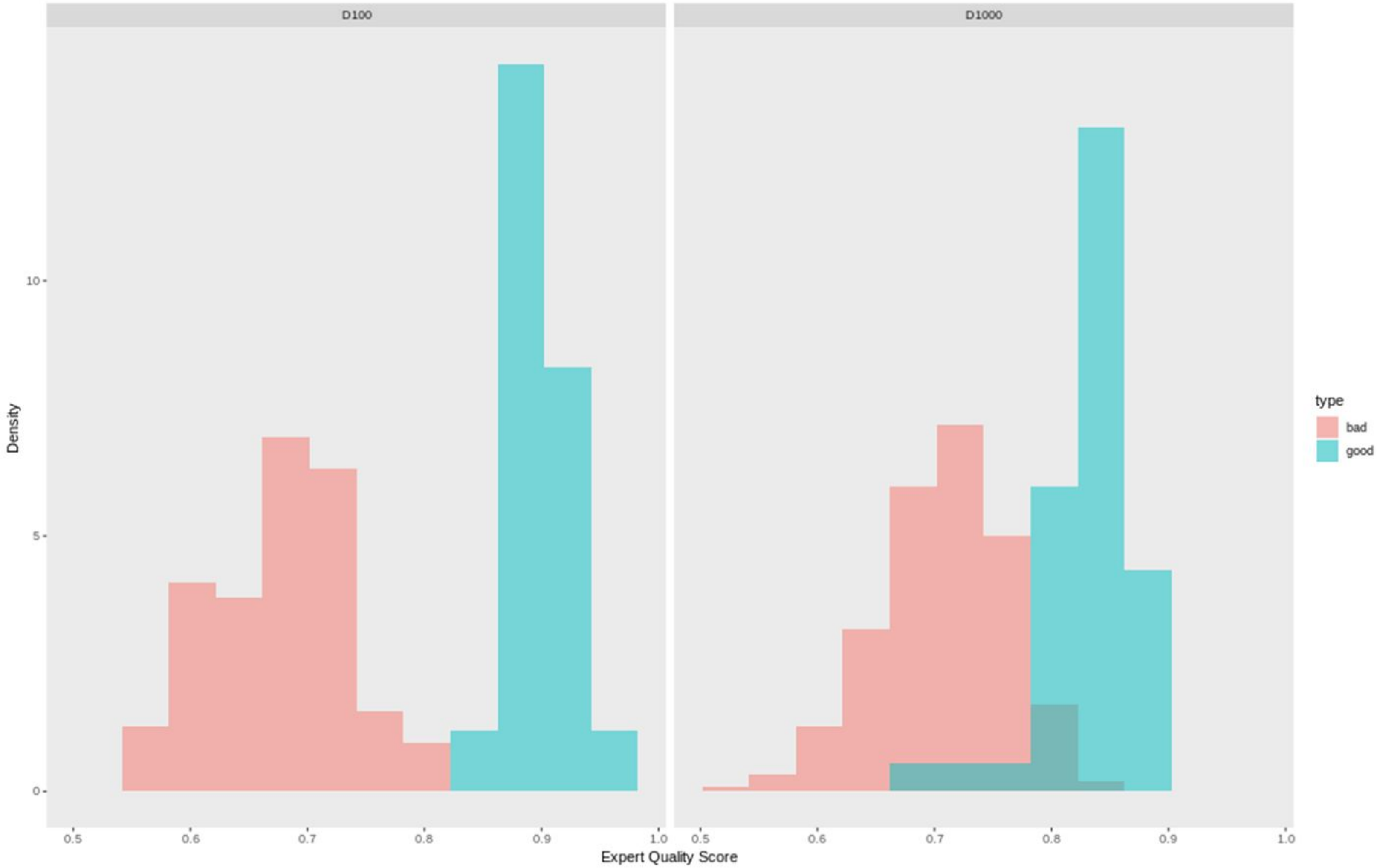
Results - the estimation of *TPRs*



Results - the estimation of *TNR*s



Results - the identification of reliable experts



An alternative approach

- Ideally, we would like to estimate expert's reliability and get the ground truth labels without annotating the same sample more than once.
  - Surrogate ML models approximate votes of each expert.
  - ML voters have to be perfectly consistent with expert annotations and their voting weights should be lower.
- Limitations:
  - The number of annotated samples per each expert and class needs to be large enough.
  - Experts have to be reasonably reliable.

Other ideas for dealing with noisy labels?

- When crowdsourcing the annotation task, we need to closely monitor the reliability of labels.
  - Assigning control queries from a predefined (known) pool.
  - High redundancy of acquired labels is advisable.
  - Means of protection against adversarial labels are necessary.
    - Monitoring of IP addresses.
    - Anti-spam protection (e.g., CAPTCHA).
- Providing a communication platform for the experts.
- Repeating the same queries a few times at different timepoints to check the labeling consistency.

Smart assignment of queries

- The estimation of expert reliability allows to optimize the query assignment:
  - We may assign a query to an expert with the highest expectation of assigning the correct label…
  - or to an expert who we believe assigned correct labels to similar queries in the past.
  - Experts availability - the workload control.

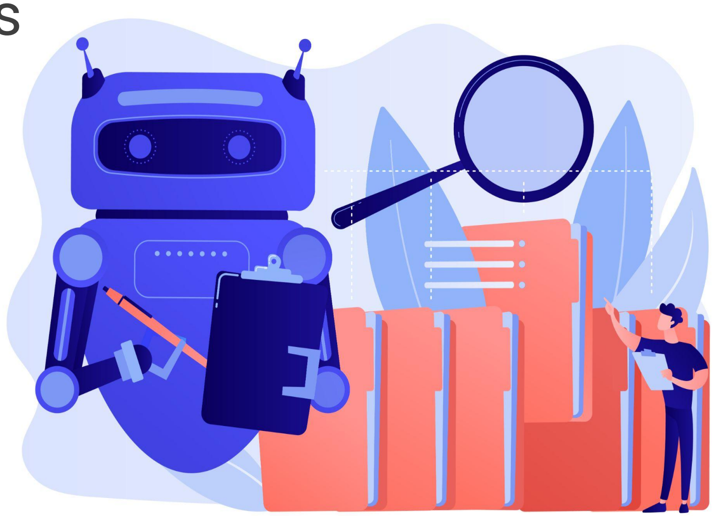


Image: Freepik.com

- Does the optimization of query assignment biases the estimation of expert reliability?
  - It definitely does.
  - <u>The optimization of the labeling process is an open problem!</u>

Design of an experiment

- For the purpose of experiments, we model experts using prediction algorithms:
  - Independent data is used to train the experts models.
  - For each "expert", data is biased in a different way to express various specializations and skills.
  - The expected quality of experts is estimated in advance.
- Queries are assigned to experts with the highest expectation of assigning the correct label (given the prediction of the model used for the query selection).
  - Significant improvement of the label quality.
  - Doesn't work well in combination with the assessment of the reliability of experts…

Variable labeling costs?

- The active learning objective can be modified to minimize the overall labeling cost.
  - Samples have the cost proportional to their “size”, e.g. number of words in a document, length of a recording.
  - The cost of labeling is expressed in the same currency as the cost of misclassification.
- The labeling costs may be predefined or approximated.
  - E.g., we may try to predict the expected labeling time.
- The labeling cost may depend on a particular annotator.
- <u>Many open problems remain!</u>

Unusual query types

- Examples of queries for structured data:
  - Selecting images for the segmentation task.
  - Selecting phrases for the named entity recognition task.
- Active semi-supervised learning.
  - Each query is composed of a pair of samples - we ask if they are similar.
  - Can be more intuitive for experts.
- Active class selection.
- Active feature value acquisition.

Image: Freepik.com

Stopping criteria for active learning

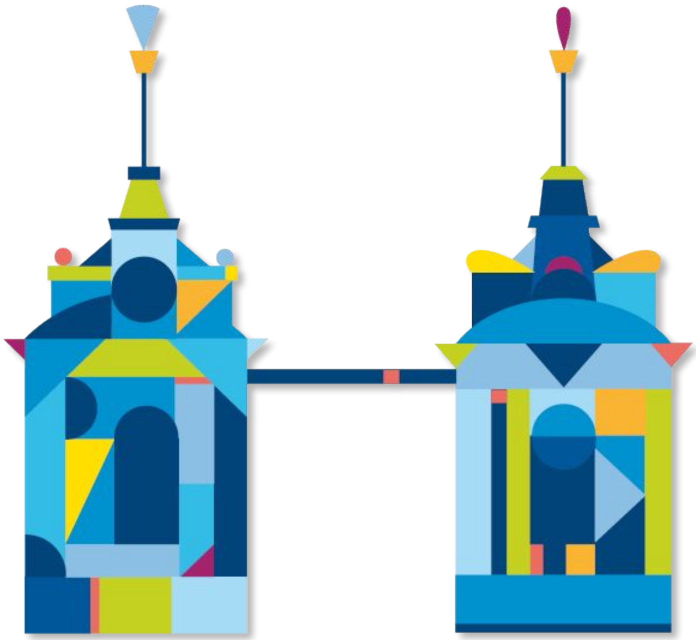
- Active learning usually stops when the labeling costs exceed the gains resulting from the expected model improvements.
  - Our querying budget ends.
  - Model improvement in a few consecutive iterations is lower than a predefined threshold.
- The life-long learning setup:
  - The learning never stops.
  - Suitable for detecting new classes or dealing with the concept drift.

Summary

- We discussed the basic principles of active learning.
- We considered three different active learning application scenarios, with their pros and cons.
- We talked about the informativeness of instances in the context of AL and its relation to the uncertainty of the learner.
- We analyzed a few AL algorithms and application examples for different real-world ML tasks.

Literature:

1. B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, (2010).
2. R. Monarch. Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-centered AI. Simon and Schuster, (2021).
3. S. Vijayanarasimhan, K. Grauman. Multi-level active prediction of useful image annotations for recognition. In Advances in Neural Information Processing Systems (NIPS), volume 21, pages 1705–1712. MIT Press, (2009).
4. S. Vijayanarasimhan, K. Grauman. What's it going to cost you? Predicting effort vs. informativeness for multi-label image annotations. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Press, (2009).
5. B. Settles, M. Craven, L. Friedland. Active learning with real annotation costs. In Proceedings of the NIPS Workshop on Cost-Sensitive Learning, pages 1–10, (2008).
6. V. S. Sheng, C. X. Ling. Feature value acquisition in testing: A sequential batch test algorithm. In Proceedings of the International Conference on Machine Learning (ICML), pages 809–816. ACM Press, (2006).



QUESTIONS OR COMMENTS?

a.janusz@mimuw.edu.pl

or

d.kaluza@mimuw.edu.pl