UNIVERSITY OF WARSAW

WYDZIAŁ MATEMATYKI, INFORMATYKI I MECHANIKI

Active Learning - the introduction

Andrzej Janusz
Daniel Kałuża

THE PLAN

- Introduction
- Active Learning basics
- A discussion of various Active Learning scenarios
- A deeper dive into the membership query synthesis and stream-based selective sampling
- Exemplary algorithms and use-cases
- Summary

Active Learning

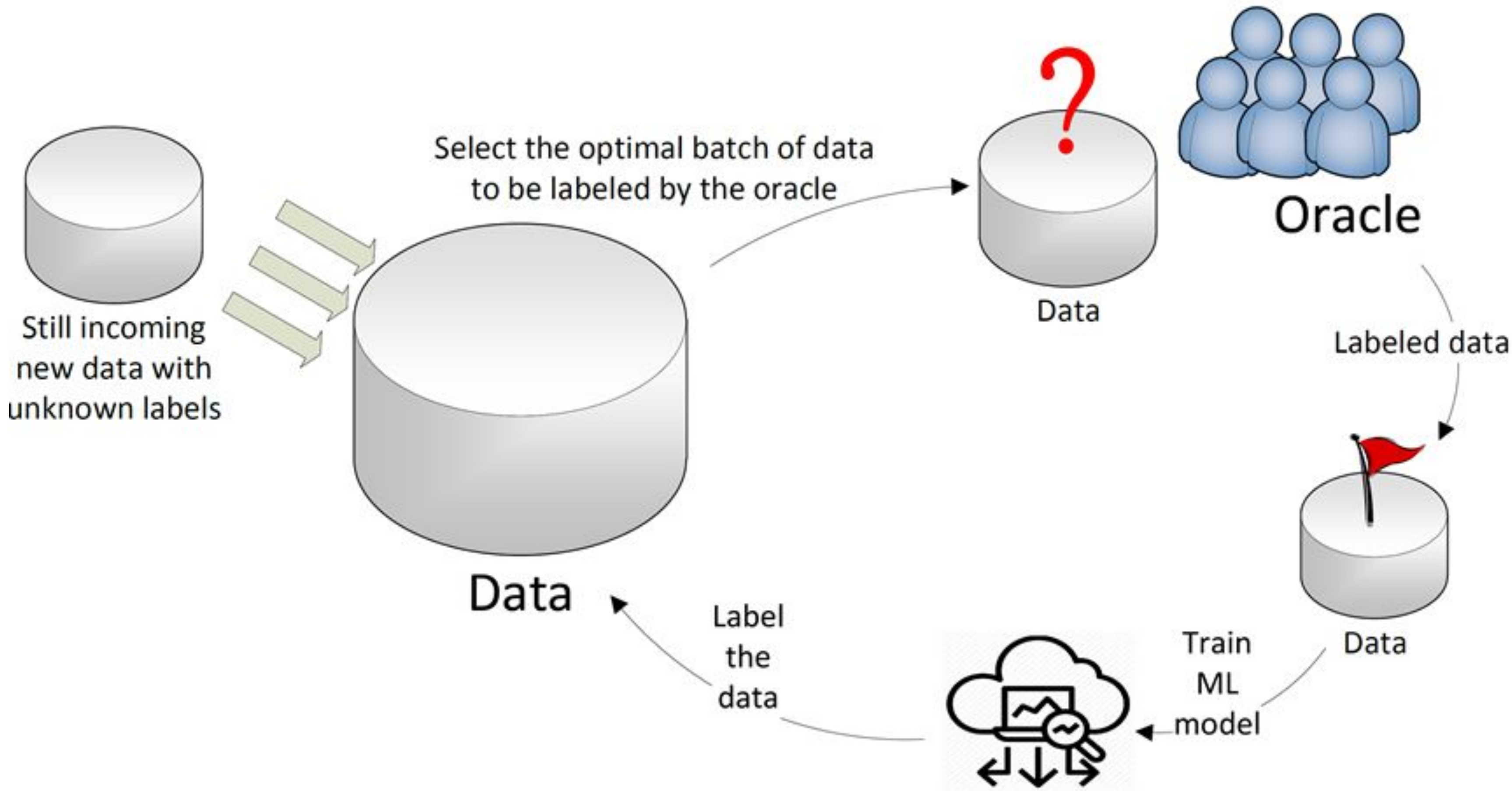
- Goal: obtain the best possible model with limited labeling capabilities, assuming that we can interact with the experts asking them to label indicated samples.
- How: iteratively query experts about labels for the most interesting/informative samples.
- Which samples are informative?
- Can we trust our experts?

Image: Freepik.com

A glossary (simplified)

- Learner - typically, an ML model that we want to actively train to perform a given prediction task.
- An instance/case - an entity described in our data for which we are making predictions/decisions.
- A query - an instance that we send to the oracle to obtain the label.
- A sample - one or more data instances drawn from the data space.
- The oracle - typically, a committee of experts that can assign labels to data instances.
- Ground truth - the actual label that should be assigned to a data instance.

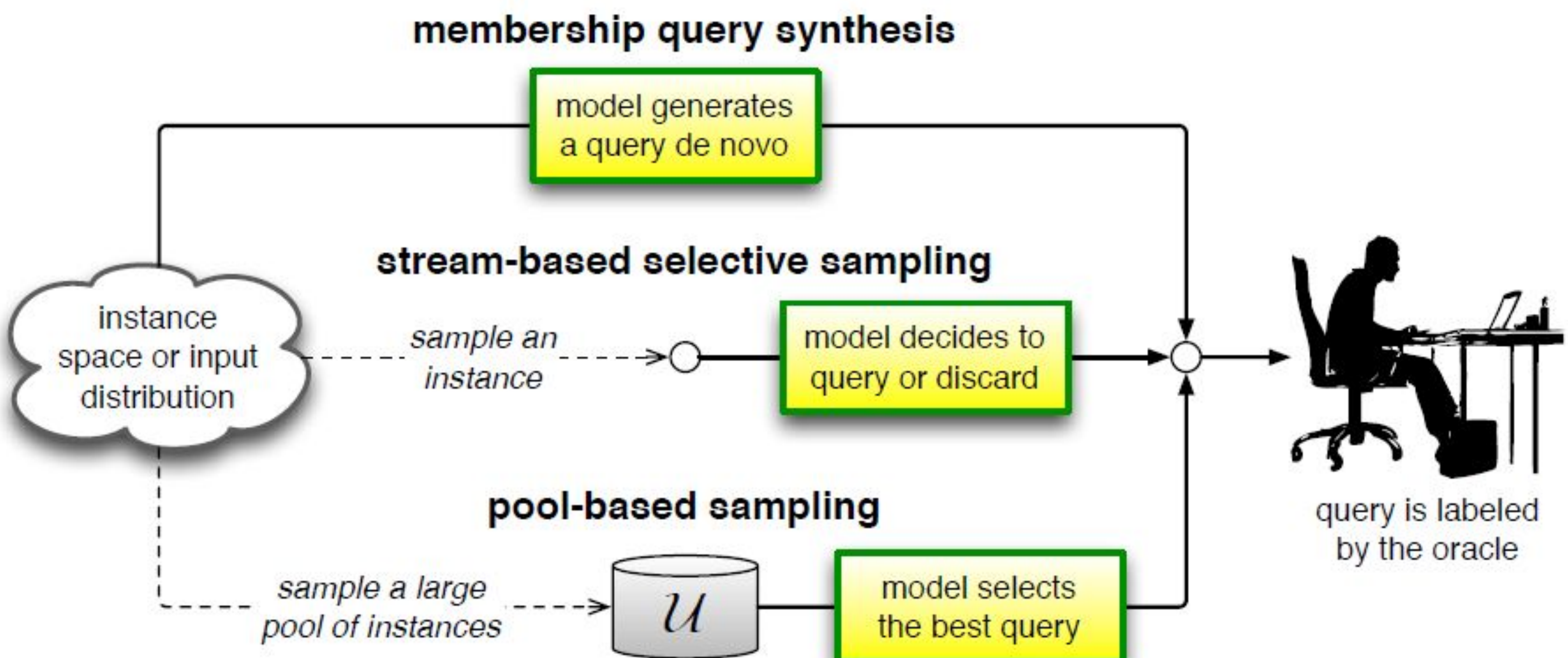
The active learning cycle



Exemplary application areas

- **Speech recognition:** It is extremely time-consuming and requires trained linguists. The annotation of speech recordings on a phoneme level can take 400 times longer than the actual recordings.
- **Natural language processing:** Many NLP-related ML tasks (information extraction, entity recognition, sentiment analysis) require detailed annotated data. The annotations/labels are often domain-specific and require expertise in a particular application field.
- **Image/video data processing:** Image classification, object detection or image segmentation require labeled examples. As with the NLP, in many domains (e.g., medicine) it requires specialized knowledge. Additionally, marking complex shapes is laborious.
- **Process mining:** Annotation of multivariate time series requires specialized tools, and if the data is multi-modal (e.g., video games data) the task is even more difficult.

The three main Active Learning scenarios



Schema from *Burr Settles: Active Learning Literature Survey (2010)*

Membership query synthesis

- One of the first Active Learning methods considered in ML literature.
- **The learner may request a label for any unlabeled instance in the input space, even if it is synthetic (not necessarily sampled from available data).**
- The generation of valid instances/queries might be problematic.
- Particularly effective in scenarios where the oracle is automatic (non-human), e.g., scientific experiments, computer simulations.

Speeding up simulations

- Membership query synthesis can be used to speed up scientific simulations.
  - We want to learn to predict experimental outcomes.
  - We can acquire some exemplary labels by performing costly or time-consuming experiments.
  - A reliable model would allow for tremendous efficiency improvements!
- How to synthesize a query?
  - Loss gradient-based optimization (if the model’s loss is smooth and differentiable with regard to the input).
  - Other optimization heuristics (e.g., hill climbing, simulated annealing, genetic algorithms).

UNIVERSITY OF WARSAW

An exemplary application 1

- King et al. (2004) show an example of a successful application of the membership query synthesis strategy.
  - Experimental discovery of metabolic pathways in the yeast.
  - Experiments performed by a robot.
  - Instances - a mixture of chemical solutions (a growth medium, yeast mutant).
  - Labels - the experiment outcomes.
- The active learning approach resulted in 100-fold decrease in cost compared to randomly generated experiments and 3-fold decrease in comparison to the greedy sampling heuristic.

Image: Freepik.com

Selective sampling

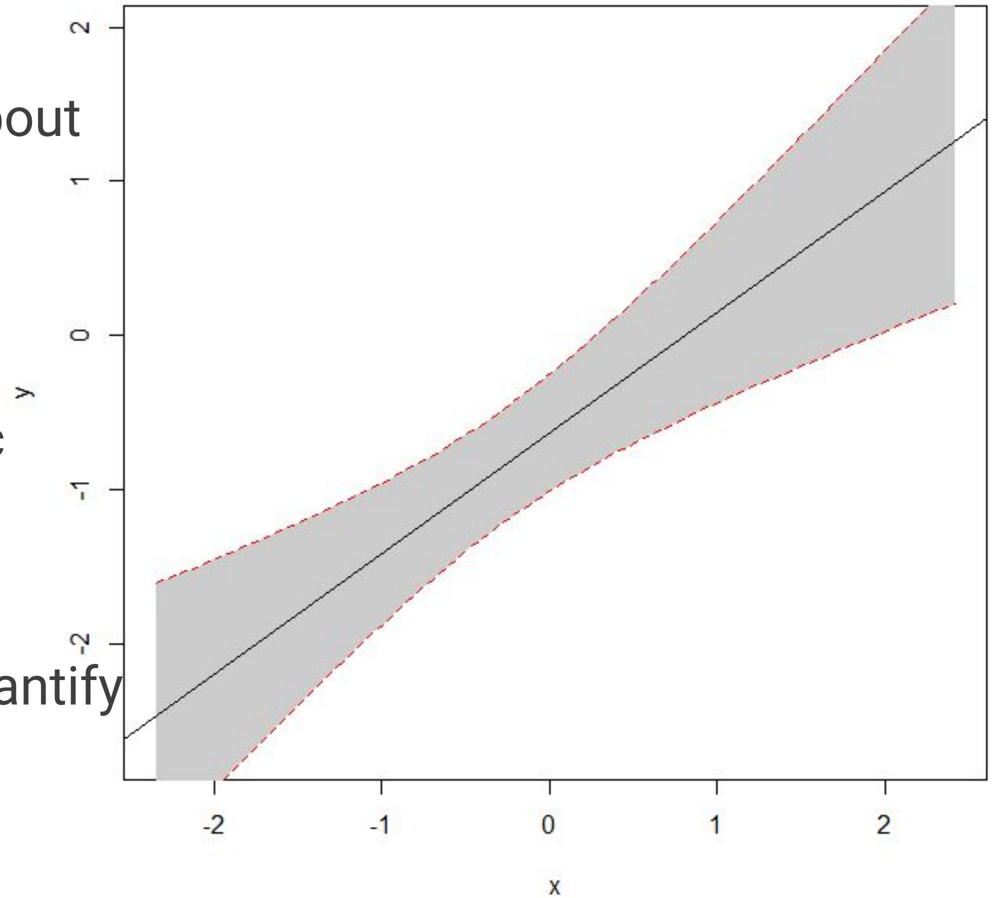
- Instead of synthesizing queries/instances, sample real cases from available examples.
  - No awkward or unrealistic cases which can be difficult to label by human annotators…
  - but the number of available examples needs to be large!
- Two main application scenarios:
  - Stream-based selective sampling.
  - Pool-based selective sampling.

Stream-based selective sampling

- Learner pulls out unlabeled instances one at a time from the data source.
  - For each instance, the learner has to decide whether to query the oracle for the label or discard it.
  - Even if the data distribution is not uniform, the sampled instances will reflect it.
- If the stream velocity is high, random sampling techniques are often used prior to the querying decision.
- The decision about the query is made based on the informativeness of an instance (e.g., the uncertainty of the learner).

Informativeness and uncertainty

- Informativeness of an instance expresses how much new information the knowledge about its label could provide to the learner.
- The informativeness is often associated with the epistemic uncertainty of a learner with regard to its predictions.
- In practice, we can usually quantify only the joint epistemic and aleatoric uncertainty.
- More details in the next lecture! 😊



Uncertainty-biased random sampling

- One of the simplest algorithms for selecting instances for querying.
- For any sampled instance *u*, the learner queries the oracle with a probability depending on the informativeness of *u*, e.g.:

$$P_{query}(u) = \frac{1}{1 + e^{\left(-\frac{Info(u)-\mu}{\sigma}\right)}} > rand(0,1)$$

- In practice, a possible informativeness function in this case is the Least Confidence uncertainty:

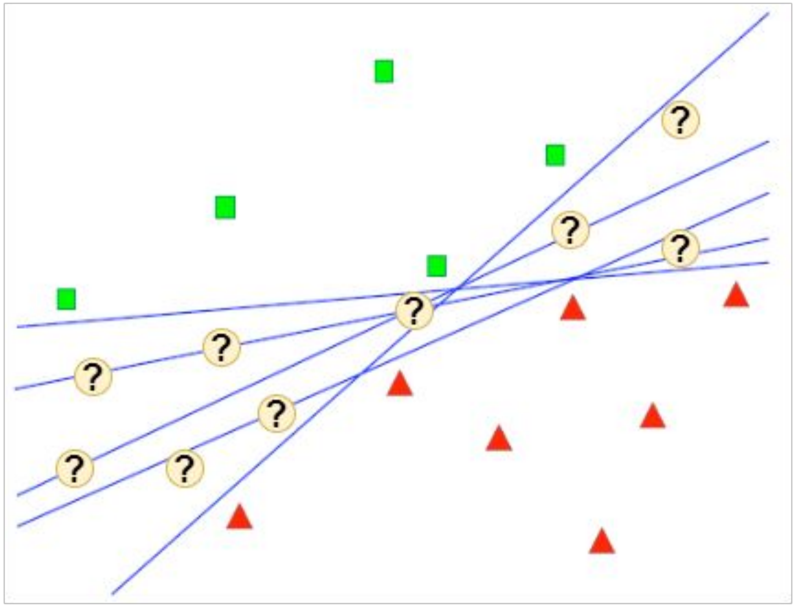
$$Info(u) = \frac{k\left(1 - P(y^*|u)\right)}{k-1}$$

Region of uncertainty querying

- Learner queries the oracle only if the sampled instance falls into learner's "*region of uncertainty*".
- How can we decide where exactly that region is?
  - We can set a specific threshold… (not so good idea).
  - We may try to decompose the data space into uncertainty regions using synthetic samples…
    (not so good idea either).
  - We can use a committee of prediction models!
    - The problem with this lies in the computational cost…
- The uncertainty region needs to be recalculated after each update of the learner.

The Query-By-Committee algorithm

- Proposed by Seung et al. (1992).
- The learner is composed of at least two prediction models of the same model class.
  - The models are sampled from, so-called, *version space*.
  - In computational learning theory, the version space is a set of hypotheses (models with different parametrization) that equally fit the training data but may disagree on new cases.
- An instance sampled from the stream is queried if the models sufficiently disagree.



Sampling from the version space?

- The main idea of the QBC algorithm is to query from the uncertainty region, thus limiting the “size” of the version space.
- It can be computationally challenging.
  - Typically, it is done approximately…
- For probabilistic (Bayesian) models, we may sample the parameter space.
- For other types of models, many ensemble learning techniques can be used:
  - Bootstrapping.
  - Segmentation of the feature space.

Disagreement in a committee

- We need to measure if the disagreement in the committee is sufficient to query the sample.
  - It is easy when there are only two models - in practice, it is often the case.
  - If there are many models, one may use the number of disagreeing pairs of voters.
  - Entropy of votes: $$\phi_{VE}(u) = -\sum_{l \in L} \frac{|\{i : y_i^* = l\}|}{K} \log\left(\frac{|\{i : y_i^* = l\}|}{K}\right)$$
  - Mean KL divergence: $$\phi_{KL}(u) = \frac{1}{K} \sum_{i=1}^{K} \sum_{l \in L} P_{\theta(i)}(y = l|u) \log\left(\frac{P_{\theta(i)}(y = l|u)}{P_{\Theta}(y = l|u)}\right)$$
- In the case of many voters, it is necessary to set a reasonable threshold - it may not be that easy.

An exemplary application 2

- Settles and Craven (2008) show an application of the QBC algorithm in the NLP domain.
  - Sequences of words -> streams of tokens.
  - The named entity recognition (NER) task on several different text corpora.
  - Instances - sequences of tokens (various length) with part-of-speech annotations.
  - Labels - the entity type associated with the sequence.
- The learners -> CRF models trained by bagging (an approximation of the version space).
  - Sequence vote entropy was proposed as the disagreement/uncertainty measure.

$$\phi_{SVE}(u) = -\sum_{l \in L^N} P_\Theta(y = l|u) \log(P_\Theta(y = l|u))$$

- The QBC algorithm performed consistently better than random and uncertainty samplings.

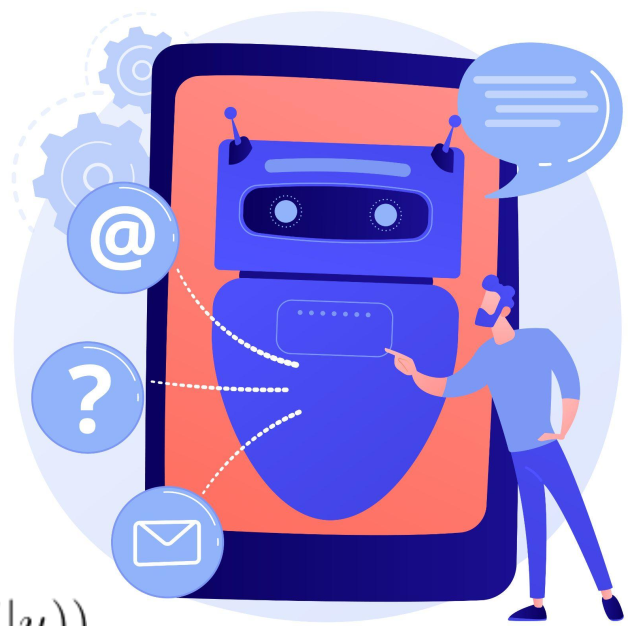
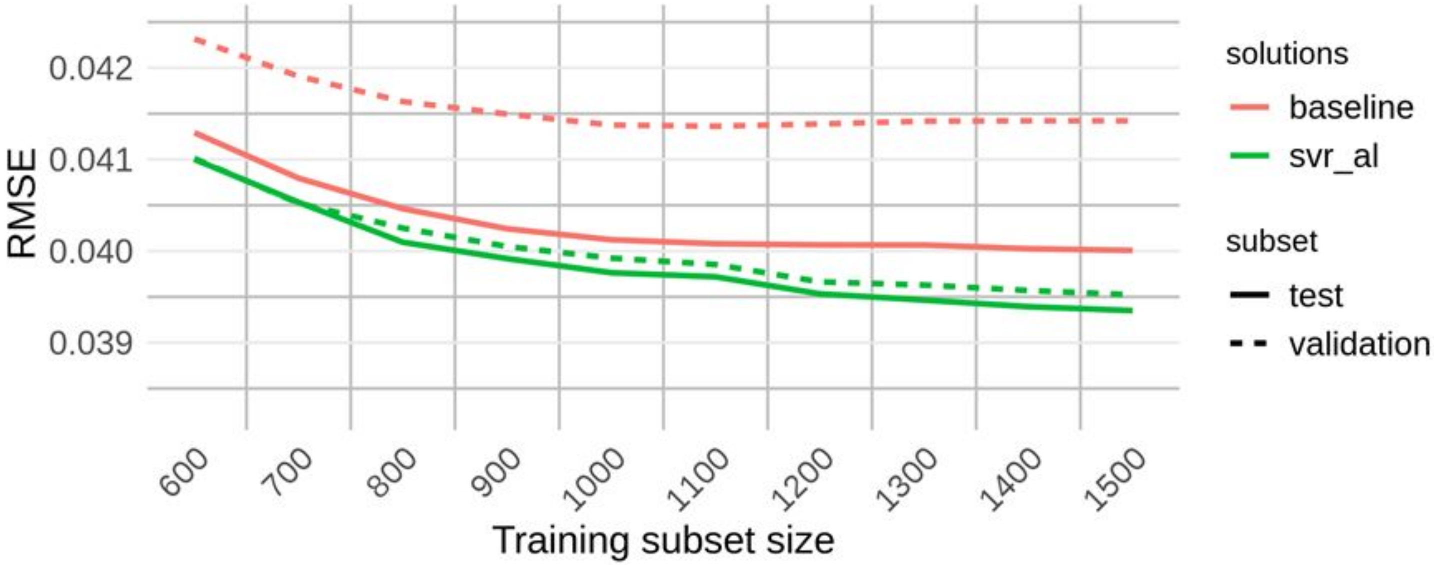


Image: Freepik.com

Pool-based selective sampling

- The <u>most common</u> active learning scenario.
  - Applicable when a large collection of unlabeled data is available.
  - At each iteration of the AL cycle, we may choose from many instances.
  - The unlabeled data pool may grow in time…
- An informativeness measure is used to evaluate all instances from the pool.
  - If the pool size is very large, some subsampling can be used…
- Queries are typically chosen in a greedy fashion.
- Numerous real-world applications!
- More about this approach in the next lecture ☺

An exemplary application 3



- Janusz et al. (2019) proposed a method based on a combination of informativeness density and diversity sampling for active learning of deck win-rates in a popular mobile video game Clash Royale.
- Historical win-rates were available for a large pool of decks. How will the win-rates change in a new season?
- Active learning outperformed random sampling and nu-SVR baselines. https://knowledgepit.ml/clash-royale-challenge/

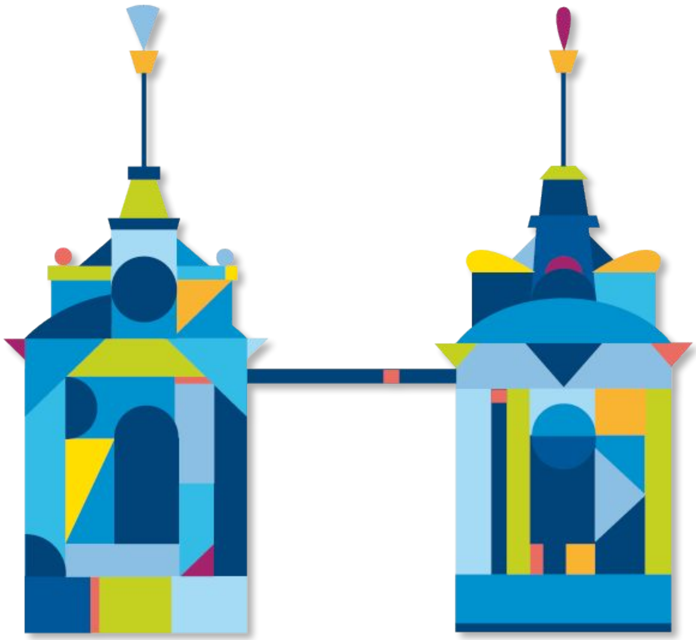
UNIVERSITY OF WARSAW

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. D. Angluin. Queries and concept learning. Machine Learning, 2:319–342, 1988
3. D. Angluin. Queries revisited. In Proceedings of the International Conference on Algorithmic Learning Theory, pages 12–31. Springer-Verlag, 2001.
4. R.D. King, K.E. Whelan, F.M. Jones, P.G. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, and S.G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature, 427(6971):247–52, 2004.
5. R.D. King, J. Rowland, S.G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L.N. Soldatova, A. Sparkes, K.E. Whelan, and A. Clare. The automation of science. Science, 324(5923):85–89, 2009.
6. H.S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In Proceedings of the ACM Workshop on Computational Learning Theory, pages 287–294, 1992.
7. B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1069–1078. ACL Press, 2008.
8. A. Janusz, Ł. Grad, M. Grzegorowski: Clash Royale Challenge: How to Select Training Decks for Win-rate Prediction. FedCSIS 2019: 3-6, 2019



QUESTIONS OR COMMENTS?

a.janusz@mimuw.edu.pl

or

d.kaluza@mimuw.edu.pl