UNIVERSITY OF WARSAW

WYDZIAŁ MATEMATYKI, INFORMATYKI I MECHANIKI

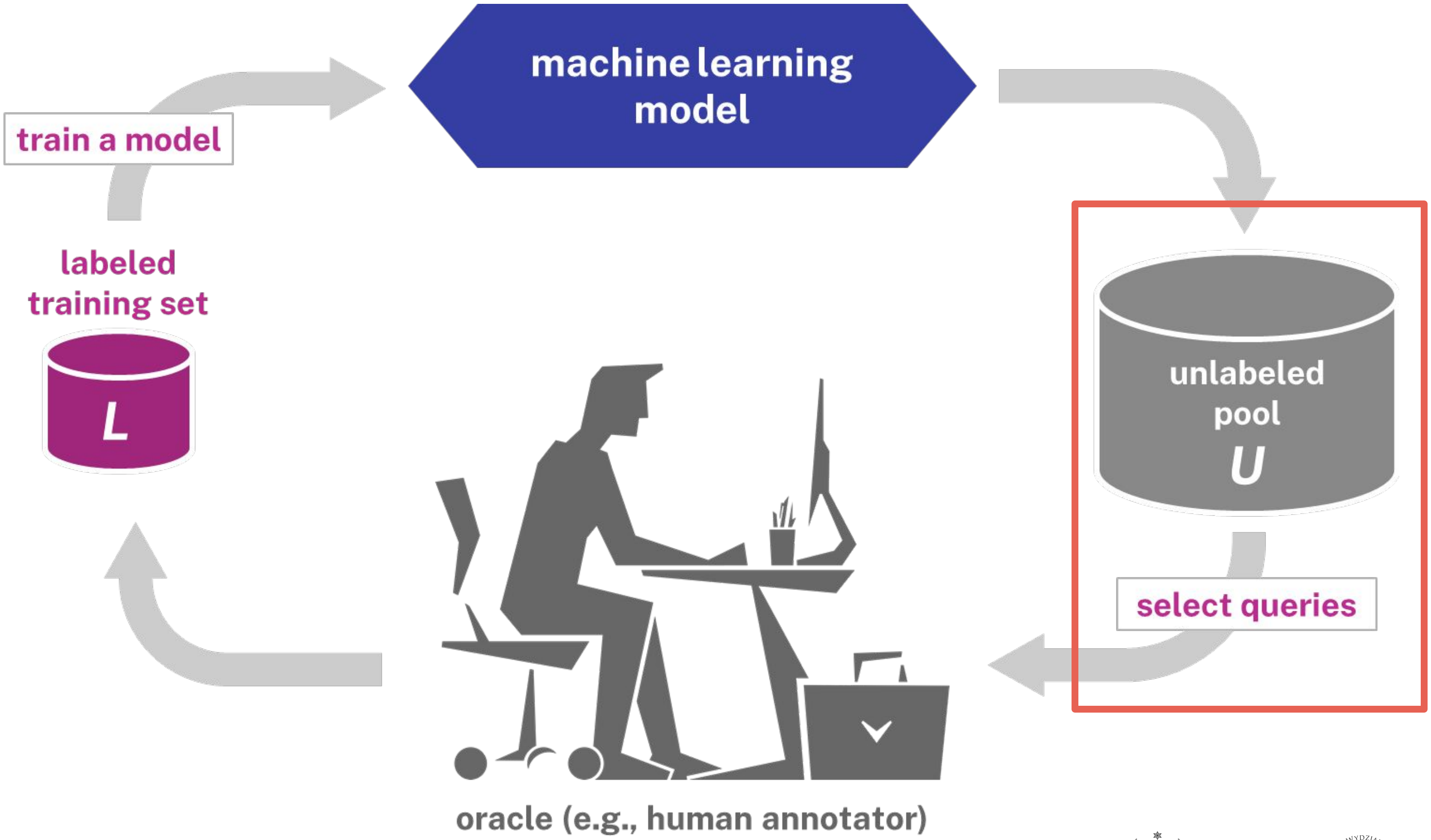
Active Learning - the pool-based selective sampling (part 3)

Andrzej Janusz
Daniel Kałuża

THE PLAN

- A recap of the previous lectures.
- Expected Model Change framework.
- Expected Error Reduction.
- Monte Carlo Dropout-based technique for training Bayesian neural networks.
- The BALD algorithm and decomposition of uncertainty.
- Exemplary algorithms and use-cases.
- Summary.

The active learning cycle - again



Uncertainty, representativeness, diversity

We search for a subset of data pool $U^* \subset DP$ that maximizes generalization capacity of our model:

$$U^* = \arg\max_{U:|U|=K} \mathbb{E}_{(X,Y)}[q(Y, f^U(X))]$$

where f^U is a model trained on a subset $U \subset DP$ whose size is *K* and *q* is a predefined quality metric.

We considered modeling the sample informativeness using the “uncertainty-representativeness-diversity” approach:

$$Info(u,B) = \frac{1}{c} \cdot Unc(u) + \left(\frac{1}{r} \cdot R(u)\right)^{\alpha} + \left(\frac{1}{d} \cdot Dis(u,B)\right)^{\beta}$$

UNIVERSITY OF WARSAW

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.
- Regardless of the initial batch selection method, random samples are always needed for the evaluation!

A different view on the informativeness

- It is difficult to discern between aleatory and epistemic uncertainty.
- It might be difficult to tell which uncertainty function is the most suitable for a given model/task.
- We may try to take a more direct approach to the assessment of sample's usefulness.
  - Can we measure the impact of a sample on the training process?

Expected model change (EMC) framework

- Instead of measuring the prediction uncertainty, we may measure how the addition of a given sample to the labeled set *would change the model*.
  - We need to compute the expectation over all possible labelings.
  - May be computationally challenging.
- We quantify "*the change*" differently for different types of prediction models.
- Typically, this method is combined with gradient descent learning algorithms.

Gradient descent algorithm

- Gradient descent (GD) is a first-order optimization algorithm.
- Having defined a loss function, we iteratively search for the local minimum moving in the opposite direction to the gradient (the steepest descent).
- A common training algorithm for ML models:

$$\theta := \theta - \alpha \frac{\partial \ell_\theta(U_{tr})}{\partial \theta}$$

- Stochastic gradient descent (SGD) is a stochastic approximation of GD.

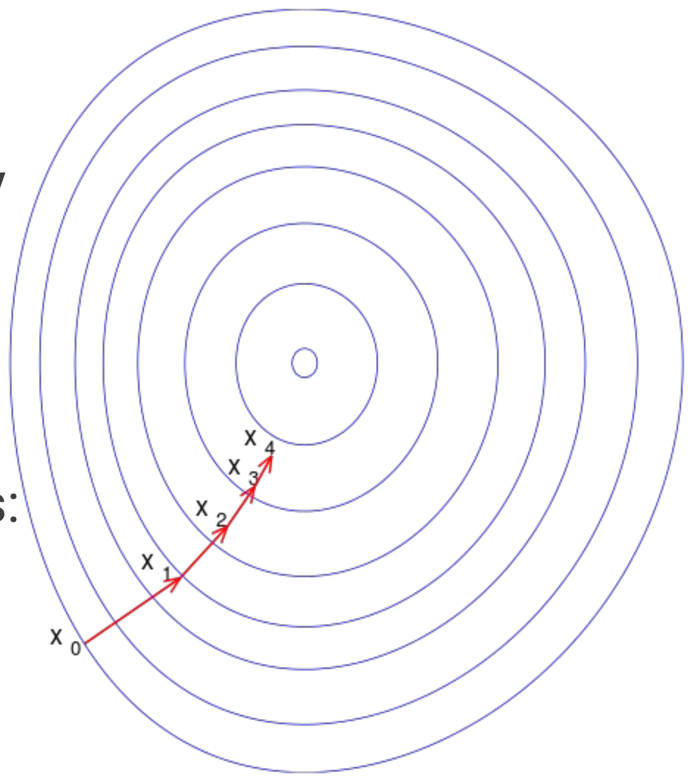


Image taken from Wikipedia (public domain).

Expected gradient length

- A special case of EMC for gradient-based learning algorithms.
- The "change" of the model corresponds to the length of the training gradient $\nabla \ell_\theta(U_{tr})$.
  - We query using the rule:

$$u^*_{EGL} = \arg\max_u \sum_i P(y_i|u)\, ||\nabla \ell_\theta(U_{tr} \cup \langle u, y_i \rangle)||$$

  - If we assume that samples are independent, we get:

$$u^*_{EGL} = \arg\max_u \sum_i P(y_i|u)\, ||\nabla \ell_\theta(\langle u, y_i \rangle)||$$

- Appropriate data scaling is a must!

An example - AL with linear regression

- W. Cai et al. (2013) demonstrate effectiveness of the EMC method for training linear regression model on several benchmark data sets.
- Bootstrapping of *K* models is used to estimate the distribution of predictions.
- For the typical squared error loss, the query selection rule simplifies to:

$$u^*_{EGL} = \arg\max_u \frac{1}{K} \sum_{k=1}^{K} ||(\Phi_k(u) - \Phi(u)) \cdot u||$$

- EMC outperformed the query-by-committee (QBC) and other baselines on all considered data sets.

***Concrete* data set from the UCI repository.**

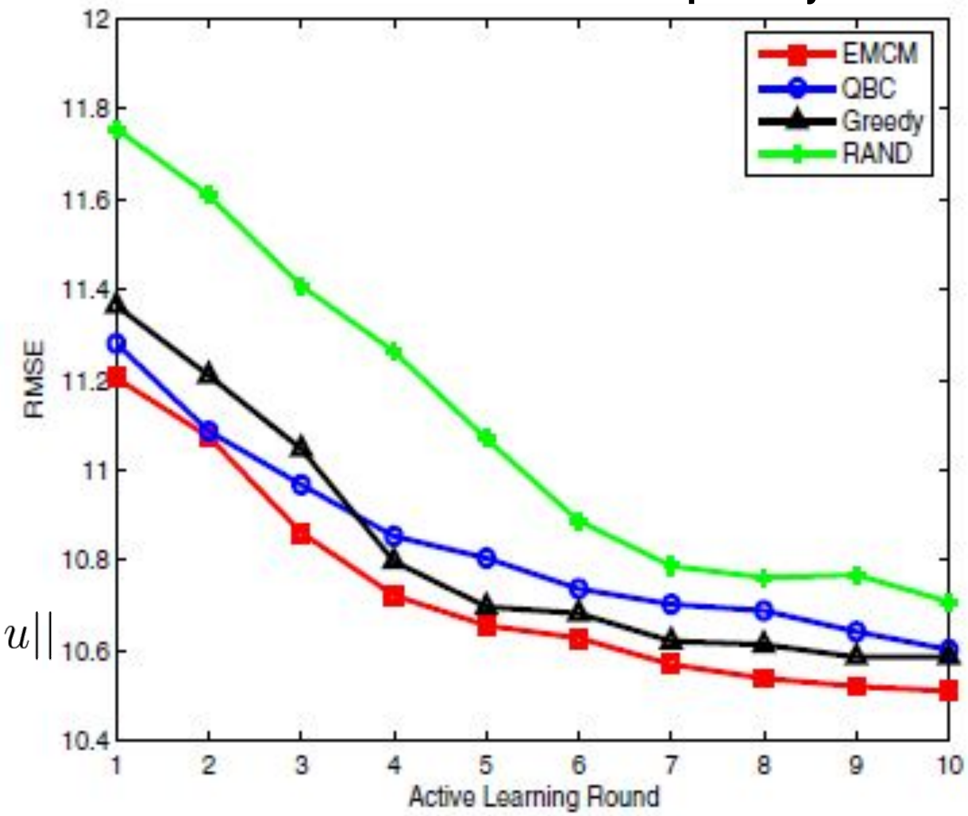


Image taken from W. Cai, et al. (2013), Maximizing Expected Model Change for Active Learning in Regression.

Expected error reduction (EER) approach

- A decision-theoretic framework aiming to measure how much model's generalization error is likely to be reduced by the information about the true label.
  - As in the case of EMC, we need to compute the expectation over all possible labels.
  - We assume that predictions of the new model are correct and estimate the generalization error on U.
  - Or we estimate the error using leave-one-out technique on the labeled data set U_{tr}.

EER for various losses

- We may select the query that minimizes the expected 0-1 loss:

$$u^*_{0-1} = \arg\min_{u \in U} \sum_i P_\theta(y_i|u) \left(\sum_{x \in U \setminus u} 1 - P_{\theta+(u,y_i)}(\hat{y}|x) \right)$$

- or the one that minimizes the expected log-loss:

$$u^*_{log} = \arg\min_{u \in U} \sum_i P_\theta(y_i|u) \left(-\sum_{x \in U \setminus u} \sum_j P_{\theta+(u,y_i)}(y_j|x) \log(P_{\theta+(u,y_i)}(y_j|x)) \right)$$

- In any case, the required computations might be prohibitive even for moderate-size data pools.

An example - Monte Carlo Estimation of EER

- In *N. Roy & A. MacCallum 2001*, authors propose a modification of the typical EER approach:

$$u^*_{EER} = \arg\min_{u \in U} \sum_i P_\theta(y_i|u) \left(\int_{x \in X \subset U} L(\hat{y}, P_{\theta+(u,y_i)}(y|x)) P(x) \right)$$

- The loss is estimated on a random subset of the unlabeled pool *U*.
- Incremental training instead of re-training the model from the scratch.
- Only a part of (similar) instances needs to be re-classified.
- Experiments with Naive Bayes and document classification.
- Good performance using four times less queries than considered baselines (QBC, uncertainty, random).

Variance reduction framework

- A direct minimization of the expected loss is usually extremely expensive computationally, and in general cannot be done in closed form.
- An indirect minimization of the expected loss is sometimes possible...
  - For a regression problem and the squared error loss:

$$
\begin{aligned}
E_U[(\hat{y} - y)^2 | u] &= E[(y - E[y|u])^2] && \text{model-independent} \\
&+ (E_{U_{tr}}[\hat{y}] - E[y|u])^2 && \text{model's bias} \\
&+ E_{U_{tr}}[(\hat{y} - E_{U_{tr}}[\hat{y}])^2] && \text{model's variance}
\end{aligned}
$$

Fisher information matrix

- By reducing the variance, we always reduce the generalization error.
- For some models (e.g. NN, LR, GMM), the variance of output can be easily estimated.
  - For a neural network and the squared error loss:

$$\sigma_{\hat{y}}^2 \approx \left[\frac{\partial \hat{y}}{\partial \theta}\right]^T \left[\frac{\partial^2}{\partial \theta} L_\theta(U_{tr})\right]^{-1} \left[\frac{\partial \hat{y}}{\partial \theta}\right] \approx \nabla u^T F^{-1} \nabla u$$

- In some cases model retraining is not necessary.
- Query: $u_{VR}^* = \arg\min_{u \in U} \left\langle \hat{\sigma}_{\hat{y}}^2 \right\rangle^{+u}$

Bayesian NNs and variational inference

- BNNs allow to express the uncertainty of predictions.
  - We assume that the model's weights are stochastic: $\theta \sim p(\theta)$
  - We train the model using the Bayesian inference.

$$p(\theta|U_{tr}) = \frac{p(Y_{tr}|U_{tr},\theta)p(\theta)}{\int p(Y_{tr}|U_{tr},\theta)p(\theta)\partial\theta} \propto p(Y_{tr}|U_{tr},\theta)p(\theta)$$

- This computation is intractable - we have to approximate:

$$p(y|u,U_{tr}) = \int p(y|u,\theta)p(\theta|U_{tr})\partial\theta$$

$$\approx \int p(y|u,\theta)q^*(\theta)\partial\theta \approx \frac{1}{T}\sum_{i=1}^{T} p(y|u,\hat{\theta}_i)$$

Monte Carlo dropout

- Gal & Ghahramani (2016b) show that random zeroing of weights before each layer of NN is equivalent to Bayesian approximation of the model's parameter distribution:

$$\theta \sim p(\theta | U_{tr}) \approx q^*(\theta) = W_i \cdot diag([z_{i,j}]_{j=1}^{K_i})$$

$$where\ z_{i,j} \sim Bernoulli(p_i)$$

  - Standard DNNs are easier to train than Bayesian equivalents.
  - We train the model using dropout layers and keep them active for the inference.
- It can be applied to a wide variety of model families.

The BALD algorithm

- Introduced by Houlsby et al. (2011) for GPC but can be applied for any Bayesian model.

  Predictive distribution: $p(y|u,\theta) \;\; where \; \theta \sim p(\theta|U_{tr})$

- We want to query the case that maximizes the decrease in expected posterior entropy of parameters:

$$u^* = \arg\max_u H(\theta|U_{tr}) - E_{y\sim p(y|u,\theta)}[H(\theta|y,u,U_{tr})]$$

  - This can be equivalently rewritten as:

$$u^*_{BALD} = \arg\max_u \underbrace{H(y|u,U_{tr})}_{\text{total prediction uncertainty}} - \underbrace{E_{\theta\sim p(\theta|U_{tr})}[H(y|u,\theta)]}_{\text{aleatoric uncertainty}}$$

BALD application example

- In Murray et al. (2021) authors show an interesting application of the BALD algorithm.
  - Unlike in a typical BNNs, weights are kept deterministic while the model's depth is assumed to be stochastic.
  - A categorical prior is placed over the model's depth distribution:

$$p(d) = Cat(d|\{\beta_i\}_{i=0}^{D})$$

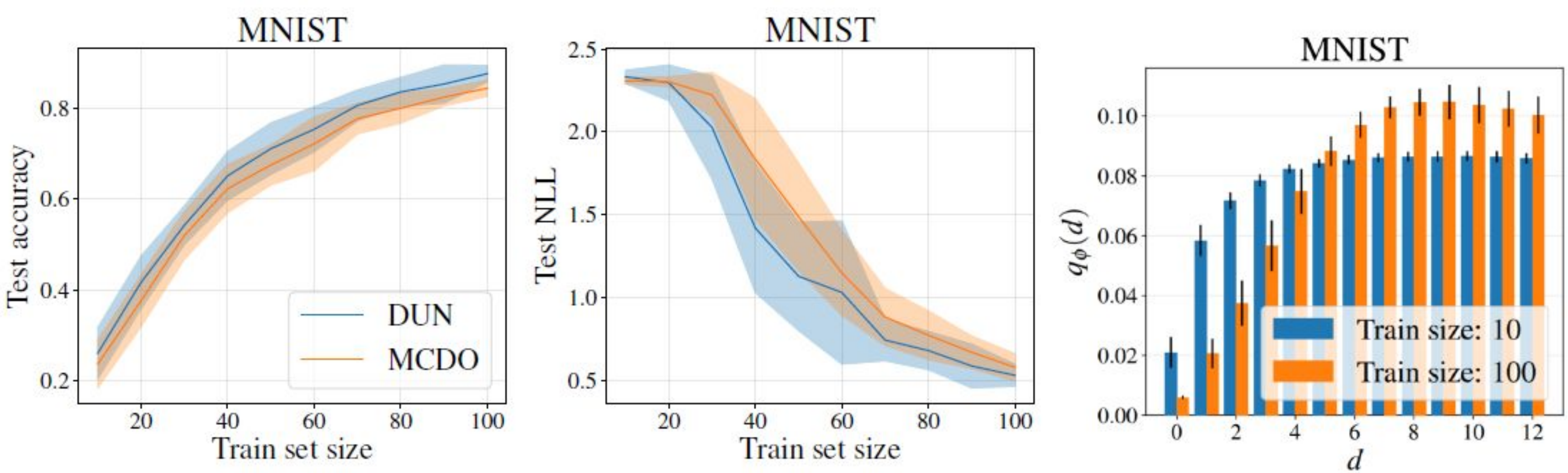


Image taken from Murray et al. (2021)

Pool-based selective sampling - summary (1)

- The most common AL scenario.
- A greedy selection of queries:
  - Uncertainty sampling.
  - Uncertainty-representativeness sampling.
- Diversification of query batches.
- Query-by-committee algorithm can be adapted to the pool-based scenario.

Pool-based selective sampling - summary (2)

- A different view of the uncertainty/informativeness of data samples:
  - How the sample's label would impact the model?
  - How much can we expect to reduce the model's error by querying for the sample's label?
- Decomposition of the model's uncertainty into the areatoric and epistemic parts.
  - The BALD approach to the uncertainty of Bayesian models.

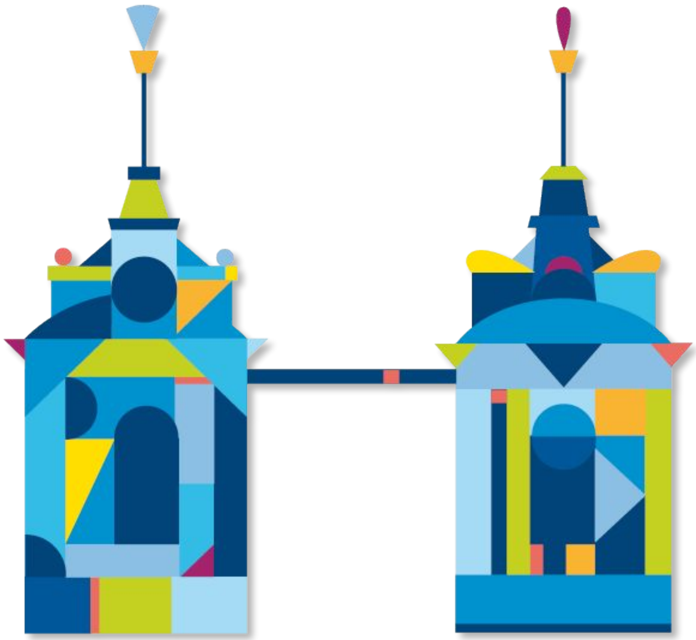
Summary

- We discussed several methods for choosing the most informative samples in the pool-based selective sampling scenario.
- We considered variational inference and Bayesian neural networks for estimating the model's uncertainty.
- We talked about a decomposition of model's uncertainty into the aleatoric and epistemic parts.
- We analyzed a few AL algorithms and application examples for different ML tasks.

UNIVERSITY OF WARSAW

Literature:

1. B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, (2010).
2. W. Cai, Y. Zhang and J. Zhou, "Maximizing Expected Model Change for Active Learning in Regression," 2013 IEEE 13th International Conference on Data Mining, 2013, pp. 51-60 (2013)
3. N. Roy, A. McCallum, "Toward Optimal Active Learning through Sampling Estimation of Error Reduction". In Proceedings of the 18th International Conference on Machine Learning,: 441-448 (2001)
4. Y. Gal, Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference." ICLR workshop track, (2016a).
5. Y. Gal, Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning." In Proceedings of the 33th International Conference on Machine Learning, (2016b).
6. Y. Gal, R. Islam, Z. Ghahramani. "Deep Bayesian active learning with image data." In Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17). JMLR.org, 1183–1192. (2017).
7. N. Houlsby, F. Huszar, Z. Ghahramani, M. Lengyel: "Bayesian Active Learning for Classification and Preference Learning". CoRR abs/1112.5745 (2011)
8. Chelsea Murray, James Urquhart Allingham, Javier Antorán, José Miguel Hernández-Lobato: Depth Uncertainty Networks for Active Learning. CoRR abs/2112.06796 (2021)



QUESTIONS OR COMMENTS?

a.janusz@mimuw.edu.pl

or

d.kaluza@mimuw.edu.pl