Information Theoretic Active Learning

Shachar Shayovitz

26/08/2024

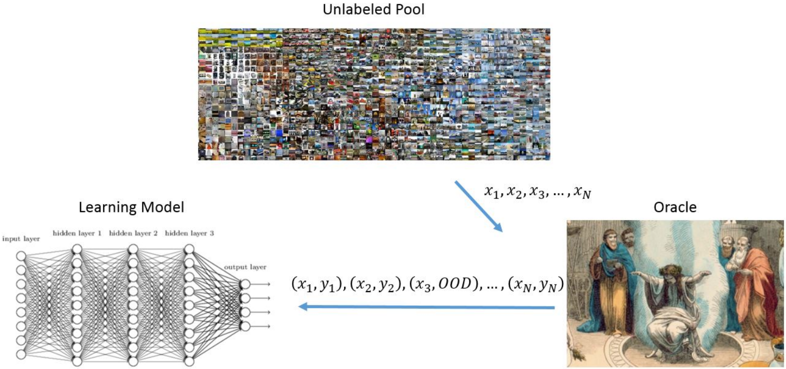
Ph.D. student under the supervision of Prof. Meir Feder

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

Table of Contents

Motivation

Passive Learning



Motivation

Models are hungry for high quality data!

- Data storage becomes cheaper.
- **Bottleneck**: Someone needs to label the data!

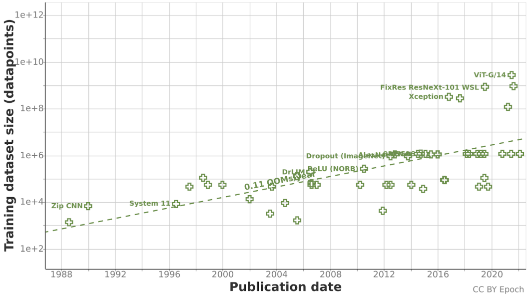
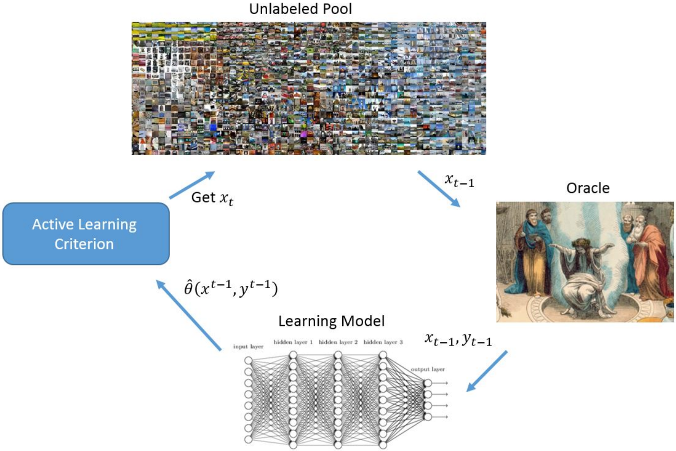


Figure: Trends in training dataset sizes [1]

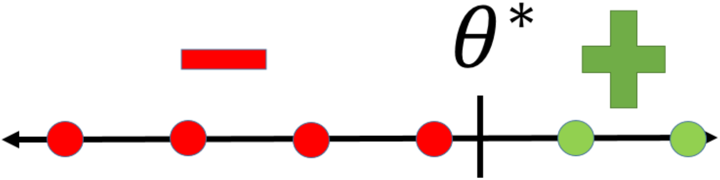
[1] [AI24]

Active Learning

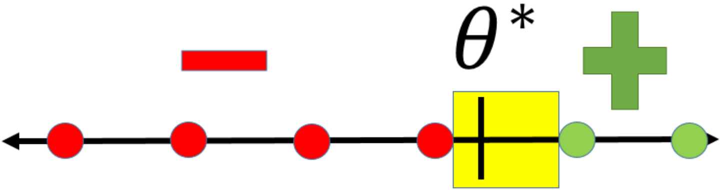


Motivating Example

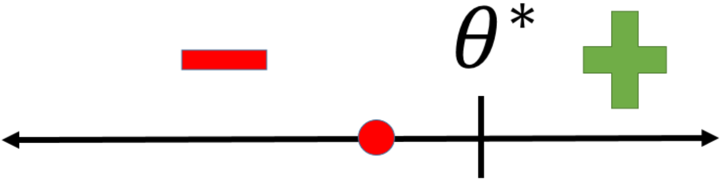
Motivating Example



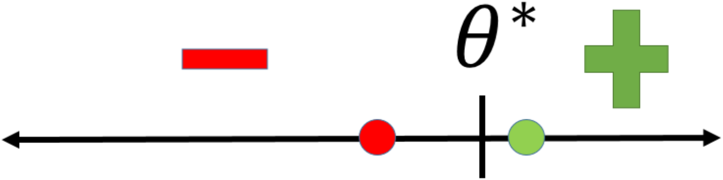
Motivating Example



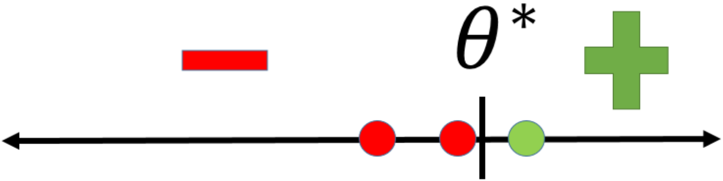
Motivating Example



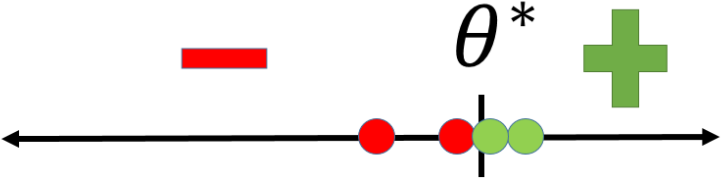
Motivating Example



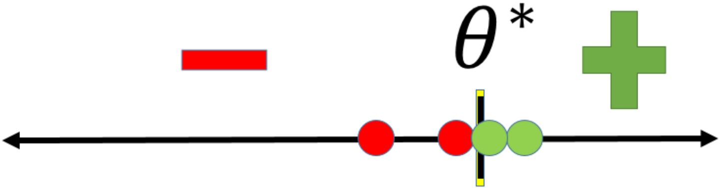
Motivating Example



Motivating Example



Motivating Example



Real World Active Learning

Data Collection in the Wild

Out of Distribution Data

- In the real world, data is collected from diverse sources.
- These can be open source datasets which are not tailored to a specific task.
- Therefore, the data pool **will** contain Out Of Distribution (OOD) samples.
- Removing OOD samples requires expert assistance.

Data Collection in the Wild

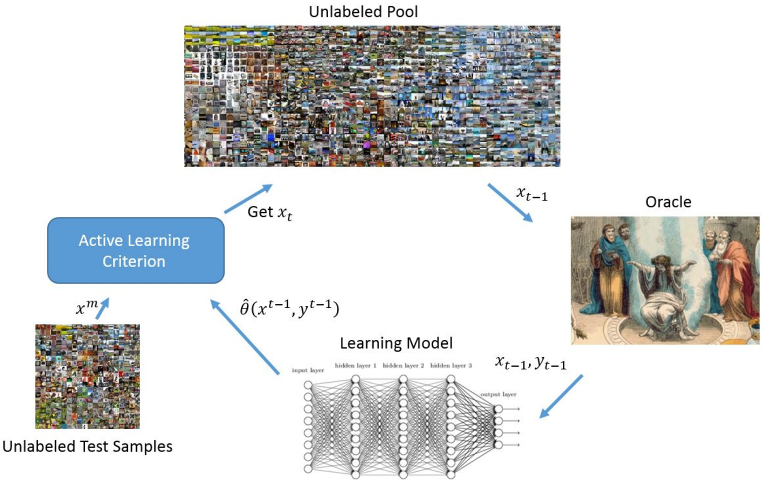
Out of Distribution Data

- In the real world, data is collected from diverse sources.
- These can be open source datasets which are not tailored to a specific task.
- Therefore, the data pool **will** contain Out Of Distribution (OOD) samples.
- Removing OOD samples requires expert assistance.

Test Aware Active Learning

We propose to use a small **un-labelled** sample from the test distribution to minimize expert assistance and training set selection.

Test Aware Active Learning



Part 1: The Stochastic Setting

Includes:

- Shayovitz Shachar, and Meir Feder. "Universal active learning via conditional mutual information minimization." IEEE Journal on Selected Areas in Information Theory 2.2 (2021): 720-734. [SF21]
- Shayovitz Shachar, and Meir Feder. "Minimax active learning via minimal model capacity." 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2019 (**Best paper finalist**) [SF19]

Mathematical Setup

Learning Setting

- **Stochastic setting:**
  - Examples (x, y) are drawn from some family of hypotheses $p(y|x, \theta)$ where $\theta \in \Theta$.
  - Test feature drawn from $p(x)$
- Labeling budget of N queries.
- Probabilistic learners: $q(y|x)$.
- Log-loss cost function: $-\log(q(y|x))$.

Mathematical Setup

Learning Setting

- **Stochastic setting:**
  - Examples (x, y) are drawn from some family of hypotheses $p(y|x, \theta)$ where $\theta \in \Theta$.
  - Test feature drawn from $p(x)$
- Labeling budget of N queries.
- Probabilistic learners: $q\left(y|x\right)$.
- Log-loss cost function: $-\log\left(q\left(y|x\right)\right)$.

Informal Objective

Sequentially select features based on past examples (x^N, y^N) and construct a learner, $q\left(y|x, x^N, y^N\right)$, which will perform "well".

Active Learning Criteria

- Maximum Uncertainty (MU)
  - $\hat{x}_n = \arg\max_{x_n} H\left(y_n | x^n, y^{n-1}\right)$.
  - Sensitive to noise.
- Bayesian Active Learning by Disagreement (BALD) [HHGL11]
  - $\hat{x}_n = \arg\max_{x_n} I\left(\theta; y_n | x^n, y^{n-1}\right)$.
  - Focused on model estimation.

Active Learning Criteria

- Maximum Uncertainty (MU)
  - $\hat{x}_n = \arg\max_{x_n} H\left(y_n | x^n, y^{n-1}\right)$.
  - Sensitive to noise.
- Bayesian Active Learning by Disagreement (BALD) [HHGL11]
  - $\hat{x}_n = \arg\max_{x_n} I\left(\theta; y_n | x^n, y^{n-1}\right)$.
  - Focused on model estimation.

Main Issues

- No justification for the prior, $\pi\left(\theta\right)$.
- No focus on prediction.

Mathematical Setup

Optimal Learner

- Similarly to the statistical learning approach, we would like to find a learner $\hat{q}(y|x)$ which minimizes:

$$\hat{q}(y|x) = \arg\min_{q} E_{p(y|x,\theta)} \left(-\log q(y|x)\right)$$

- Clearly this implies that $\hat{q}(y|x) = p(y|x,\theta)$ (minimal KLD).

Mathematical Setup

Optimal Learner

- Similarly to the statistical learning approach, we would like to find a learner $\hat{q}(y|x)$ which minimizes:

$$\hat{q}(y|x) = \arg\min_{q} E_{p(y|x,\theta)} \left(-\log q(y|x)\right)$$

- Clearly this implies that $\hat{q}(y|x) = p(y|x,\theta)$ (minimal KLD).

Problem

- Unfortunately, the learner has no access to the true θ.

Minimax Active Learning Formulation

- Find a sequential selection strategy $\left\{\phi(x_t|x^{t-1}, y^{t-1})\right\}_{t=1}^{N}$ which optimizes the minimax regret to the optimal learner for a random test point (x, y):

$$R = \min_{\{\phi_t\}_{t=1}^{N}} \min_{q} \max_{\theta} E\left\{\log\left(\frac{p\left(y|x,\theta\right)}{q\left(y|x,x^N,y^N\right)}\right)\right\}$$

  where x^N, y^N are the training examples.
- The expectation is performed over the joint probability:

$$p\left(y, x, x^N, y^N|\theta\right) = p\left(y|\theta, x\right) \Pi_{t=1}^{N} p\left(y_t|x_t, \theta\right) \phi\left(x_t|x^{t-1}, y^{t-1}\right) p(x)$$

Minimax Active Learning Alternative Formulation

- Another useful formulation is:

$$R = \min_{\{\phi_t\}_{t=1}^{N}} \min_{q} \max_{\pi(\theta)\in\Pi} E\left\{\log\left(\frac{p\,(y|x,\theta)}{q\,(y|x,x^N,y^N)}\right)\right\}$$

  where x^N, y^N are the training examples and Π is a set of distributions on the random variable θ.
- This formulation will be useful for regularized linear regression and Gaussian Process Classification where the prior on θ is either regularized or explicitly given.
- The expectation is performed over the joint probability:

$$p\left(y,x,x^N,y^N,\theta\right) = p\,(y|\theta,x)\,\Pi_{t=1}^{N} p\,(y_t|x_t,\theta)\,\phi\left(x_t|x^{t-1},y^{t-1}\right)p(x)\pi(\theta)$$

Capacity Redundancy Theorem for Active Learning

Theorem [SF19]

The minimax active learning problem is equivalent to the following criterion:

$$R = \min_{\left\{\phi(x_t|x^{t-1},y^{t-1})\right\}_{t=1}^{N}} C_{Y;\theta|X,Y^N,X^N}$$

where[a],

$$C_{Y;\theta|X,Y^N,X^N} = \max_{\pi(\theta)} I\left(Y;\theta|X,Y^N,X^N\right)$$

and the optimal learner is:

$$q^*\left(y|x,x^N,y^N\right) = \sum_{\theta} p\left(\theta|y^N,x^N\right) p\left(y|\theta,x\right)$$

[a]For the alternative formulation, we can use $\pi(\theta) \in \Pi$

Linear Regression

The linear regression hypothesis class:

$$\underline{y} = X\underline{\theta} + \underline{z}$$

Assumptions:

- $X \in \mathbb{R}^{nxd}$ is a design matrix of n feature vectors.
- $\underline{y} \in \mathbb{R}^n$ is the vector of observable responses.
- $\theta \in \mathbb{R}^d$ is the model vector.
- $\underline{z} \sim N\left(0, \sigma^2 \mathbb{I}_{\mathbb{n}}\right)$.

The error covariance of the OLS solution is:

$$\Sigma^{-1} = \sigma^2 \left(X^T X\right)^{-1}$$

Experimental Design

- The design problem reduces to find a design matrix X which **minimizes** some function of the covariance matrix: $f\left(\Sigma^{-1}\right)$.
- Extensive research over the last decade under the mathematical field of "Optimal Experimental Design": [Puk06]
  - **A** Optimal Design: $f_A(\Sigma) = \frac{1}{p} Tr\left(\Sigma^{-1}\right)$
  - **D** Optimal Design: $f_D(\Sigma) = det\left(|\Sigma|\right)^{-\frac{1}{d}}$
  - **G** Optimal Design: $f_G(\Sigma) = \max \text{diag}\left(X_{test}\Sigma^{-1}X_{test}^T\right)$
  - **V** Optimal Design: $f_V(\Sigma) = Tr\left(X_{test}\Sigma^{-1}X_{test}^T\right)$

Universal Active Learning for Linear Regression

Consider the following hypothesis class:

$$P_\Theta = \{p\left(y|x,\theta\right)|\theta \in \mathbb{R}^d\}$$

Each member learner defined as:

$$p\left(y|x,\theta\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(y - x^T\theta\right)^2\right)$$

The model prior, $\pi(\theta) \in \Pi$

$$\Pi = \left\{\pi(\theta)|\,\mathbb{E}\left(\theta\right) = 0, \frac{1}{d}\,\mathrm{Tr}\left(\mathbb{E}\left(\theta\theta^T\right)\right) \leq \sigma_\theta^2\right\}$$

Universal Active Learning for Linear Regression

Theorem [SF21]

- Assuming the hypothesis class, as defined in the previous slide, then the following holds (with equality for high SNR):

$$R \leq \min_{\underline{x}^n} \mathrm{Tr}\left(\mathbb{E}\left(X_{test} X_{test}^T \right) \left(X_n^T X_n + \frac{\sigma^2}{\sigma_\theta^2} I_d \right)^{-1} \right)$$

  X_n and X_{test} are the concatenation of the training and test vectors respectively.

- The capacity achieving prior is:

$$\theta \sim \mathcal{N}(0, \sigma_\theta^2 I_d)$$

Universal Active Learning for Linear Regression

- Closed form solution to the Active Learning problem.
- This criterion is closely related to the A and V optimal design criteria
- There is no real need for online feedback in the active linear regression problem and the training set problem can be cast as a subset selection problem performed offline.
- This problem is NP hard and thus approximate solutions are needed.

Gaussian Process Classification

Gaussian Process Classification (GPC) is a powerful, non-parametric kernel-based model.

$$f \sim GP(\mu(\cdot), k(\cdot, \cdot))$$

$$y|x, f \sim Bernoulli\left(\Phi\left(f_x\right)\right)$$

- f is a function of a feature point x and is assigned a Gaussian process prior with mean $\mu(\cdot)$ and covariance function $k(\cdot, \cdot)$.
- The label y is Bernoulli distributed with probability $\Phi(f_x)$, where Φ is the Gaussian CDF.

Gaussian Process Classification

Gaussian Process Classification (GPC) is a powerful, non-parametric kernel-based model.

$$f \sim GP(\mu(\cdot), k(\cdot, \cdot))$$

$$y|x, f \sim Bernoulli\left(\Phi\left(f_x\right)\right)$$

- f is a function of a feature point x and is assigned a Gaussian process prior with mean $\mu(\cdot)$ and covariance function $k(\cdot, \cdot)$.
- The label y is Bernoulli distributed with probability $\Phi(f_x)$, where Φ is the Gaussian CDF.

Problem

Direct computation of the posterior in GPC is intractable.

Variational Inference

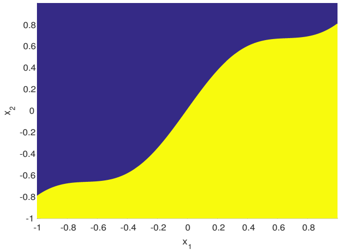
- Variational inference is a technique used in probabilistic modeling to approximate complex probability distributions that are difficult or impossible to calculate exactly.
- The goal of variational inference is to find an approximation, $q^*(\theta)$ from a parametric family $\mathbb{Q}$, to the true distribution, $p(\theta|z^{n-1})$, that is as close as possible to the true distribution, but is also computationally tractable.

$$q^*(\theta) = \arg\min_{q \in \mathbb{Q}} D_{KL}\left(q(\theta)||p(\theta|z^{n-1})\right)$$

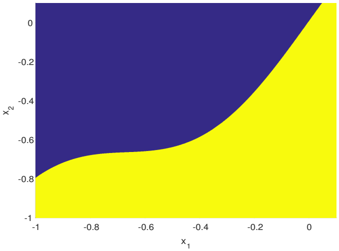
- UAL for GPC uses Expectation Propagation.

Synthetic Data

- Two dimensional feature vectors with binary labels: yellow color indicates '-1' label and blue is '+1'

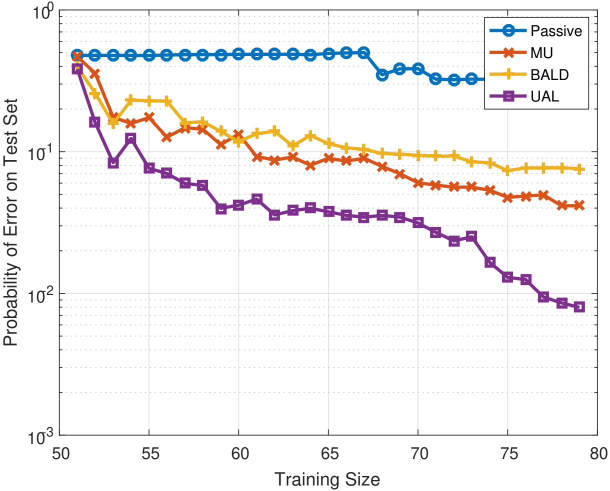


Training set



Test Set

Simulation Results



Part 2: The Individual Setting

Includes:

- Shayovitz Shachar, and Meir Feder. "Active Learning for Individual Data via Minimal Stochastic Complexity." 2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2022. [SF22]
- Shayovitz Shachar, and Meir Feder. "Active Learning via Predictive Normalized Maximum Likelihood Minimization," in IEEE Transactions on Information Theory, vol. 70, no. 8, Aug. 2024, [SF24]
- Shayovitz Shachar, Koby Bibas, and Meir Feder. "Deep Individual Active Learning: Safeguarding against Out-of-Distribution Challenges in Neural Networks." Entropy 26.2 (2024): 129. [SBF24]

Active Learning Criteria

- Maximum Uncertainty (MU)
  - $\hat{x}_n = \arg\max_{x_n} H\left(y_n | x^n, y^{n-1}\right)$.
  - Sensitive to noise.
- Bayesian Active Learning by Disagreement (BALD) [HHGL11]
  - $\hat{x}_n = \arg\max_{x_n} I\left(\theta; y_n | x^n, y^{n-1}\right)$.
  - Focused on model estimation and not prediction.
- Universal Active Learning (UAL) [SF21]
  - $\hat{x}_n = \arg\min_{x_n} I\left(\theta; y | x, x^n, y^n\right)$.
  - Derived using the Capacity - Redundancy Theorem.
  - Takes into account the un-labelled test set.

Active Learning Criteria

- Maximum Uncertainty (MU)
  - $\hat{x}_n = \arg\max_{x_n} H\left(y_n | x^n, y^{n-1}\right)$.
  - Sensitive to noise.
- Bayesian Active Learning by Disagreement (BALD) [HHGL11]
  - $\hat{x}_n = \arg\max_{x_n} I\left(\theta; y_n | x^n, y^{n-1}\right)$.
  - Focused on model estimation and not prediction.
- Universal Active Learning (UAL) [SF21]
  - $\hat{x}_n = \arg\min_{x_n} I\left(\theta; y | x, x^n, y^n\right)$.
  - Derived using the Capacity - Redundancy Theorem.
  - Takes into account the un-labelled test set.

Data assumed to follow some parametric distribution

Cannot be validated for real world data!

Learning in Individual Setting

Assumptions

- **No underlying parametric distribution.**
- Training pool: $z^n = (x^n, y^n)$
- Test pair: (x, y)
  - x can be accessed.
  - y is not available (privacy preserving).
- Probabilistic learners: $q\,(y|x)$.
- Log-loss cost function: $-\log\left(q\left(\cdot|x, z^n\right)\right)$.

Learning in Individual Setting

Assumptions

- **No underlying parametric distribution.**
- Training pool: $z^n = (x^n, y^n)$
- Test pair: (x, y)
  - x can be accessed.
  - y is not available (privacy preserving).
- Probabilistic learners: $q\,(y|x)$.
- Log-loss cost function: $-\log\left(q\left(\cdot|x, z^n\right)\right)$.

Fundamental Problem

Minimizing the log-loss in the individual setting is ill-posed.

Learning in Individual Setting

Define a hypothesis class:

$$P_\Theta = \{p\left(y|x,\theta\right)|\theta \in \Theta\}$$

Define the learning problem:

$$R\left(x; z^n\right) = \min_q \max_{y \in \mathbb{Y}} \log\left(\frac{p\left(y|x,\hat{\theta}\right)}{q\left(y|x,z^n\right)}\right)$$

where $p\left(y|x,\hat{\theta}\right) \in P_\Theta$ and the best learner is:

$$\hat{\theta} = \arg\max_{\theta\in\Theta}\left[\sum_{i=1}^{n}\log p\left(y_i|x_i,\theta\right) + \log p\left(y|x,\theta\right) + \log\left(w\left(\theta\right)\right)\right]$$

Predictive Normalized Maximum Likelihood (pNML)

Theorem ([FF18])

The universal learner, q_{pNML}, which minimizes $R\left(x; z^n\right)$:

$$q_{pNML}(y|x, z^n) = \frac{p\left(y|x, \hat{\theta}\right)}{\sum_y p\left(y|x, \hat{\theta}\right)}$$

$$R\left(x; z^n\right) = \log \sum_{y \in \mathbb{Y}} p\left(y|x, \hat{\theta}\right)$$

Note that any estimation algorithm can be used to estimate θ and the same Theorem will hold for the respective $\hat{\theta}$.

Active Learning in Individual Setting

What is a "good" training set, z^n?

Small $R(x; z^n)$ on as many test features x as possible!

Active Learning in Individual Setting

What is a "good" training set, z^n?

Small $R(x; z^n)$ on as many test features x as possible!

Problem

y^n is not available a-priori and thus optimizing over z^n is not possible!

Active Learning in Individual Setting

Minimizing the worst case training set

Find x^n which minimize average regret for the worst case y^n:

$$C_n^A = \min_{x^n \in \mathbb{X}^n} \max_{y^n \in \mathbb{Y}^n} \sum_x R\left(x; z^n\right)$$

Equivalently [FF18]:

Individual Active Learning (IAL)

$$C_n = \min_{x^n \in \mathbb{X}^n} \max_{y^n \in \mathbb{Y}^n} \sum_x \log \sum_{y \in \mathbb{Y}} p\left(y | x, \hat{\theta}\left(x, y, z^n\right)\right)$$

Active Learning in Individual Setting

Sequential Scheme

- For most hypothesis classes, the batch is exponentially hard to solve.
- A simpler approach is the sequential form:

$$C_{n|n-1} = \min_{x_n} \max_{y_n} \sum_{x} \log \left(\sum_{y} p\left(y|x, \hat{\theta}\right) \right)$$

Active Learning in Individual Setting

- In the next slides we examine IAL for different hypothesis classes:
  - One dimensional Barrier
  - Linear Regression
  - Gaussian Process Classification
- It will be shown that IAL coincides with known class specific criteria and thus is a unified framework for active learning!

One Dimensional Barrier - Separable Data

The 1-dimensional barrier hypotheses class is defined as:

$$p(y = 1|x, \theta) = \begin{cases} \alpha & \text{if } x > \theta \\ 1 - \alpha & \text{otherwise} \end{cases}$$

where:

- $\alpha \in \{0, 1\}$
- Input $x \in [0, 1]$
- Output $y \in \{0, 1\}$
- Unknown threshold $\theta \in [0, 1]$.

One Dimensional Barrier - Separable Data

The 1-dimensional barrier hypotheses class is defined as:

$$p(y = 1|x, \theta) = \begin{cases} \alpha & \text{if } x > \theta \\ 1 - \alpha & \text{otherwise} \end{cases}$$

where:

- $\alpha \in \{0, 1\}$
- Input $x \in [0, 1]$
- Output $y \in \{0, 1\}$
- Unknown threshold $\theta \in [0, 1]$.

Theorem ([SF24])

For 1 dimensional linearly separable data, IAL induces a selection policy which coincides with binary search.

Proof Outline

- The greedy IAL can be written as

$$C_{n|n-1} = \min_{x_n \in \mathbb{X}} \max_{y_n \in \mathbb{Y}} \int_{x \in \mathbb{X}} \log \sum_{y \in \mathbb{Y}} p\left(y|x, \hat{\theta}^n\right) dx$$

where y, x and $\hat{\theta}^n$ are the test label, feature and maximum likelihood estimation based on training and test data respectively

$$\hat{\theta}^n = \arg\max_{\theta \in \Theta} p\left(y^n, y|x^n, x, \theta\right).$$

- We can write the likelihood for z^{n-1} as

$$p\left(y^{n-1}|x^{n-1}, \theta\right) \sim \mathbb{1}\left(\theta \geq \theta_{min}^{n-1}\right) \mathbb{1}\left(\theta < \theta_{max}^{n-1}\right)$$

where θ_{min}^{n-1} and θ_{max}^{n-1} represent the support of the posterior on θ given x^{n-1}, y^{n-1}.

Proof Outline

- For each unlabelled pool point x_n, the updated likelihood window function gets split based on y_n.
- For $y_n = 1 - \alpha$:

$$\int_0^1 \log \sum_{y=0}^{1} p\left(y|x, \hat{\theta}^n\right) dx = |x_n - \theta_{max}^{n-1}|$$

- For $y_n = \alpha$:

$$\int_0^1 \log \sum_{y=0}^{1} p\left(y|x, \hat{\theta}^n\right) dx = |\theta_{min}^{n-1} - x_n|.$$

- Therefore,

$$C_{n|n-1} = \min_{x_n \in \mathbb{X}} \max\{|x_n - \theta_{max}^{n-1}|, |\theta_{min}^{n-1} - x_n|\}.$$

- The point x_n which minimizes the maximal length is the mid point of the interval $\left[\theta_{min}^{n-1}, \theta_{max}^{n-1}\right]$.

IAL for Linear Regression

Theorem ([SF24])

Consider the hypothesis class:

$$P_\Theta = \{p\left(y|x,\theta\right)|\theta \in \mathbb{R}^d\}$$

$$p\left(y|x,\theta\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(y - x^T\theta\right)^2\right)$$

Then the following upper bound holds (with equality for high SNR):

$$C_n \le \min_{X_n} \mathrm{Tr}\left(X_{test}^T X_{test}\left(X_n^T X_n + \frac{\sigma^2}{\lambda} I\right)^{-1}\right)$$

- $\hat{\theta}$ *is computed using L2 regularization with a factor* λ.

Gaussian Process Classification

- The IAL for GPC:

$$C_{n|n-1} = \min_{x_n \in \mathbb{X}} \max_{y_n \in \mathbb{Y}} \sum_{v \in \mathbb{V}} \int_{u \in \mathbb{U}} p\left(v | \hat{f}_u\right) du$$

- The MAP estimation for the model parameter vector, $\underline{f}$ (for all possible feature points):

$$\hat{\underline{f}} = \arg\max_{\underline{f}} p\left(y_n | f_{x_n}\right) p\left(v | f_u\right) p(\underline{f} | x^{n-1}, y^{n-1})$$

- $p(\underline{f})$ is a Gaussian process which acts as a regularization prior over the latent vector $\underline{f}$.
- Given a training set, the posterior over f becomes non-Gaussian and too complicated to work with.

EP Approximation

- Due to the likelihood factorization, no need to re-compute EP with all the training data for every training and test points.
- Assume $q\left(\underline{f}|y^{n-1}, x^{n-1}\right)$ is a Gaussian distribution.
- EP approximates $p\left(f_{x_n}, f_u|y^n, x^n, u, v\right)$ as a 2-Dimensional Gaussian:
  - $q\left(\underline{f}|y^{n-1}, x^{n-1}\right)$ as a prior.
  - The new data points $[u, v]$ and $[x_n, y_n]$.
- The MAP estimators $\hat{f}_{x_n}^{y_n}$ and $\hat{f}_u^v$ are computed based on:

$$\hat{f}_{x_n}^{y_n}, \hat{f}_u^v = \arg\max_{f_{x_n}, f_u} q\left(f_{x_n}, f_u|y^n, x^n, u, v\right)$$

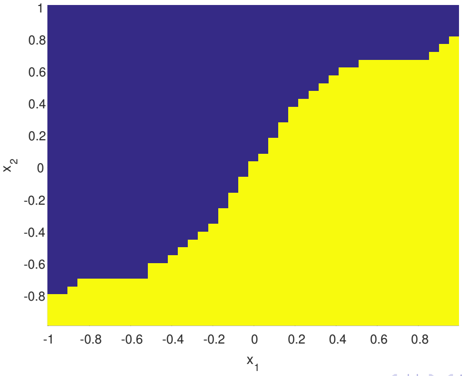
- These are used to compute the average regret.

Algorithm

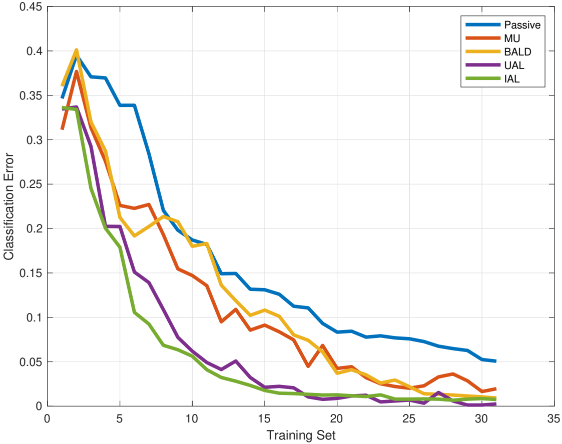
Input: Training Data $\{x^{n-1}, y^{n-1}\}$
Training and Test samples $\{x_i\}_{i=1}^{N}$ and $\{u_i\}_{i=1}^{K}$.
Output: Next data point for labelling - x_n
procedure IAL - GPC
  Set $D = [x^{n-1}, y^{n-1}]$
  Set EP prior $q_{prior}^{EP} = \mathcal{N}\left(\underline{f}|0, \log \lambda I\right)$
  Run EP: $q^{n-1}\left(\underline{f}\right) = EP(D, q_{prior}^{EP})$
  $\mathbf{S} = zeros(N, |\mathbb{Y}|)$
  for $i \leftarrow 1$ to N **do**
    for $j \in \mathbb{Y}$ **do**
      for $k \leftarrow 1$ to K **do**
        for $l \in \mathbb{Y}$ **do**
          Set $D = [x_i, j, u_k, l]$
          Set EP prior $q_{prior}^{EP} = q^{n-1}\left(\underline{f}\right)$
          $\mathcal{N}\left(f_{u_k}, f_{x_i}|\hat{\mu}, \hat{V}\right) = EP(D, q_{prior}^{EP})$
          $\hat{f}_{u_k}^{l}, \hat{f}_{x_i}^{j} = \hat{\mu}$
          $\mathbf{S}(i,j) = \mathbf{S}(i,j) + \Phi\left(l \cdot \hat{f}_{u_k}^{l}\right)$
  $\hat{i} =_i \max_j \mathbf{S}$
  $x_n = x_{\hat{i}}$

Synthetic Data

- The training pool is a square in the two dimensional plane and divides it to two non overlapping regions.
- The test set is a smaller sub-set with corners at four points (-1, -0.5), (1, -0.5), (-1, -0.25), and (1, -0.25).



Classification Error



USPS Data Set

- USPS hand-written digits data set.
- Total of 9298 handwritten single digits between 0 and 9.
- Test and train distributions do not necessarily belong to the GPC hypothesis class.
- Classify the digit 7 versus 9 (graphically similar → hard to classify)

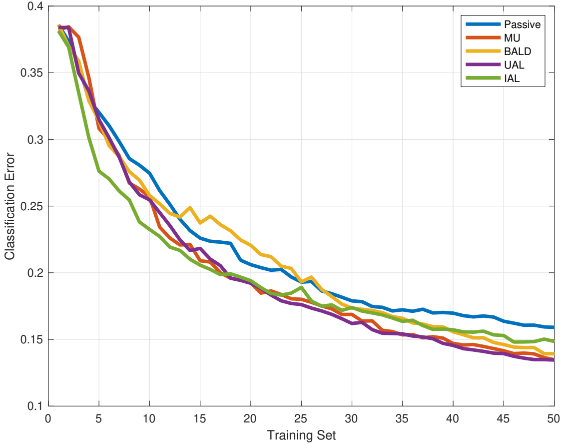
Algorithm Parameters

Dimension reduction (for EP complexity):

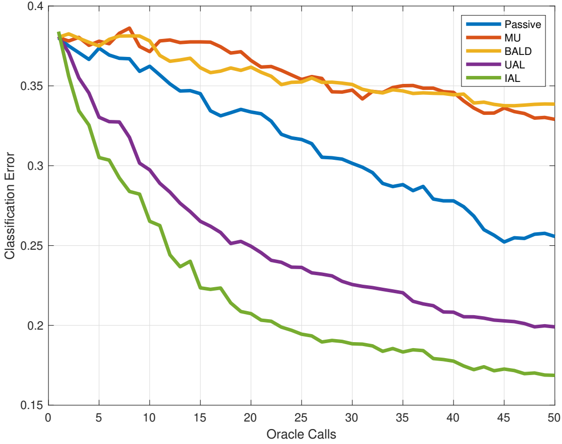
- PCA is applied using the un-labelled training data
- After centering and PCA, the eigen-vectors corresponding to the 65% largest Eigen-values of the PCA are used.

Parameter	Value
Passive Regularization λ	5
MU Regularization λ	5
BALD Regularization λ	5
UAL Regularization λ	5
IAL Regularization λ	5
Initial training set	2 examples (1 for each class)
Unlabelled test set	5 random test features

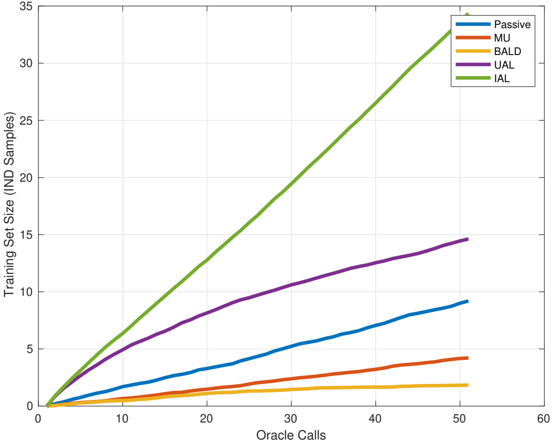
Error Probability: Hand-written digits data set, IND



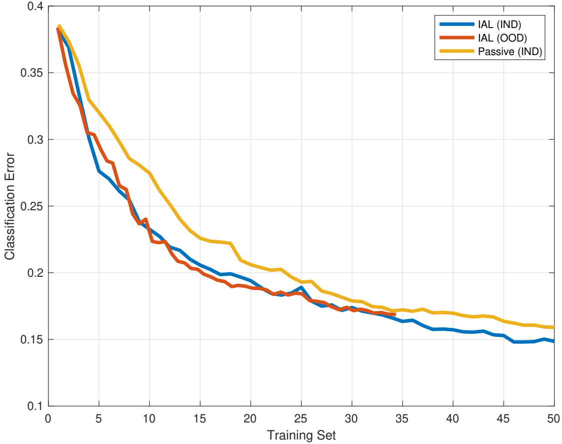
Error Probability: Hand-written digits data set, OOD



Training set size vs Oracle calls



IAL selects informative IND samples → not just an OOD detector



Active Learning with Deep Neural Networks (DNN)

DNNs are the most dominant machine learning hypothesis class in practical use.

Challenge

- The computational bottleneck for DNNs is the long training time.
- Direct application of IAL for DNNs:
  - infeasible for real world large-scale data since it requires training the entire model for each possible training and test points!
  - Previous work approximated the pNML for DNNs by fine-tuning the last layer for each test input and label combination.
  - Does not work well in practice for AL since AL affects all layers.

System Model

- We define the hypothesis class in this case as follows:

$$p\left(y|x,\theta\right) = \mathit{softmax}\left(f_\theta\left(x\right)\right)$$

  where θ are all the weights and biases of the network and $f_\theta\left(x\right)$ is the model output before the last softmax layer.
- The MAP estimation for θ is:

$$\hat{\theta} = \arg\max_\theta p\left(y^n, y|x^n, x, \theta\right) p(\theta),$$

  where the prior $p(\theta)$ acts as a regularizer.

Factorization Trick

- Using the following factorization, train the network using x^{n-1}, y^{n-1}:

$$\hat{\theta} = \arg\max_{\theta} p\left(y|x,\theta\right) p\left(y_n|x_n,\theta\right) p\left(\theta|y^{n-1},x^{n-1}\right)$$

- $p\left(\theta|y^{n-1},x^{n-1}\right)$ is not dependent on the test data (x,y) and the evaluated labeling candidate (x_n,y_n).
- No need to retrain the network for every (x,y) and (x_n,y_n), just run forward passes (inference) $p\left(y|x,\theta\right)$ and $p\left(y_n|x_n,\theta\right)$.
- Significant reduction in computational complexity, as the number of possible points x_n can be huge.

Posterior Approximation

Problem

- The posterior $p\left(\theta|y^{n-1}, x^{n-1}\right)$ is intractable!
- For GPC, this posterior was approximated using EP:
  - Computing EP with every training and test points on a DNN is computationally prohibitive.
  - EP is based on a single mode Gaussian approximation while the $p\left(\theta|y^{n-1}, x^{n-1}\right)$ is multi-modal $\rightarrow$ empirically didn't produce good results for DNN's.
- A different approach for approximating the posterior with low complexity is needed.

MC Dropout

- MC (Monte Carlo) Dropout [GG16] is a technique used in deep learning to estimate the uncertainty of a neural network's predictions.
- An estimate of the network's uncertainty is performed by running multiple forward passes with different dropout masks.
- The variance of the outputs across the different passes gives an estimate of the uncertainty of the prediction.
- We opted to use MC-Dropout , due to its computational simplicity and favorable performance.

MC Dropout

- Dropout training applied before every layer is mathematically equivalent to minimizing the KL divergence between the weight posterior of the full network and a parametric distribution, $q(\theta)$ which is controlled by a set of Bernoulli random variables with the dropout probability [GG16].
- We replace the full posterior, $p\left(\theta|y^{n-1}, x^{n-1}\right)$, with the approximate distribution $q\left(\theta|y^{n-1}, x^{n-1}\right)$.
- Therefore,

$$\hat{\theta} \approx \arg\max_{\theta} p(y|x,\theta)\, p(y_n|x_n,\theta)\, q\left(\theta|y^{n-1}, x^{n-1}\right)$$

MC Dropout

- Dropout training applied before every layer is mathematically equivalent to minimizing the KL divergence between the weight posterior of the full network and a parametric distribution, $q(\theta)$ which is controlled by a set of Bernoulli random variables with the dropout probability [GG16].
- We replace the full posterior, $p\left(\theta | y^{n-1}, x^{n-1}\right)$, with the approximate distribution $q\left(\theta | y^{n-1}, x^{n-1}\right)$.
- Therefore,

$$\hat{\theta} \approx \arg\max_{\theta} p(y|x,\theta)\, p(y_n|x_n,\theta)\, q\left(\theta | y^{n-1}, x^{n-1}\right)$$

Problem

$q\left(\theta | y^{n-1}, x^{n-1}\right)$ is still too complex to analytically compute.

Deep Individual Active Learning (DIAL)

- Instead of computing $q\left(\theta|y^{n-1}, x^{n-1}\right)$, we propose to sample M weights, $\{\theta_m\}_{m=1}^{M}$ (just by running dropout in inference) from $q\left(\theta|y^{n-1}, x^{n-1}\right)$ and find $\hat{\theta}$ among all the different samples.
- Another simplification:

$$\hat{\theta} = \arg\max_{\{\theta_m\}_{m=1}^{M}} p\left(y|x, \theta_m\right) p\left(y_n|x_n, \theta_m\right)$$

- In short, it means running M forward passes with Dropout ON and taking the softmax output for $p\left(y|x, \theta_m\right)$ and $p\left(y_n|x_n, \theta_m\right)$ (using same seed)

DIAL Algorithm

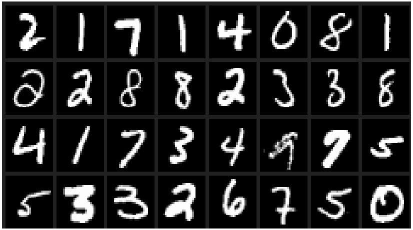
Input Training set z^{n-1}, unlabeled pool and test samples $\{x_i\}_{i=1}^{N}$ and $\{x_k\}_{k=1}^{K}$.
Output Next data point for labeling $\hat{x}_i$
Run MC-Dropout using z^{n-1} to get $\{\theta_m\}_{m=1}^{M}$
$\mathbf{S} = zeros(N, |\mathbb{Y}|)$
for $i \leftarrow 1$ to N **do**
    for $y_i \in \mathbb{Y}$ **do**
        for $k \leftarrow 1$ to K **do**
            $\Gamma = 0$
            for $y_k \in \mathbb{Y}$ **do**
                $\hat{\theta} = \text{argmax}_{\theta_m} p\left(y_k | x_k, \theta_m\right) p\left(y_i | x_i, \theta_m\right)$
                $\Gamma = \Gamma + p\left(y_k | x_k, \hat{\theta}\right)$
            $\mathbf{S}\left(i, y_i\right) = \mathbf{S}\left(i, y_i\right) + \log \Gamma$
$\hat{x}_i = \text{argmin}_{x_i} \max_{y_i} \mathbf{S}$

Experiments Datasets

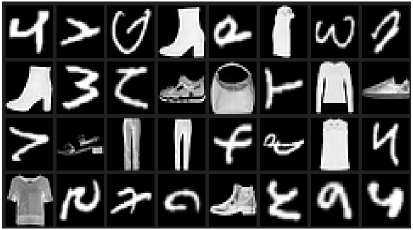
- **The MNIST dataset** consists of 28x28 grayscale images of handwritten digits, with 60K images for training and 10K images for testing.
- **The EMNIST dataset** is a variant of the MNIST dataset that includes a larger variety of images. It consists of 240K images with 47 different labels.
- **The CIFAR10 dataset** consists of 60K 32x32 color images in 10 classes.
- **Fashion MNIST** 70K images with each image is 28x28 grayscale pixels.
- **The SVHN dataset** contains 600K real-world images with digits and numbers in natural scene images collected from Google Street View.

Training and Test Data

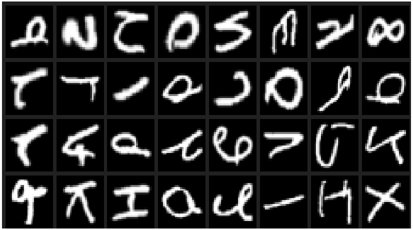
MNIST and OOD images



MNIST test images

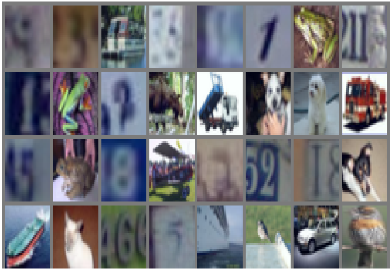


EMNIST and OOD images

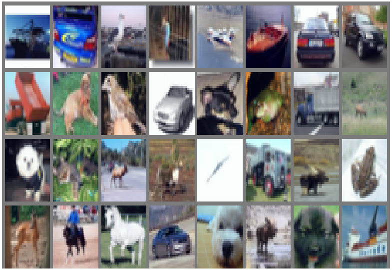


EMNIST test images

Training and Test Data



CIFAR10 and OOD images



CIFAR10 test images

Active Learning Algorithms

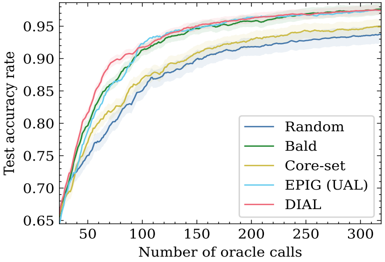
- **The Random sampling** selects samples to label randomly, without considering any other criteria.
- **The Bayesian Active Learning by Disagreement (BALD)** [GIG17] calculates the mutual information between the model's predictions and the model's parameters.
- **The Core-set** algorithm aims to find a small subset from a large labeled data-set such that a model learned from this subset will perform well on the entire data-set.
- **The Expected Predictive Information Gain (EPIG)** method [SKF+23] was motivated by BALD's weakness in prediction-oriented settings. This acquisition function directly targets a reduction in predictive uncertainty on inputs of interest by utilizing the unlabelled test set.

MNIST experimental results

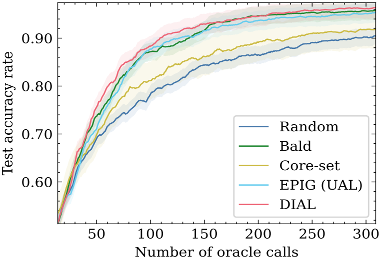
- We considered a model consisting of two blocks of convolution, dropout, max-pooling, and ReLu, with 32 and 64 5x5 convolution filters.
- These blocks are followed by 2 fully connected layers that include dropout between them.
- The layers have 128 and 10 hidden units respectively.
- The dropout probability was set to 0.5 in all three locations.
- For BALD, EPIG, and DIAL we used 100 dropout iterations and employed the criterion on 512 random samples from the unlabeled pool.
- The 256 samples with the highest score are taken [2].

[2] Significant room for improvement!

MNIST Results



MNIST



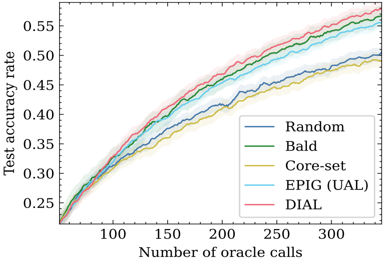
MNIST with OOD

MNIST with OOD: Number of Oracle Calls at x% accuracy

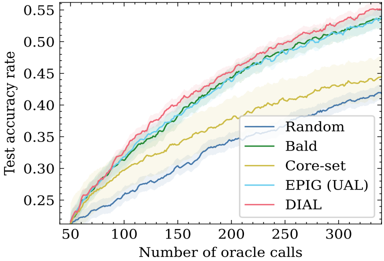
Methods	85% Acc.	75% Acc.	65% Acc.
Random	145	73	36
Core-set	117	61	33
BALD	83	51	32
DIAL	**73 (-12.1%)**	**48 (-5.9%)**	**30 (-6.2%)**

EMNIST experimental results

Larger model than MNIST consisting of three blocks of convolution, dropout, max-pooling, and ReLu.



EMNIST



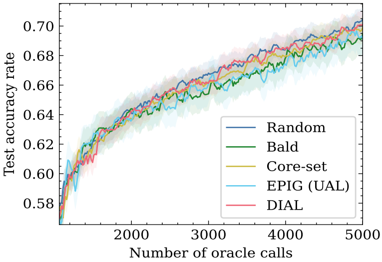
EMNIST with OOD

EMNIST with OOD: Number of Oracle Calls at x% accuracy

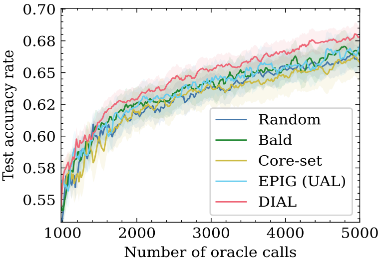
Methods	40% Acc.	30% Acc.	25% Acc.
Random	281	140	80
Core-set	221	96	62
BALD	154	85	**59**
DIAL	**138 (-10.4%)**	**84 (-1.2%)**	**59 (0%)**

CIFAR10 experimental results

- For the CIFAR10 data-set, we utilized ResNet-18 with acquisition size of 16 samples.
- We used 1K initial training set size.

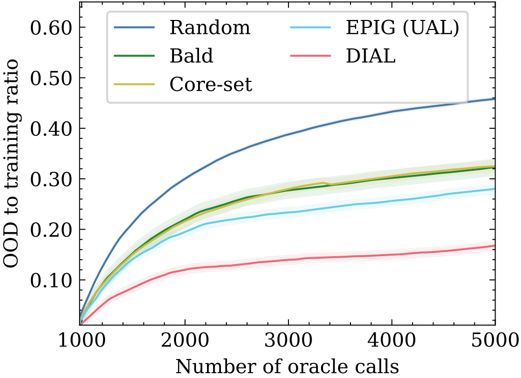


CIFAR10



CIFAR10 with OOD

CIFAR10 experimental results



CIFAR10 in the presence of OOD samples: number of Oracle calls at specific accuracy rate values

Methods	**66% Acc.**	**62% Acc.**	**58% Acc.**
Random	3956	1828	1220
Core-set	4468	1844	1412
BALD	4020	1636	1202
EPIG	3636	1700	1108
DIAL	**3076 (−15.4%)**	**1556 (−4.9%)**	**1060 (−4.3%)**

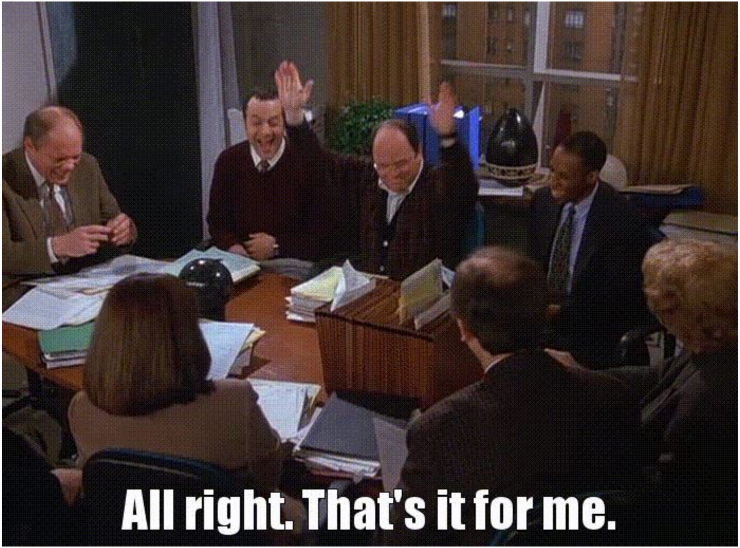
Future Direction: Batch Active Learning

- Practical active learning choose a batch of samples and not one sample at a time.
- Selecting the highest-score batch using IAL or UAL gives bad performance since samples with high correlation are chosen.
- Consider the relationship between the selected samples and the overall composition of the batch, which may lead to even further improvements in performance.

Summary

- Proposed AL criteria for the stochastic and individual settings:
  - Both take into account a small **un-labelled** sample of the test set.
  - Unified active learning framework for a variety of hypothesis classes (binary classification and linear regression).
- Proposed an AL scheme for Deep Neural Networks (DIAL).
  - Scheme is based on a low complexity uncertainty quantification approach (MC-Dropout).
  - In the presence of out-of-distribution data, DIAL reduces the required number of Oracle calls by up to 15.4%, 10.4%, and 12% for CIFAR10, EMNIST, and MNIST datasets respectively.
- Proposed a near-optimal, low complexity, algorithm (SPM) for active learning of high dimensional linear separators with various label noise models.

Thank You!



- Epoch AI, **Trends in training dataset sizes**, 2024, Accessed: 2024-07-14.
- Yaniv Fogel and Meir Feder, **Universal batch learning with log-loss**, 21–25.
- Yarin Gal and Zoubin Ghahramani, **Dropout as a bayesian approximation: Representing model uncertainty in deep learning**, international conference on machine learning, PMLR, 2016, pp. 1050–1059.
- Yarin Gal, Riashat Islam, and Zoubin Ghahramani, **Deep bayesian active learning with image data**, 1183–1192.

- Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel, **Bayesian active learning for classification and preference learning**, arXiv preprint arXiv:1112.5745 (2011).
- Friedrich Pukelsheim, **Optimal design of experiments**, SIAM, 2006.
- Shachar Shayovitz, Koby Bibas, and Meir Feder, **Deep individual active learning: Safeguarding against out-of-distribution challenges in neural networks**, Entropy **26** (2024), no. 2, 129.
- Ofer Shayevitz and Meir Feder, **Communication with feedback via posterior matching**, 391–395.

- Shachar Shayovitz and Meir Feder, **Minimax active learning via minimal model capacity**, 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2019, pp. 1–6.
- ______, **Universal active learning via conditional mutual information minimization**, IEEE Journal on Selected Areas in Information Theory **2** (2021), no. 2, 720–734.
- ______, **Active learning for individual data via minimal stochastic complexity**, 2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2022, pp. 1–5.

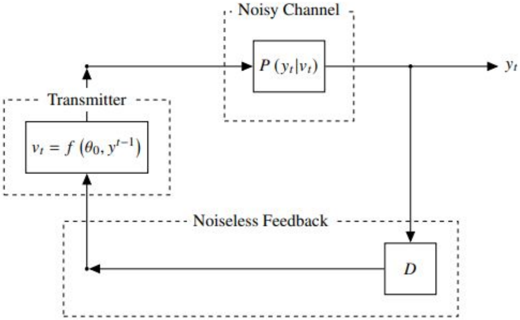
———, **Active learning via predictive normalized maximum likelihood minimization**, IEEE Transactions on Information Theory **70** (2024), no. 8, 5799–5810.

Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth, **Prediction-oriented bayesian active learning**, International Conference on Artificial Intelligence and Statistics, PMLR, 2023.

Active Learning for Linear Binary Classification in the Stochastic Setting

Communication over Noisy Channels with Noiseless Feedback

- Feedback cannot increase the capacity of memoryless channels
- Can boost reliability and simplify transmission schemes.



Posterior Matching Scheme

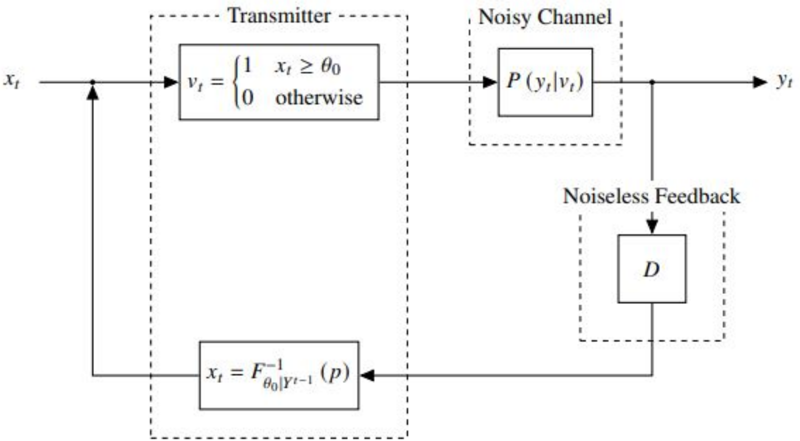
- Capacity achieving scheme proposed by Shayevitz and Feder [SF07], suitable for any memory-less channel $P(Y|V)$.
- Information bits are encoded to a point θ_0 in the interval $[0, 1]$.
- Next symbol v_t is computed via:

$$v_t = F_V^{-1}\left(F_{\theta_0|Y^{t-1}}\left(\theta_0|y^{t-1}\right)\right)$$

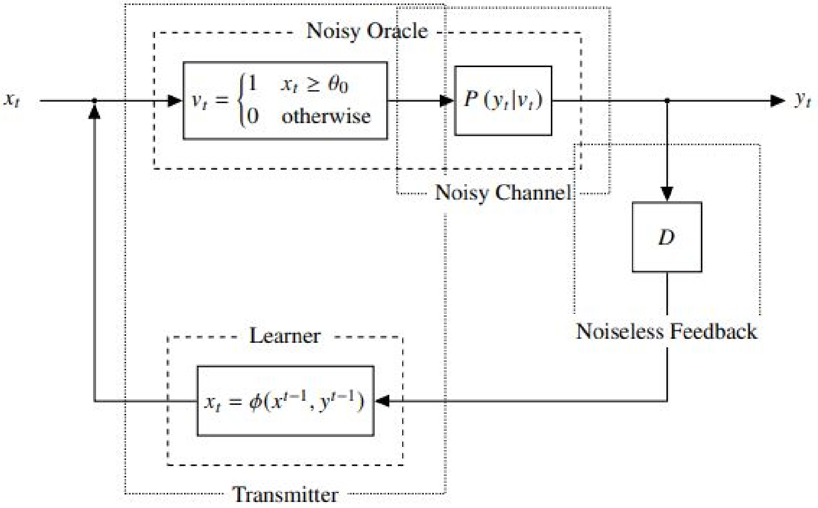
- The estimation error on θ_0 drops exponentially fast.
- For a binary valued v_t, with $V \sim Ber(p)$, the PM scheme reduces to:

$$v_t = \begin{cases} 1, & \text{if } \theta_0 > F^{-1}_{\theta|y^{t-1}}(p) \\ 0, & \text{otherwise} \end{cases}$$

Posterior Matching Scheme



Active Learning as a Communication Problem



Active Learning for 1d Classifier

- The Idea is to look at the problem as communicating θ_0 over a noisy channel.
- Pass as much information bits on θ_0 using few channel uses and correctly decode θ_0.
- If we choose $\phi\left(x_t|x^{t-1}, y^{t-1}\right) = F^{-1}_{\theta|y^{t-1}}(p)$, we achieve capacity!
- Using this scheme we get an exponential decay on minimax redundancy with the channel capacity as the decay factor!

High Dimensional Linear Separators

- Features $\underline{x} \in \mathbb{R}^d$ satisfy $\|\underline{x}\| \leq R$ with uniform $p(\underline{x})$.
- The hypotheses class contains all possible hyper-planes with normal vector $\underline{w}$ and threshold b.
- The relation between feature $\underline{x}$ and **clean** label v is defined as,

$$p(v|\underline{x}, \underline{w}, b) = \begin{cases} 1 & \textit{if } \underline{\mathrm{w}}^T\underline{x} > b \\ 0 & \textit{otherwise} \end{cases}$$

- v passes through a discrete memory-less channel $p(y|v)$ and produces the noisy label - y.

Successive Posterior Matching (SPM)

Question

How do we use Posterior Matching for the high dimensional problem?

Successive Posterior Matching (SPM)

Question

How do we use Posterior Matching for the high dimensional problem?

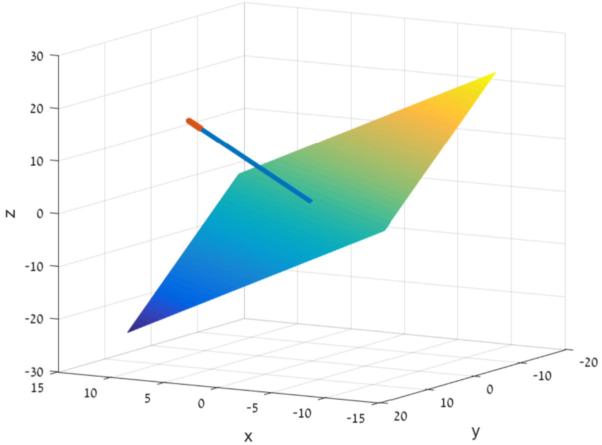
SPM Idea

- True classifier is fully described by its normal vector.
- The idea is to successively localize the spherical coordinates of the normal vector $\underline{w}$ using Posterior Matching.
- Each coordinate lives on the arc: $\theta_i \in [0, \pi]$.
- The intersection of the hyper-plane and the arc is the barrier between classification regions.
- For each spherical coordinate we have a noisy one dimensional barrier problem.

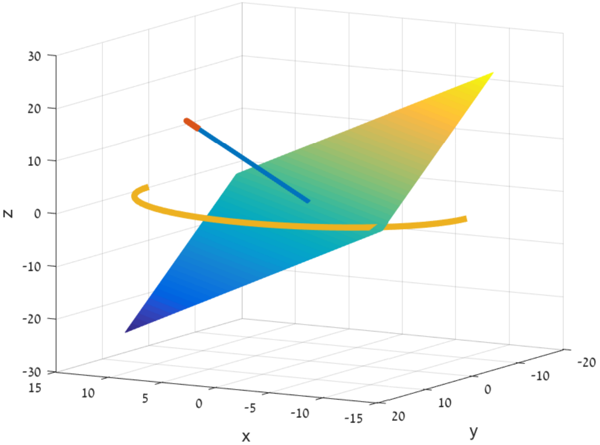
Successive Posterior Matching (SPM)

Init: $\underline{\hat{\theta}} = [\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, ..., \frac{\pi}{2}]$,
Init: $\forall i \in [1 : d-1], p(\theta_i) = Unif[0, \pi]$
for $i \leftarrow d-1$ to 1 **do**
    for $k \leftarrow 1$ to n **do**
        $\hat{\theta}_i = F^{-1}_{\theta_i | \underline{x}^i_{1:k-1}, y^i_{1:k-1}} \left(\frac{p-0.5}{p+q-1} \right)$
        $\underline{x}^i_k = [\Pi^{d-1}_{l=1} \sin(\hat{\theta}_l), \cos(\hat{\theta}_{d-1}) \Pi^{d-2}_{l=1} \sin(\hat{\theta}_l)$
            $, ..., \cos(\hat{\theta}_i) \Pi^{i-1}_{l=1} \sin(\hat{\theta}_l), ..., \cos(\hat{\theta}_1)]$
        $y^i_k = Label(\underline{x}^i_k)$
        Update $p(\theta_i | \underline{x}^i_{1:k}, y^i_{1:k})$
    $\hat{\theta}_i = \hat{\theta}_i + \frac{\pi}{2}$

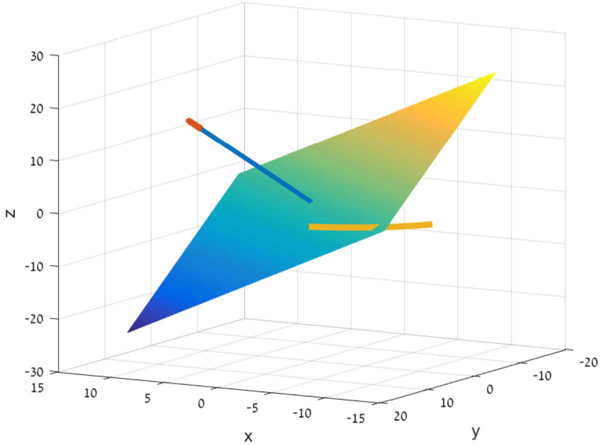
Classifier



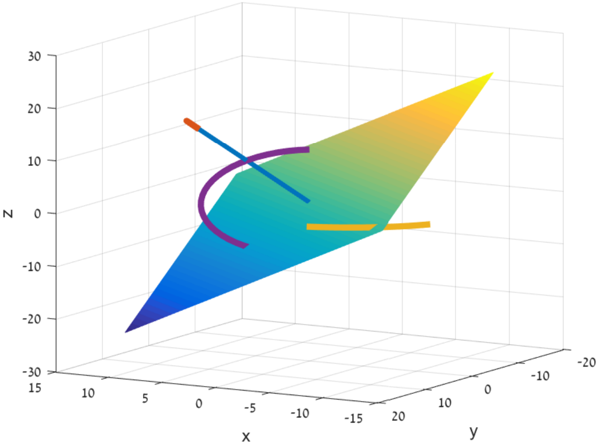
PM on Azimuth



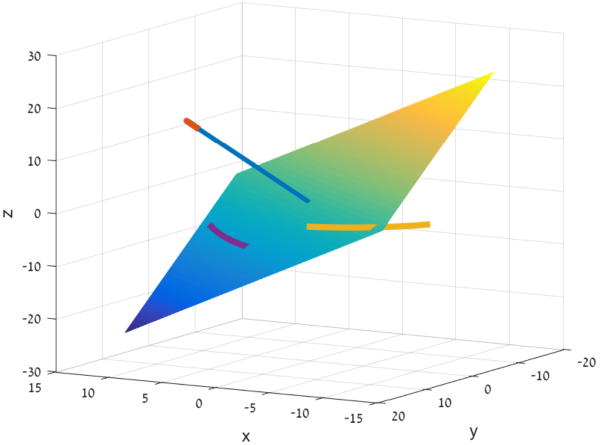
Estimated Barrier between Classification Regions



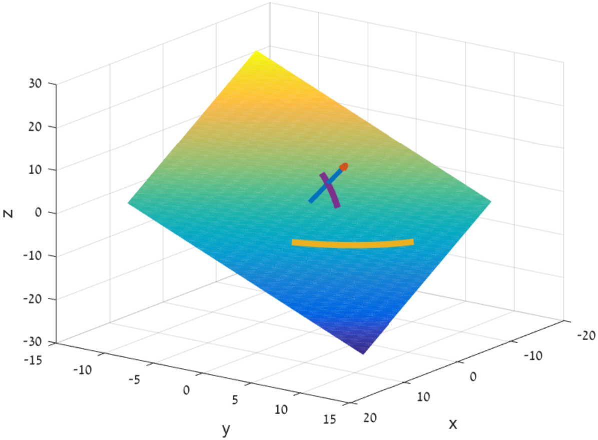
PM on Elevation



Estimated Barrier between Classification Regions



Estimated Normal Vector



Minimax Redundancy Convergence for SPM

Theorem [[SF19]]

Assuming:

- $\underline{x} \in \mathbb{R}^{d+1}$ with a bounded feature p.d.f.
- The Oracle is some member of a d dimensional homogeneous hyper-plane hypotheses class followed by a BAC.
- n is the total number of Oracle queries
- C_W is the Shannon capacity of the BAC with transition probability W.

Then, SPM produces a selection policy for which the minimax Redundancy decays exponentially fast to zero:

$$\lim_{n\to\infty} R = \lim_{n\to\infty} I(\theta; Y|X, \underline{x}^n, y^n) = \mathcal{O}\left(2^{-\frac{n}{d}C_W}\right)$$

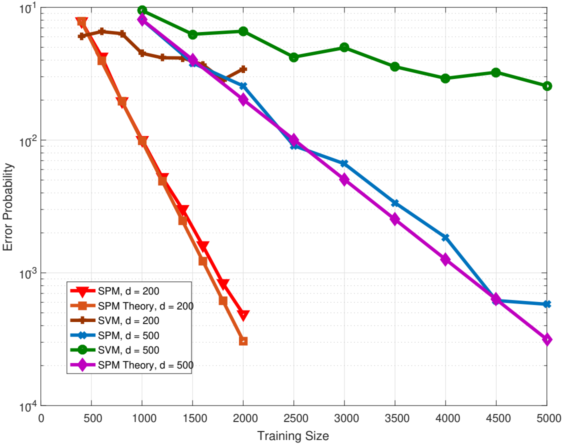
SPM Complexity

- $p(\theta_i | \underline{x}^i_{1:n}, y^i_{1:n})$ is updated at each iteration and the threshold point needs to be localized with very high accuracy.
- The Naïve approach would be to quantize the interval $[0, \pi]$ and compute the posterior.
- However, this approach is computationally expensive.
- Hypothesis class is a linear separator followed by a noisy binary channel, then the posterior of the intersection angle is a multiplication of different step functions.
- Only maintain a list of the step points and update the value of the posterior between these points.
- The number of points is exactly the number of training examples is linear with it.

Simulation Results

- SPM is compared to a widely used passive learning algorithm for learning hyper planes - Support Vector Machine (SVM) which is known to perform very well even in noisy conditions.
- A Monte Carlo simulation was implemented to estimate the error probability for an active learner based on SPM and a passive learner based on SVM.
- The comparison will be for feature spaces with $d = 200$ and $d = 500$ and using a BAC with $q = 10^{-2}$ and $p = 10^{-3}$.

Error Probability for BSC(10^{-2})



Error Probability for BSC