IAL

Interactive Adaptive Learning (IAL 2024)-Tutorial

Colocated with ECML PKDD 2024

Marek Herde, Tuan Pham Minh, Alaa Tharwat, Bernhard Sick

September 9, 2024

Time	Program	Presenter / Author
09:00–11:00	**Session 1: Tutorials & Poster Session**	
09:00–10:15	Tutorial Part I: Introduction to Uncertainty-Based Active Learning	A. Tharwat
10:15–11:00	Poster Session	
	Coffee Break (11:00–11:20)	
11:20–13:00	**Session 2: Tutorials**	
11:20–12:30	Tutorial Part II: Hands-on Pool-based Active Learning via scikit-activeml	M. Herde
12:30–13:00	Tutorial Part III: Towards Pool-based Active Learning with Error-prone Annotators	M. Herde
	Lunch Break (13:00–14:00)	

Lunch Break (13:00–14:00)

14:00–15:50	**Session 3: Keynote & Workshop Contributions**	
14:00–14:15	Deep Transfer Hashing for Adaptive Learning on Federated Streaming Data	M. Röder & F.-M. Schleif
14:15–14:35	General Reusability: Ensuring Long-Term Benefits of Deep Active Learning	P. Hahn, D. Huseljic, M. Herde & B. Sick
14:35–14:55	Suitability of Modern Neural Networks for Active and Transfer Learning in Surrogate-Assisted Black-Box Optimization	M. Holena & J. Koza
14:55–15:15	Amortized Active Learning for Nonparametric Functions	C.-Y. Li, M. Toussaint, B. Rakitsch & C. Zimmer
15:15–15:30	Towards Deep Active Learning in Avian Bioacoustics	L. Rauch, D. Huseljic, M. Wirth, J. Decke, B. Sick & C. Scholz
15:30–15:50	Active Learning with Physics-Informed Graph Neural Networks on Unstructured Meshes	J. Decke, A. Heinen, B. Sick & C. Gruhl

Coffee Break (15:50–16:20)

Coffee Break (15:50–16:20)

16:20–17:45	**Session 4: Workshop Contributions & Closing**	
16:20–16:40	Combining Large Language Model Classifications and Active Learning for Improved Technology-Assisted Review	M. P. Bron, B. Greijn, B. M. Coimbra, R. van de Schoot & A. Bagheri
16:40–17:00	Contextual kNN Ensemble Retrieval Approach for Semantic Postal Address Matching	E. M. Faraoun, N. Mellouli, M. Lamolle & S. Millot
17:00–17:45	Round Table & Closing	

IAL

Introduction to Uncertainty-based Active Learning

Alaa Tharwat

IAL

Agenda

- Introduction to active learning
- Sources of uncertainty
- Predictive uncertainty: Epistemic and Aleatoric uncertainty

IAL

Agenda

- Introduction to active learning
- Sources of uncertainty
- Predictive uncertainty: Epistemic and Aleatoric uncertainty

IAL

Introduction to active learning (AL)

- With the explosion of IoT devices and Internet data, there is a huge amount of free unlabeled data.

IAL

Introduction to active learning (AL)

- With the explosion of IoT devices and Internet data, there is a huge amount of free unlabeled data.
- We need to annotate some of this data to prepare training data, but the annotation process is:

IAL

Introduction to active learning (AL)

- With the explosion of IoT devices and Internet data, there is a huge amount of free unlabeled data.
- We need to annotate some of this data to prepare training data, but the annotation process is:
  - **time-consuming** (e.g., annotating long documents or videos).

IAL

Introduction to active learning (AL)

- With the explosion of IoT devices and Internet data, there is a huge amount of free unlabeled data.
- We need to annotate some of this data to prepare training data, but the annotation process is:
  - **time-consuming** (e.g., annotating long documents or videos).
  - **expensive** (e.g., annotating a patient may need medical tests, MRI scan, experts).

IAL

Introduction to active learning (AL)

- With the explosion of IoT devices and Internet data, there is a huge amount of free unlabeled data.
- We need to annotate some of this data to prepare training data, but the annotation process is:
  - **time-consuming** (e.g., annotating long documents or videos).
  - **expensive** (e.g., annotating a patient may need medical tests, MRI scan, experts).
  - **difficult to collect** due to the limited number of instances of some classes in many applications.

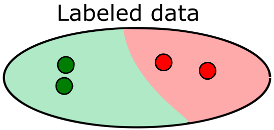
IAL

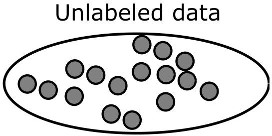
- This annotation problem could be solved by minimizing the size of the training data and selecting only the high-quality training data

IAL

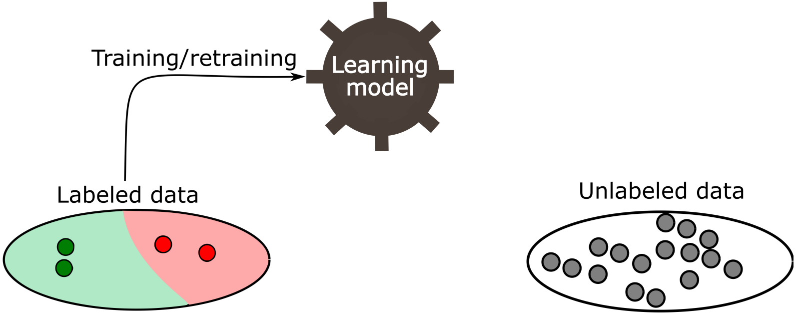
- This annotation problem could be solved by minimizing the size of the training data and selecting only the high-quality training data
- **Active Learning** actively queries or annotates the most representative and informative data points from a large pool of unlabeled data

Basic Workflow of Active Learning



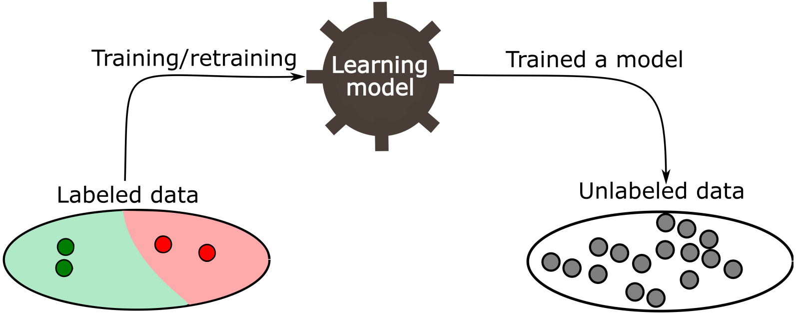


Basic Workflow of Active Learning

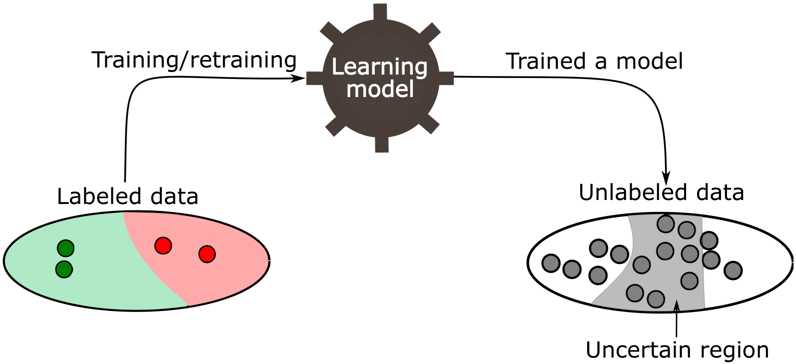


IAL

Basic Workflow of Active Learning

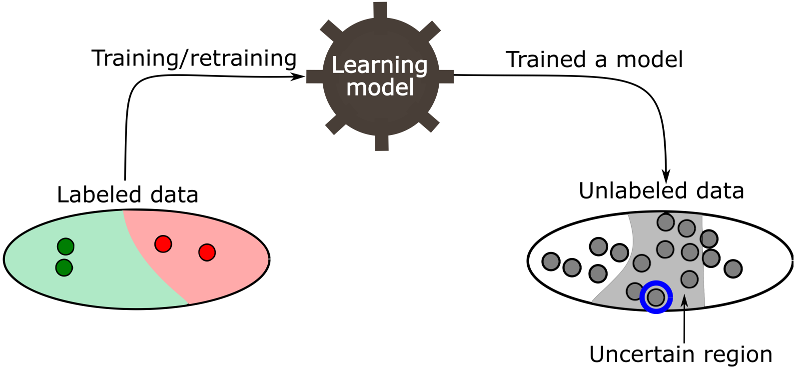


Basic Workflow of Active Learning



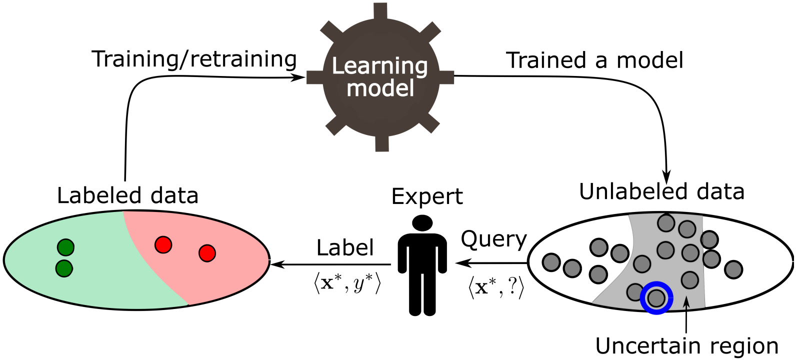
IAL

Basic Workflow of Active Learning



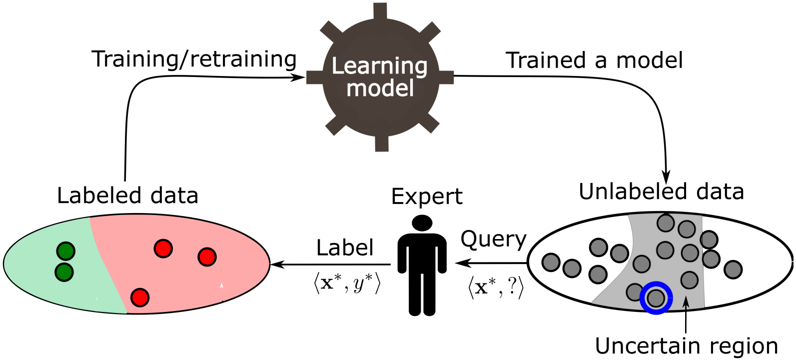
IAL

Basic Workflow of Active Learning



IAL

Basic Workflow of Active Learning



IAL

The main components of Active Learning

- **Data**: (i) unlabeled data (D_U), which represents the pool from which a new point is selected and (ii) labeled data (D_L) is used to train a model (h)

IAL

The main components of Active Learning

- **Data**: (i) unlabeled data (D_U), which represents the pool from which a new point is selected and (ii) labeled data (D_L) is used to train a model (h)
- **Learning algorithm** (h): The learning model (h) is trained on D_L.

IAL

The main components of Active Learning

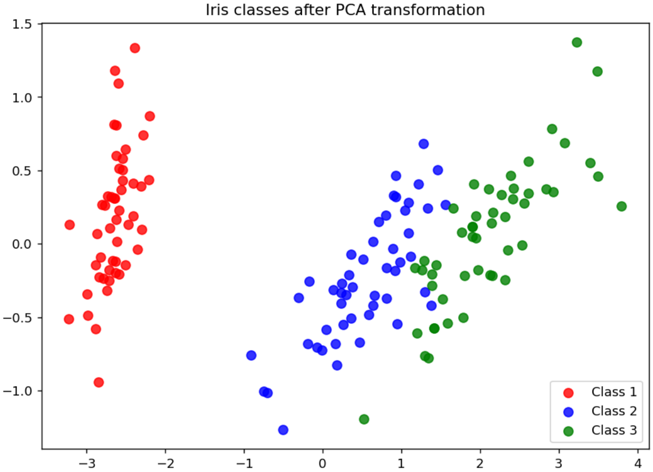
- **Data**: (i) unlabeled data (D_U), which represents the pool from which a new point is selected and (ii) labeled data (D_L) is used to train a model (h)
- **Learning algorithm** (h): The learning model (h) is trained on D_L.
- **Query strategy** (or **acquisition function**): This uses a specific utility function for evaluating the instances in D_U for selecting and querying point(s) in D_U

IAL

The main components of Active Learning

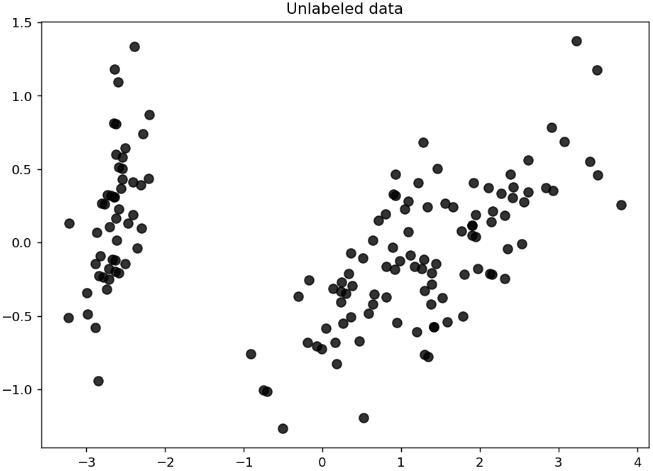
- **Data**: (i) unlabeled data (D_U), which represents the pool from which a new point is selected and (ii) labeled data (D_L) is used to train a model (h)
- **Learning algorithm** (h): The learning model (h) is trained on D_L.
- **Query strategy** (or **acquisition function**): This uses a specific utility function for evaluating the instances in D_U for selecting and querying point(s) in D_U
- **Annotator/labeler/oracle/Expert**: Who annotates/labels the queried unlabeled points

Simple AL example



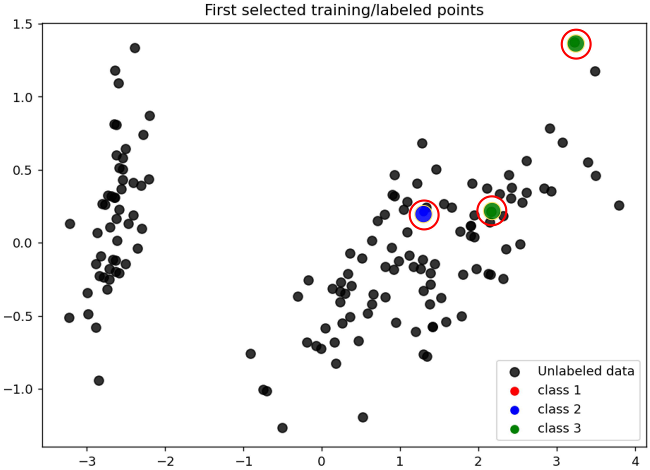
IAL

Simple AL example



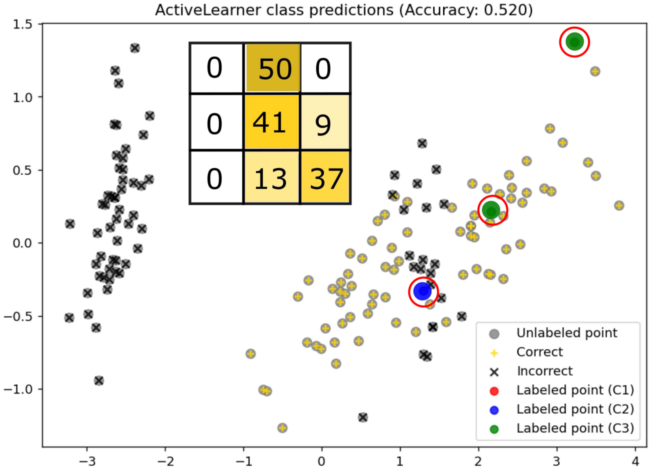
IAL

Simple AL example



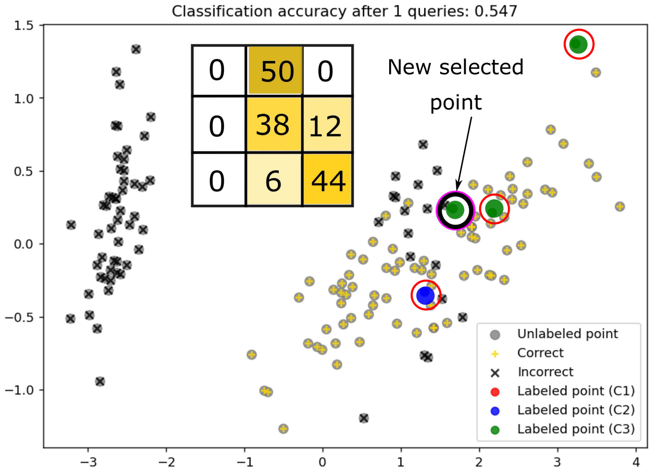
IAL

Simple AL example

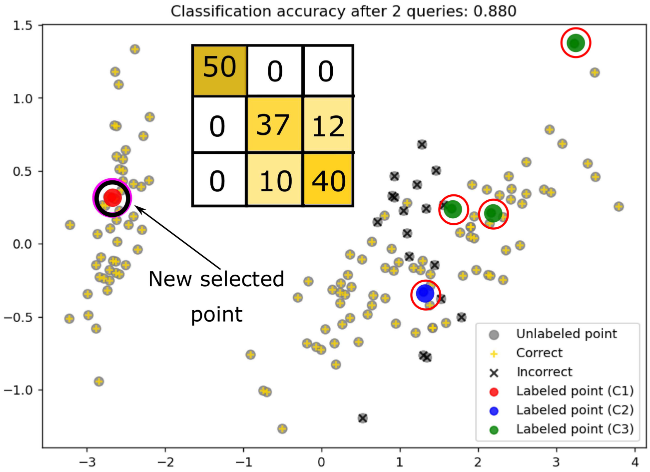


IAL

Simple AL example

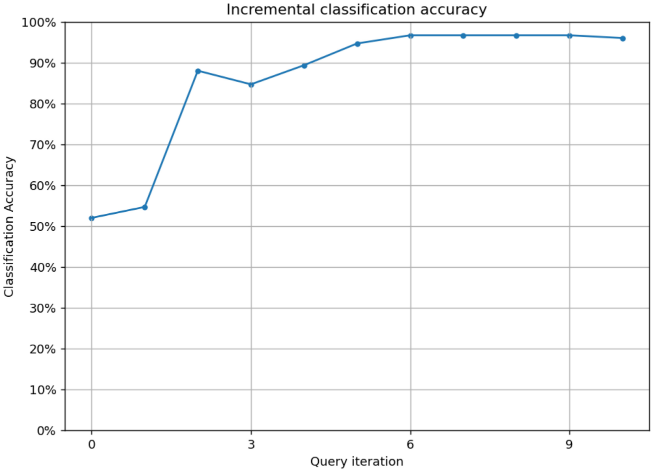


Simple AL example



IAL

Simple AL example



IAL

- AL uses two phases:

IAL

- AL uses two phases:
  - The "**Exploration phase**" focusing on diverse and representative points aligns well with the goal of improving the model's generalizability.

IAL

- AL uses two phases:
  - The "**Exploration phase**" focusing on diverse and representative points aligns well with the goal of improving the model's generalizability.
  - The "**Exploitation phase**" targeting highly informative data points around uncertain regions, such as decision boundaries, is a critical aspect of active learning.

IAL

- AL uses two phases:
  - The "**Exploration phase**" focusing on diverse and representative points aligns well with the goal of improving the model’s generalizability.
  - The "**Exploitation phase**" targeting highly informative data points around uncertain regions, such as decision boundaries, is a critical aspect of active learning.
- AL uses uncertainty quantification methods to decide which instances to query for labels, **how can we quantify this uncertainty?**

IAL

Agenda

- Introduction to active learning
- Sources of uncertainty
- Predictive uncertainty: Epistemic and Aleatoric uncertainty

IAL

General framework for supervised learning

- Given
  - a set of training data (fixed dataset): $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is an instance space and $\mathcal{Y}$ the set of outcomes.
  - a hypothesis space $\mathcal{H} = \{h_1, \ldots, h_m\}$ and a loss function l, and h is identified by a unique parameter vector $\theta \in \Theta$.

IAL

General framework for supervised learning

- Given
  - a set of training data (fixed dataset): $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is an instance space and $\mathcal{Y}$ the set of outcomes.
  - a hypothesis space $\mathcal{H} = \{h_1, \ldots, h_m\}$ and a loss function l, and h is identified by a unique parameter vector $\theta \in \Theta$.
- The goal of the learner is to search within $\mathcal{H}$ for a hypothesis $h^* \in \mathcal{H}$ with minimum risk.

IAL

General framework for supervised learning

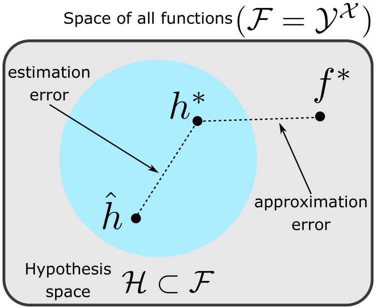
- Given
  - a set of training data (fixed dataset): $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is an instance space and $\mathcal{Y}$ the set of outcomes.
  - a hypothesis space $\mathcal{H} = \{h_1, \ldots, h_m\}$ and a loss function l, and h is identified by a unique parameter vector $\theta \in \Theta$.
- The goal of the learner is to search within $\mathcal{H}$ for a hypothesis $h^* \in \mathcal{H}$ with minimum risk.
- The choice of the hypothesis is guided by the empirical risk $R_{emp}(h_i \in \mathcal{H}) := \frac{1}{N} \sum_{i=1}^{N} l(h_i(\mathbf{x}, y))$.

General framework for supervised learning

- Given
  - a set of training data (fixed dataset): $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is an instance space and $\mathcal{Y}$ the set of outcomes.
  - a hypothesis space $\mathcal{H} = \{h_1, \ldots, h_m\}$ and a loss function l, and h is identified by a unique parameter vector $\theta \in \Theta$.
- The goal of the learner is to search within $\mathcal{H}$ for a hypothesis $h^* \in \mathcal{H}$ with minimum risk.
- The choice of the hypothesis is guided by the empirical risk $R_{emp}(h_i \in \mathcal{H}) := \frac{1}{N} \sum_{i=1}^{N} l(h_i(\mathbf{x}, y))$.
- $\hat{h} := \underset{h \in \mathcal{H}}{\arg\min}\, \hat{R}_{emp}(h)$ is an estimation of the true risk minimizer, $h^* := \underset{h \in \mathcal{H}}{\arg\min}\, R(h)$.

Types of errors:

- **Estimation error** (how good is the learned predictor relative to the hypothesis class)
- **Approximation error** (how good is the hypothesis class)



Sources of uncertainty

There are different sources of uncertainty:

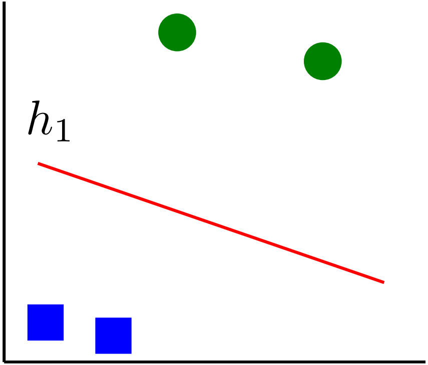
1. The dependency between $\mathcal{X}$ and $\mathcal{Y}$ is stochastic ($p(y|\mathbf{x}_q) = \frac{p(\mathbf{x}_q, y)}{p(\mathbf{x}_q)}$); so, uncertainty arises from imperfect data **"Aleatoric Uncertainty, Irreducible"**

IAL

- There is an uncertainty arising from the quality of how $\hat{h}$ estimates h^*.

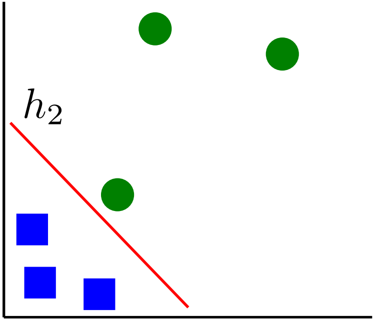
IAL

- There is an uncertainty arising from the quality of how $\hat{h}$ estimates h^*.
- For example, given a small training data, two classes.



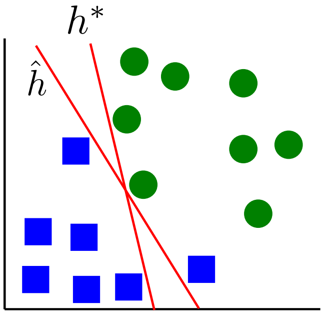
IAL

- There is an uncertainty arising from the quality of how $\hat{h}$ estimates h^*.
- For example, given a small training data, two classes.
- Adding more training points (getting more knowledge) reduces the gap between $\hat{h}$ and h^*.



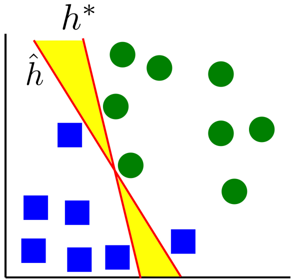
IAL

- There is an uncertainty arising from the quality of how $\hat{h}$ estimates h^*.
- For example, given a small training data, two classes.
- Adding more training points (getting more knowledge) reduces the gap between $\hat{h}$ and h^*.



IAL

- There is an uncertainty arising from the quality of how $\hat{h}$ estimates h^*.
- For example, given a small training data, two classes.
- Adding more training points (getting more knowledge) reduces the gap between $\hat{h}$ and h^*.



IAL

There are different sources of uncertainty:

1. The dependency between $\mathcal{X}$ and $\mathcal{Y}$ is stochastic ($p(y|\mathbf{x}_q) = \frac{p(\mathbf{x}_q,y)}{p(\mathbf{x}_q)}$); so, uncertainty arises from imperfect data **"Aleatoric Uncertainty, Irreducible"**
2. $\hat{h}$ is an estimation of h^*, and the quality of this estimation depends on the quality and the amount of training data, this is called "**approximation uncertainty**"

IAL

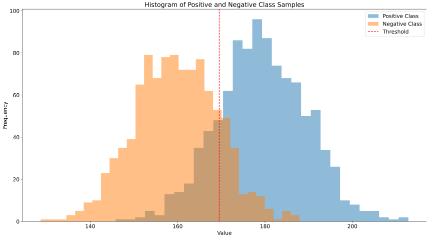
Uncertainty may arise based on the type of learning model

IAL

Uncertainty may arise based on the type of learning model

- Let we have two classes (positive and negative) distributed as follows: $N(180, 10)$ and $N(160, 10)$ for the positive and negative classes, respectively, and

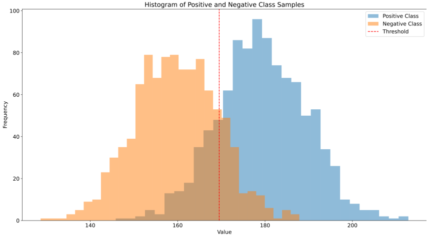
$$\mathcal{H} = \{h_t | t \in R\}, h_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x} < t \\ +1 & \text{if } \mathbf{x} \geq t \end{cases}$$



IAL

Uncertainty may arise based on the type of learning model

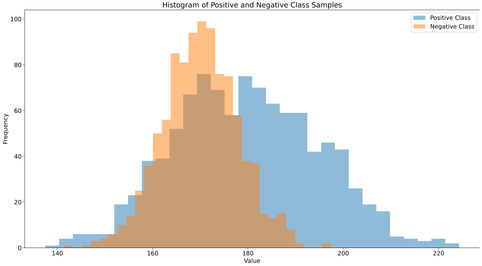
- Let we have two classes (positive and negative) distributed as follows: $N(180, 10)$ and $N(160, 10)$ for the positive and negative classes, respectively, and $\mathcal{H} = \{h_t | t \in R\}, h_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x} < t \\ +1 & \text{if } \mathbf{x} \geq t \end{cases}$



- With 0/1 loss, then the Bayes predictor (f^*) will be $f^*(\mathbf{x}) = h^*(\mathbf{x}) = h_{170}$

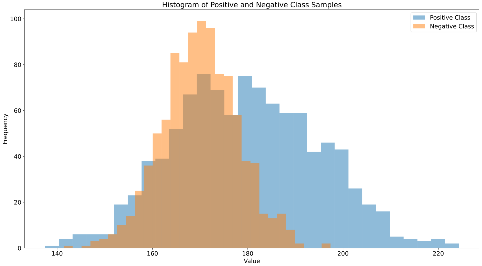
Uncertainty may arise based on the type of learning model

- If we change the distribution as follows: $N(180, 15)$ and $N(170, 8)$ for positive and negative classes, respectively, and $\mathcal{H} = \{h_t | t \in R\}$, $h_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x} < t \\ +1 & \text{if } \mathbf{x} \geq t \end{cases}$



Uncertainty may arise based on the type of learning model

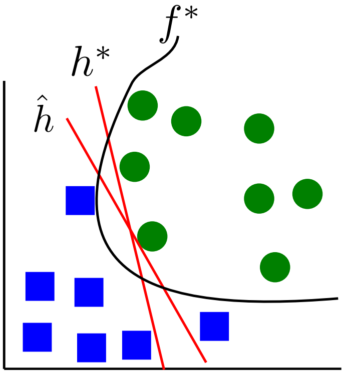
- If we change the distribution as follows: $N(180, 15)$ and $N(170, 8)$ for positive and negative classes, respectively, and $\mathcal{H} = \{h_t | t \in R\}$, $h_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x} < t \\ +1 & \text{if } \mathbf{x} \geq t \end{cases}$



- Then, $f^* \notin \mathcal{H}$, and $h^* \neq f^*$

IAL

- Another example using a linear learning model with non-linearly separable data.
- h^* will be a linear approximation of the non-linear ground truth function (f^*).



There are different sources of uncertainty:

- The dependency between $\mathcal{X}$ and $\mathcal{Y}$ is stochastic ($p(y|\mathbf{x}_q) = \frac{p(\mathbf{x}_q, y)}{p(\mathbf{x}_q)}$); so, uncertainty arises from imperfect data **"Aleatoric Uncertainty, Irreducible"**
- $\hat{h}$ is an estimation of h^*, and the quality of this estimation depends on the quality and the amount of training data, this is called "**approximation uncertainty**"
- The type of learning model (hypothesis class) has an impact on how h^* approximates f^* (ground truth), this is **"model uncertainty"**

IAL

There are different sources of uncertainty:

- The dependency between $\mathcal{X}$ and $\mathcal{Y}$ is stochastic ($p(y|\mathbf{x}_q) = \frac{p(\mathbf{x}_q, y)}{p(\mathbf{x}_q)}$); so, uncertainty arises from imperfect data **"Aleatoric Uncertainty, Irreducible"**
- $\hat{h}$ is an estimation of h^*, and the quality of this estimation depends on the quality and the amount of training data, this is called "**approximation uncertainty**"
- The type of learning model (hypothesis class) has an impact on how h^* approximates f^* (ground truth), this is **"model uncertainty"**
- Model uncertainty + Approximation uncertainty are **"Epistemic Uncertainty, Reducible"**, this uncertainty arises from imperfect knowledge

IAL

Agenda

- Introduction to active learning
- Sources of uncertainty
- Predictive uncertainty: Epistemic and Aleatoric uncertainty

IAL

Epistemic and Aleatoric uncertainty

- **Aleatoric uncertainty** (AU) refers to stochastic nature of dependence $p(y|\mathbf{x})$ between instances and outcomes.
- **Epistemic uncertainty** refers to uncertainty caused by a lack of knowledge (model or training data).

IAL

Epistemic and Aleatoric uncertainty

- **Aleatoric uncertainty** (AU) refers to stochastic nature of dependence $p(y|\mathbf{x})$ between instances and outcomes.
- **Epistemic uncertainty** refers to uncertainty caused by a lack of knowledge (model or training data).
- **Uncertainty quantification** (EU): How to quantify the learner's uncertainty? How to measure and disentangle the different types of uncertainty (aleatoric, epistemic, total)?

IAL

- Entropy is the measure of uncertainty of a random variable Y

$$H[Y] = H[p] = -\sum_{y \in \mathcal{Y}} p(y) log_2(p(y))$$

- Entropy is the measure of uncertainty of a random variable Y

$$H[Y] = H[p] = -\sum_{y \in \mathcal{Y}} p(y) log_2(p(y))$$

- Total uncertainty = entropy of the predictive posterior distribution, in the case of discrete $\mathcal{Y}$ given by

$$TU(\mathbf{x}) = H[p(y|\mathbf{x})] = -\sum_{y \in \mathcal{Y}} p(y|\mathbf{x})\ log_2\ p(y|\mathbf{x})$$

IAL

- Entropy is the measure of uncertainty of a random variable Y

$$H[Y] = H[p] = -\sum_{y \in \mathcal{Y}} p(y) log_2(p(y))$$

- Total uncertainty = entropy of the predictive posterior distribution, in the case of discrete $\mathcal{Y}$ given by

$$TU(\mathbf{x}) = H[p(y|\mathbf{x})] = -\sum_{y \in \mathcal{Y}} p(y|\mathbf{x})\ log_2\ p(y|\mathbf{x})$$

- TU can we decompose it into aleatoric and epistemic uncertainties

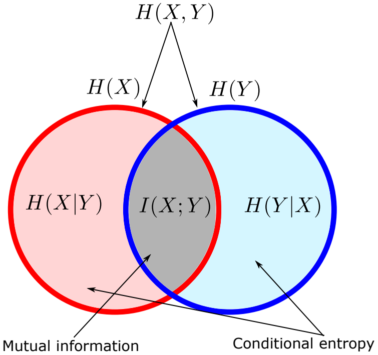
$$\underbrace{TU(\mathbf{x})}_{\text{Total uncertainty}} = \underbrace{AU(\mathbf{x})}_{\text{Aleatoric uncertainty}} + \underbrace{EU(\mathbf{x})}_{\text{Epistemic uncertainty}}$$

IAL

- The joint entropy ($H(X, Y)$) of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is defined as, $H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) log p(x, y)$

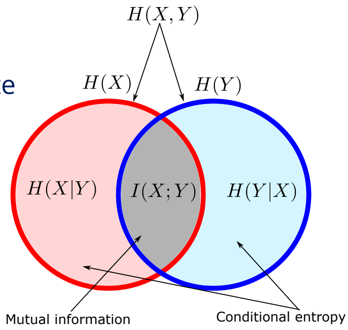
IAL

- The joint entropy ($H(X,Y)$) of a pair of discrete random variables (X,Y) with a joint distribution $p(x,y)$ is defined as, $H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) log p(x,y)$



- The joint entropy ($H(X,Y)$) of a pair of discrete r.v. (X,Y) with a joint dist. $p(x,y)$ is

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)log(p(x,y))$$

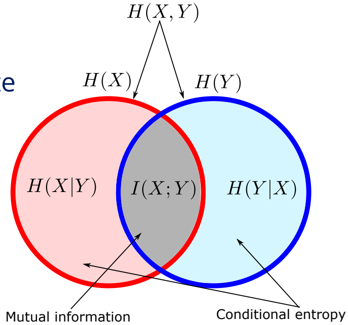


- The conditional entropy $H(Y|X)$ (uncertainty left about Y after knowing X) is defined as

$$H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x|y)log(p(x|y))$$

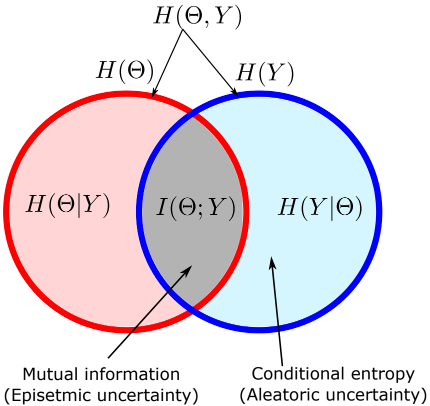
- The joint entropy ($H(X,Y)$) of a pair of discrete r.v. (X,Y) with a joint dist. $p(x,y)$ is

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) log(p(x,y))$$



The mutual information $I(X;Y)$: the reduction in the uncertainty of X due to the knowledge of Y (the shared information between X and Y)

IAL



$H(\Theta, Y)$
$H(\Theta)$
$H(Y)$
$H(\Theta|Y)$
$I(\Theta; Y)$
$H(Y|\Theta)$
Mutual information
(Episetmic uncertainty)
Conditional entropy
(Aleatoric uncertainty)

IAL

- In the case of NN, where hypotheses $h = h_\theta$ are identified by network weights θ, EU corresponds to uncertainty about these weights.

IAL

- In the case of NN, where hypotheses $h = h_\theta$ are identified by network weights θ, EU corresponds to uncertainty about these weights.
- EU captures the amount of information about the model parameters θ that would be gained through knowledge of the true outcome y.

IAL

- In the case of NN, where hypotheses $h = h_\theta$ are identified by network weights θ, EU corresponds to uncertainty about these weights.
- EU captures the amount of information about the model parameters θ that would be gained through knowledge of the true outcome y.
- EU can be quantified using the mutual information between the outcomes (Y) and the hypotheses (Θ), denoted as $I(Y;\Theta)$.

$$\underbrace{TU(\mathbf{x})}_{\substack{\text{Total uncertainty} \\ H[Y]}} = \underbrace{AU(\mathbf{x})}_{\text{Aleatoric uncertainty}} + \underbrace{EU(\mathbf{x})}_{\substack{\text{Epistemic uncertainty} \\ I(Y;\Theta)}}$$

IAL

- Fixing the parameters ignores the EU; then NN gives a probability dist. for each $\mathbf{x}$ as $p(y|\mathbf{x}, \theta)$), the expectation over the entropies of these dists.,

IAL

- Fixing the parameters ignores the EU; then NN gives a probability dist. for each **x** as $p(y|\mathbf{x},\theta)$), the expectation over the entropies of these dists.,

$$E_{p(\theta|D)}H[p(y|\mathbf{x},\theta)] = -\int \underbrace{p(\theta|D)}_{weight} \left(\underbrace{\sum_{y\in\mathcal{Y}} p(y|\mathbf{x},\theta) log_2(p(y|\mathbf{x},\theta))}_{\text{conditional entropy}} \right) d\theta$$

IAL

- Fixing the parameters ignores the EU; then NN gives a probability dist. for each **x** as $p(y|\mathbf{x},\theta)$), the expectation over the entropies of these dists.,

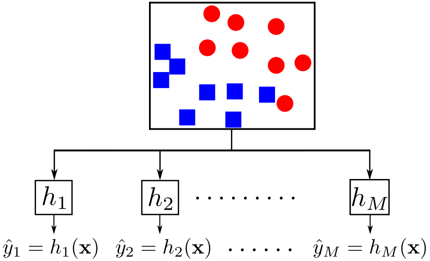
$$E_{p(\theta|D)}H[p(y|\mathbf{x},\theta)] = -\int \underbrace{p(\theta|D)}_{weight} \left(\underbrace{\sum_{y\in\mathcal{Y}} p(y|\mathbf{x},\theta) log_2(p(y|\mathbf{x},\theta))}_{\text{conditional entropy}} \right) d\theta$$

- The aleatoric uncertainty is measured in terms of conditional entropy.

$$\underbrace{TU(\mathbf{x})}_{\substack{\text{Total uncertainty}\\ H[Y]}} = \underbrace{AU(\mathbf{x})}_{\substack{\text{Aleatoric uncertainty}\\ H[Y|\Theta]}} + \underbrace{EU(\mathbf{x})}_{\substack{\text{Epistemic uncertainty}\\ I(Y;\Theta)}}$$

Ensemble methods for uncertainty quantification

- The ensemble learners may produce different class predictions for a data point (**x**), indicating the presence of uncertainty in the prediction.
- The more ensemble members will disagree on their probabilistic predictions, the higher the epistemic uncertainty.
- Each hypothesis h_i produces prediction p_1, so we will get $p_1, \dots, p_M$.



IAL

- An approximation (turning integrals to sums) of conditional entropy can be obtained by

$$AU(\mathbf{x}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{y\in\mathcal{Y}} p_i(y|\mathbf{x})log_2(p_i(y|\mathbf{x}))$$

IAL

- An approximation (turning integrals to sums) of conditional entropy can be obtained by

$$AU(\mathbf{x}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{y\in\mathcal{Y}} p_i(y|\mathbf{x}) log_2(p_i(y|\mathbf{x}))$$

- An approximation of total uncertainty (Shannon entropy) by

$$TU(\mathbf{x}) = -\sum_{y\in\mathcal{Y}} \underbrace{\left(\frac{1}{M}\sum_{i=1}^{M} p_i(y|\mathbf{x})\right)}_{\bar{p}(y|\mathbf{x})} log_2 \underbrace{\left(\frac{1}{M}\sum_{i=1}^{M} p_i(y|\mathbf{x})\right)}_{\bar{p}(y|\mathbf{x})}$$

- An approximation (turning integrals to sums) of conditional entropy can be obtained by

$$AU(\mathbf{x}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{y\in\mathcal{Y}} p_i(y|\mathbf{x})log_2(p_i(y|\mathbf{x}))$$

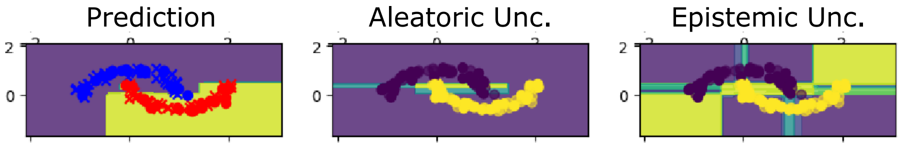
- An approximation of total uncertainty (Shannon entropy) by

$$TU(\mathbf{x}) = -\sum_{y\in\mathcal{Y}} \underbrace{\left(\frac{1}{M}\sum_{i=1}^{M} p_i(y|\mathbf{x})\right)}_{\bar{p}(y|\mathbf{x})} log_2 \underbrace{\left(\frac{1}{M}\sum_{i=1}^{M} p_i(y|\mathbf{x})\right)}_{\bar{p}(y|\mathbf{x})}$$

- EU is the difference between TU and AU, $EU = TU - AU$

IAL

- The code of an example is here:



IAL

Hands-on Pool-based Active Learning via `scikit-activeml`

Marek Herde, Minh Tuan Pham, and Bernhard Sick

`scikit-activeml`: **Motivation**

- Active learning libraries are often specialized for a specific domain
- Comparing active learning strategies implemented across libraries can be challenging
- Adapting existing libraries to special cases may not be feasible without considerable amounts of work

IAL

`scikit-activeml`: **Motivation**

scikit activeml

What is `scikit-activeml`?

- Active learning library initiated in 2020
- Published via the 3-Clause BSD license
- Build on top of `scikit-learn`

GitHub

Documentation

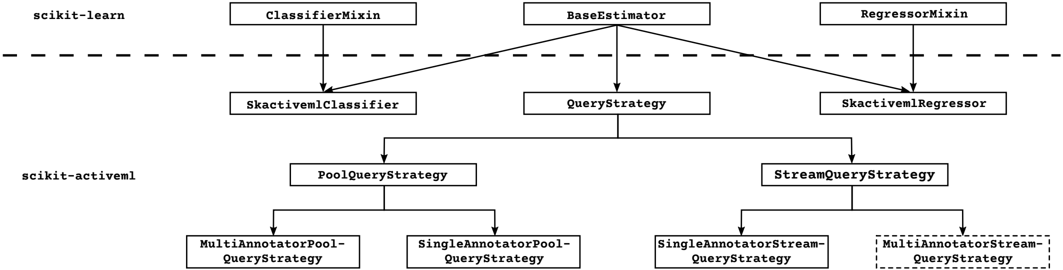
Github: `https://github.com/scikit-activeml/scikit-activeml`
Documentation: `https://scikit-activeml.github.io/scikit-activeml-docs/`

`scikit-activeml`: **Motivation**

The goal of `scikit-activeml` is to

- bridge active learning research and application
- provide a modular and unified framework for ease of use
- accommodate a wide range of domains and applications with high flexibility

scikit-activeml: Structure and Design



scikit-activeml: Minimal Example

Initialization

```python
# Generate data set.
X, y_true = make_blobs(n_samples=200, centers=4, random_state=0)
y = np.full(shape=y_true.shape, fill_value=MISSING_LABEL)
```

scikit-activeml: Minimal Example

Initialization

```
# Generate data set.
X, y_true = make_blobs(n_samples=200, centers=4, random_state=0)
y = np.full(shape=y_true.shape, fill_value=MISSING_LABEL)

# Use the first 10 samples as initial training data.
y[:10] = y_true[:10]
```

`scikit-activeml`: Minimal Example

Initialization

```
# Generate data set.
X, y_true = make_blobs(n_samples=200, centers=4, random_state=0)
y = np.full(shape=y_true.shape, fill_value=MISSING_LABEL)

# Use the first 10 samples as initial training data.
y[:10] = y_true[:10]

# Create classifier and query strategy.
clf = SklearnClassifier(
    GaussianProcessClassifier(random_state=0),
    classes=np.unique(y_true),
    random_state=0
)
```

scikit-activeml: Minimal Example

Initialization

```
# Generate data set.
X, y_true = make_blobs(n_samples=200, centers=4, random_state=0)
y = np.full(shape=y_true.shape, fill_value=MISSING_LABEL)

# Use the first 10 samples as initial training data.
y[:10] = y_true[:10]

# Create classifier and query strategy.
clf = SklearnClassifier(
    GaussianProcessClassifier(random_state=0),
    classes=np.unique(y_true),
    random_state=0
)
qs = UncertaintySampling(method='entropy', random_state=0)
```

`scikit-activeml`: **Minimal Example**

Active Learning Cycle

```
# Execute active learning cycle.
n_cycles = 30
for c in range(n_cycles):
    query_idx = qs.query(X=X, y=y, clf=clf)
    y[query_idx] = y_true[query_idx]

# Fit final classifier.
clf.fit(X, y)
```

IAL

scikit-activeml: Content Overview

		AL			Supervised Tasks	
	Query Strategies [#]	Pool	Stream	Multiple Annotators	Regression	Classification
libact	19	✓	✗	✓	✓	✓
modal	21	✓	✓	✗	✓	✓
ALiPy	25	✓	✗	✓	✗	✓
Baal	10	✓	✗	✗	✗	✓
lrtc	7	✓	✗	✗	✗	✓
ALToolbox	19	✓	✗	✗	✗	✓
PyRelationsAL	14	✓	✗	✗	✓	✓
scikit-activeml	52	✓	✓	✓	✓	✓

IAL

scikit-activeml: Documentation

What is the goal of scikit-activeml?

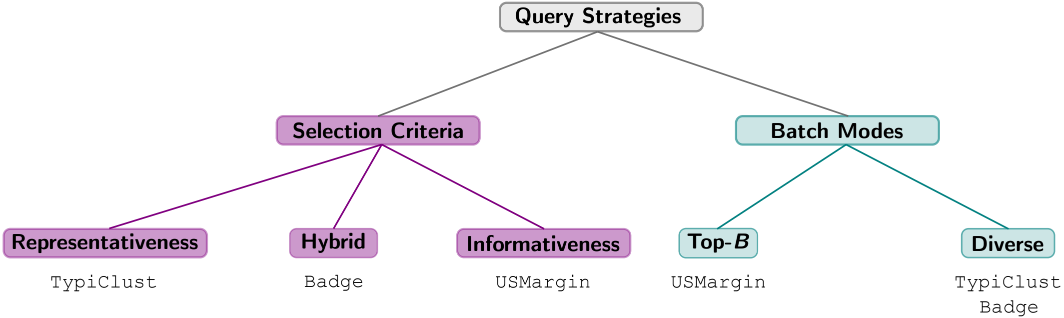
- Strategy Overview to highlight publications
- Examples for all single-annotator query strategies
- Tutorials for various use-cases with extensive descriptions:
  - Getting started tutorials for many scenarios
  - Stream-based active learning using river
  - Deep active learning using skorch
  - Deep active learning using self-supervised learning embeddings
  - Image annotation via active learning

Documentation

Documentation: https://scikit-activeml.github.io/scikit-activeml-docs/

IAL

Pool-based Active Learning: Concepts



IAL

Pool-based Active Learning: TypiClust[1]

Representativeness Query Strategy

- Cluster all (labeled $\mathcal{L}$ and unlabeled $\mathcal{U}$) samples via K-means, where the cluster number equals the sum of the number of labeled samples and batch size:

$$K = |\mathcal{L}| + B$$

- There are at least B clusters, denoted as $\mathcal{C}_1, \ldots, \mathcal{C}_B \subset \mathcal{U}$, with no labeled sample
- Select from each of these clusters the sample with the highest typicality (density):

$$\mathbf{x}_b^* = \underset{\mathbf{x} \in \mathcal{C}_b}{\arg\max} \left(\frac{1}{|\mathcal{M}_\mathbf{x}|} \sum_{\mathbf{x}' \in \mathcal{M}_\mathbf{x}} ||\mathbf{x} - \mathbf{x}'||_2 \right)^{-1},$$

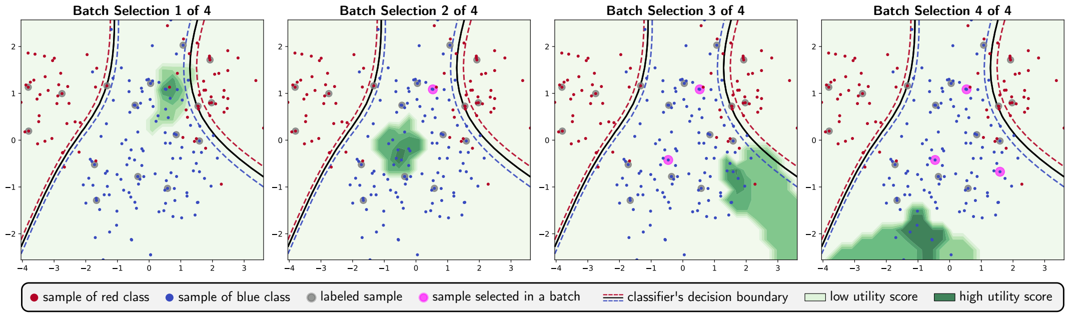
where $\mathcal{M}_\mathbf{x}$ denotes the nearest neighbors of sample $\mathbf{x}$

[1] Hacohen et al., “Active Learning on a Budget: Opposite Strategies Suit High and Low Budgets”, 2022.

IAL

Pool-based Active Learning: `TypiClust`

Visualization



IAL

Pool-based Active Learning: `USMargin`[2]

Informativeness Query Strategy

- Compute the uncertainty score as the difference between the probabilities of the most $y^{(1)}$ and second most $y^{(2)}$ probable class label prediction for all unlabeled samples $\mathbf{x} \in \mathcal{U}$:

$$\Pr\left(y^{(1)} \mid \mathbf{x}, \boldsymbol{\theta}\right) - \Pr\left(y^{(2)} \mid \mathbf{x}, \boldsymbol{\theta}\right),$$

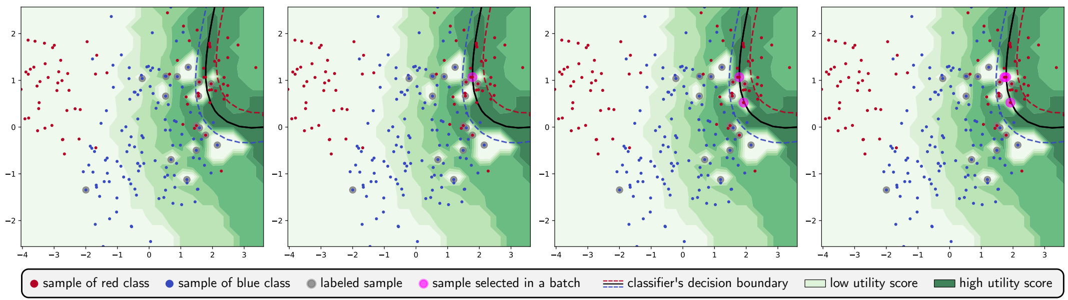
  where $\boldsymbol{\theta}$ denote the classifier's current parameters
- Select the top-B samples with the highest uncertainty scores

[2] Scheffer et al., "Active Hidden Markov Models for Information Extraction", 2001.

IAL

Pool-based Active Learning: `USMargin`

Visualization



Pool-based Active Learning: Badge[3]

Hybrid Query Strategy

- For all unlabeled samples $\mathbf{x} \in \mathcal{U}$:
  - Obtain the class label prediction of the classifier with the current parameters $\boldsymbol{\theta}$:

$$\hat{y}_{\mathbf{x}} = \arg\max_{y} \left(\Pr(y \mid \mathbf{x}, \boldsymbol{\theta}) \right) .$$

  - Compute the gradient embedding given the pseudo label $\hat{y}_{\mathbf{x}}$ regarding the parameters $\boldsymbol{\theta}_{\text{out}}$ of the final output layer:

$$\hat{\mathbf{g}}_{\mathbf{x}} = \frac{\partial}{\partial \boldsymbol{\theta}_{\text{out}}} L_{\text{CE}} \left(\Pr(y \mid \mathbf{x}, \boldsymbol{\theta}), \hat{y}_{\mathbf{x}} \right)$$

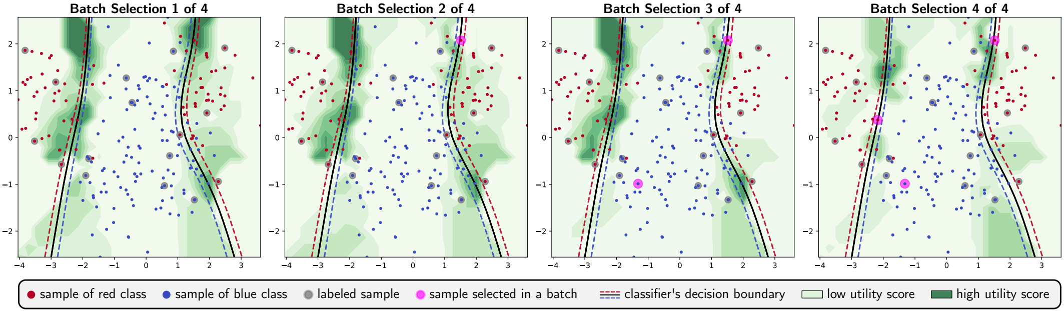
- Select B samples via the K-means++ algorithm on $\{\hat{\mathbf{g}}_{\mathbf{x}} \mid \mathbf{x} \in \mathcal{U}\}$

[3] Ash et al., “Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds”, 2020.

IAL

Pool-based Active Learning: Badge

Visualization



IAL

Pool-based Active Learning: Benchmark

Objectives

We aim to build a **benchmark in active learning** that

- evaluates a wide *variety of combinations* of query strategies, models, datasets, scenarios, etc.,
- is *continuously updated* with newly developed strategies and models,
- and *interactively visualizes* reproducible results and provides them as *downloadable resources* for the investigation of user-specific hypotheses.

IAL

Pool-based Active Learning: Benchmark

Overview of Benchmarks

Benchmarks	Zhan et al.	Munjal et al.	Holzmüller et al.	Rauch et al.	Lüth et al.	Huseljic et al.	Werner et al.	Margraf et al.	Ours
Publication									
Venue	IJCAI	CVPR	JMLR	ECML-PKDD	NeurIPS	TMLR	arXiv	arXiv	–
Year	2021	2023	2023	2023	2024	2024	2024	2024	2024
Code	✓	✓	✓	✓	✓	✓	✓	✓	✓
Web App	✗	✗	✗	✗	✗	✗	✗	✗	✓
Result Export	✗	✗	✗	✗	✗	✗	(✓)	✗	✓
Setup									
Classification	✓	✓	✗	✓	✓	✓	✓	✓	✓
Regression	✗	✗	✓	✗	✗	✗	✗	✗	✗
Datasets [#]	27	3	15	11	4	4	9	86	12
Image Data	✗	✓	✗	✗	✓	✓	✓	✗	✓
Text Data	✗	✗	✗	✓	✗	✗	✓	✗	✓
Tabular Data	✓	✗	✓	✗	✗	✗	✓	✓	✓
Strategies [#]	18	7	8	5	5	6	11	9	13

IAL

Pool-based Active Learning: Benchmark

Setup

Datasets

- Tabular Data
- Image Data
- Text Data

×

Embedding Models

- None
- DINOv2
- BERT

×

Classifiers

- ParzenWindowClassifier
- LogisticRegression
- RandomForestClassifier

×

Query Strategies

- TypiClust
- Badge
- ...

IAL

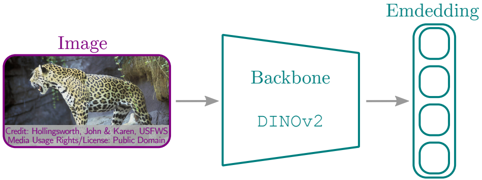
Pool-based Active Learning: Benchmark

Overview of Datasets

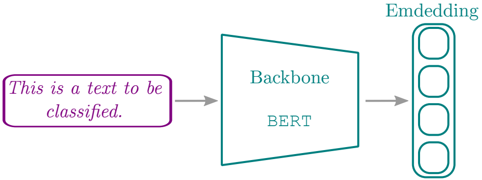
Datasets	Samples	Dimension	Classes	General Task
		Text Data		
`ag-news`	127,600	100 – 1,000 words	4	news topic classification
`bank-77`	13,083	13 – 433 words	77	banking task classification
`db-pedia`	630,000	3 – 13,600 words	14	company classification
`trec-6`	5,952	$\sim$ 10 words	6	question classification
		Image Data		
`cat-vs-dog`	23,410	$3\times4\times4$ – $3\times500\times500$ px^3	2	binary classification
`cifar-10`	60,000	32×32 px^2	10	generic object classification
`cifar-100`	60,000	32×32 px^2	100	generic object classification
`dtd`	5,640	300×300 – 640×640 px^3	47	textures classification
		Tabular Data		
`aloi`	108,000	128 features	1,000	outlier detection
`iris`	150	4 features	3	pattern classification
`letter`	20,000	16 features	26	letter classification
`pen-digits`	10,992	16 features	10	number classification

Pool-based Active Learning: Benchmark

Embedding Models



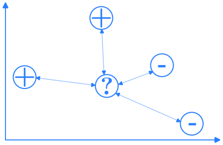
Oquab et al., “DINOv2: Learning Robust Visual Features without Supervision”, 2023



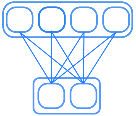
Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, 2018

Pool-based Active Learning: Benchmark

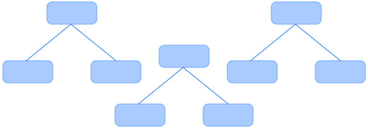
Classifiers



`ParzenWindowClassifier`



`LogisticRegression`



`RandomForestClassifier`

IAL

Pool-based Active Learning: Benchmark

Overview of Query Strategies

Query Strategies	Authors	Venue	Year	Batch Mode
Representativeness				
`RandomSampling`	–	–	–	diverse
`CoreSet`	Sener and Savarese	ICLR	2018	diverse
`GreedySamplingX`	Wu et al.	Inf. Sci.	2019	diverse
`ProbCover`	Yehuda et al.	NeurIPS	2022	diverse
`TypiClust`	Hacohen et al.	ICML	2022	diverse
Informativeness				
`USEntropy`	Settles and Craven	EMNLP	2008	top-B
`USLeastConfident`	Lewis and Gale	SIGIR	1994	top-B
`USMargin`	Scheffer et al.	IDA	2001	top-B
`Alce`	Huang and Lin	ICDM	2016	top-B
`ContrastiveAL`	Margatina et al.	EMNLP	2021	top-B
Hybrid				
`ProbabilisticAL`	Kottke et al.	ECAI	2016	top-B
`Badge`	Ash et al.	ICLR	2020	diverse
`Clue`	Prabhu et al.	ICCV	2021	diverse

IAL

Pool-based Active Learning: Benchmark

Live Demonstration

How is our benchmark presented?

- Operated as a Shiny web app[a]
- Early version for studying the feasibility
- Implemented by Alexander Benz
- Web app is accompanied by a GitHub repository[b]

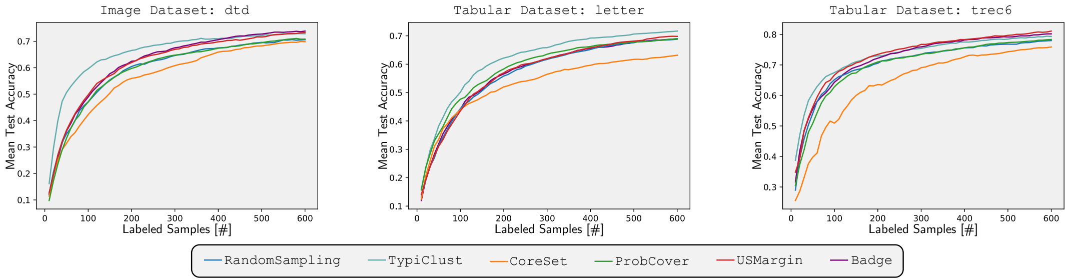
[a] `https://alexanderbenz.github.io/scikit-activeml-experiments`
[b] `https://github.com/AlexanderBenz/scikit-activeml-experiments`

Shiny Web App

GitHub

Pool-based Active Learning: Benchmark

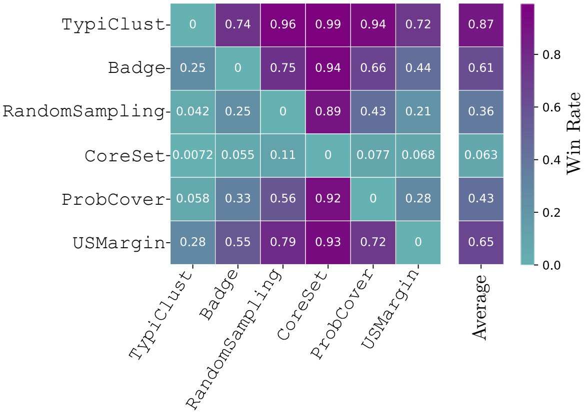
Exemplary Evaluation: Learning Curves



IAL

Pool-based Active Learning: Benchmark

Exemplary Evaluation: Pairwise Comparisons



Win rate matrix $\mathbf{W} \in [0, 1]^{6\times 6}$:

- Each row and column represents a query strategy
- $W[i, j]$ is the fraction of times strategy i had a higher accuracy than strategy j
- Diagonal entries $W[i, i]$ are not meaningful, as a strategy cannot be compared with itself

IAL

Pool-based Active Learning: Benchmark

Limitations and Outlook

Limitations:

- There are no results for
  - different initialization strategies,
  - fine-tuning backbone models,
  - additional learning tasks,
  - varying hyperparameters
- Interactive presentation of results is limited to learning curves
- No option to define the budget to explicitly study low- and high-budget scenarios

Outlook:

- The benchmark will be constantly grown
- The user interface will be expanded with new interaction elements
- An extension of `OpenML`[4] for active learning with `scikit-activeml` is being developed

[4] Feurer et al., "`OpenML-Python`: an extensible Python API for `OpenML`", 2021

Hands-on: Active Text Annotation via `scikit-activeml`

Jupyter notebook as tutorial[5]:

- The overall goal is to classify `arXiv` papers
- Classes:
  - — ML papers (cs.AI, cs.LG, cs.CV)
  - — non-ML papers (cs.LO, cs.NI, cs.CR, cs.PL)
- The task is to try out different query strategies and classifier combinations for annotating data

Notebook

[5] `https://scikit-activeml.github.io/scikit-activeml-docs/latest/generated/tutorials/06_pool_al_text_annotation_tool.html`

IAL

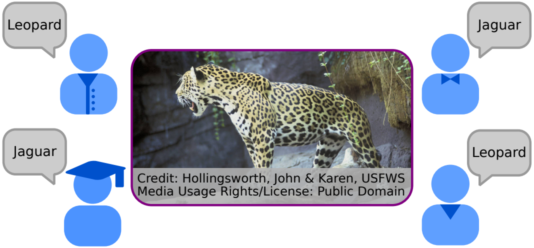
Towards Pool-based Active Learning with Error-prone Annotators

Marek Herde and Bernhard Sick

IAL

Multi-annotator Learning: Motivation

Who knows best?



- Data annotation campaigns often involve **multiple annotators**, e.g., crowdworkers
- Annotators are **error-prone** for various reasons[6]:
  - limited domain expertise
  - lack of attention
  - task complexity

[6] Herde, Huseljic, Sick, and Calma, "A Survey on Cost Types, Interaction Schemes, and Annotator Performance Models in Selection Algorithms for Active Learning in Classification", 2021

IAL

Multi-annotator Learning: Problem Setting

Objective

Given

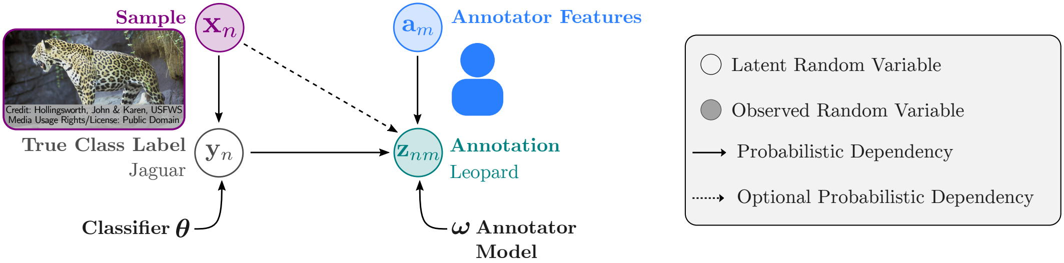
- $\Omega_Y = \{\mathbf{e}_1, \ldots, \mathbf{e}_C\}$ as the set of one-hot encoded class labels,
- $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^{N} \subseteq \Omega_X$ as the set of samples,
- $\mathcal{A} = \{\mathbf{a}_m\}_{m=1}^{M} \subseteq \Omega_A$ as the set of annotators,
- and $\mathcal{Z} = \{\mathbf{z}_{nm}\}_{n=1,m=1}^{N,M} \subseteq \Omega_Y \cup \{\mathbf{0}\}$ as the set of one-hot encoded noisy class labels with $\mathbf{0}$ denoting a missing annotation,

the **objective** is to train a classification model $\mathbf{y}_{\boldsymbol{\theta}^*} : \Omega_X \rightarrow \Omega_Y$ with parameters $\boldsymbol{\theta}^* \in \Theta$, which maximizes the accuracy:

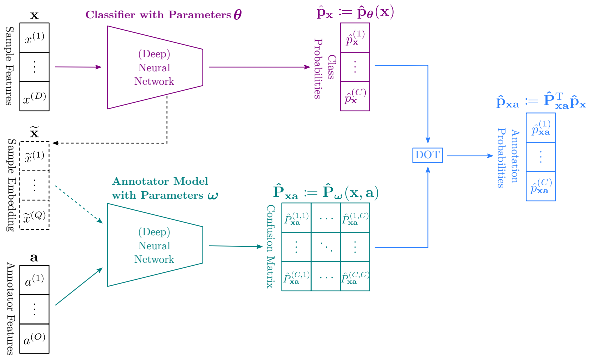
$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \Theta}{\arg\max} \left(E_{\mathbf{x},\mathbf{y}} \left[\mathbf{y}^{\mathrm{T}} \mathbf{y}_{\boldsymbol{\theta}}(\mathbf{x}) \right] \right).$$

Multi-annotator Learning: Problem Setting

Probabilistic Graphical Model



Multi-annotator Learning: MaDL[7]
Architecture



[7] Herde, Huseljic, and Sick, "Multi-annotator Deep Learning: A Probabilistic Framework for Classification", 2023.

IAL

Multi-annotator Learning: MaDL

Loss Function

Maximizing the **marginal likelihood** of the observed annotations

$$\Pr(\mathcal{Z} \mid \mathcal{X}, \mathcal{A}; \boldsymbol{\theta}, \boldsymbol{\omega}) = \prod_{n=1}^{N} \prod_{m \in \mathcal{A}_n} \mathbb{E}_{\mathbf{y}_n \mid \mathbf{x}_n; \boldsymbol{\theta}} \left[\Pr(\mathbf{z}_{nm} \mid \mathbf{x}_n, \mathbf{a}_m, \mathbf{y}_n; \boldsymbol{\omega}) \right]$$

$$= \prod_{n=1}^{N} \prod_{m \in \mathcal{A}_n} \mathbf{z}_{nm}^{\mathrm{T}} \underbrace{\hat{\mathbf{P}}_{\boldsymbol{\omega}}^{\mathrm{T}}(\mathbf{x}_n, \mathbf{a}_m) \hat{\mathbf{p}}_{\boldsymbol{\theta}}(\mathbf{x}_n)}_{\text{annotation probabilities}}$$

corresponds to minimizing the **cross-entropy** loss

$$L_{\mathcal{X}, \mathcal{A}, \mathcal{Z}}(\boldsymbol{\theta}, \boldsymbol{\omega}) = -\sum_{n=1}^{N} \sum_{m \in \mathcal{A}_n} \mathbf{z}_{nm}^{\mathrm{T}} \ln \left(\hat{\mathbf{P}}_{\boldsymbol{\omega}}^{\mathrm{T}}(\mathbf{x}_n, \mathbf{a}_m) \hat{\mathbf{p}}_{\boldsymbol{\theta}}(\mathbf{x}_n) \right)$$

IAL

Multi-annotator Learning: MaDL

Initialization

- The cross-entropy loss function alone **cannot separate** class probabilities and annotation noise[8] because there are infinitely many combinations:

$$\underbrace{\mathbf{P}^{\mathrm{T}}(\mathbf{x}_n, \mathbf{a}_m)\mathbf{p}(\mathbf{x}_n)}_{\text{true probabilities}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \underbrace{\hat{\mathbf{P}}_{\boldsymbol{\omega}}^{\mathrm{T}}(\mathbf{x}_n, \mathbf{a}_m)\hat{\mathbf{p}}_{\boldsymbol{\theta}}(\mathbf{x}_n)}_{\text{predicted probabilities}}$$

- One solution is to **initialize** the annotator model's parameters $\boldsymbol{\omega}$ such that

$$\hat{\mathbf{P}}_{\boldsymbol{\omega}}(\mathbf{x}_n, \mathbf{a}_m) \approx \eta \mathbf{I}_C + \frac{(1-\eta)}{C-1}\left(\mathbf{1}_C - \mathbf{I}_C\right),$$

where $\mathbf{I}_C \in \mathbb{R}^{C \times C}$ denotes an identity matrix, $\mathbf{1}_C \in \mathbb{R}^{C \times C}$ an all-one matrix, and $\eta \in (0, 1)$ the prior probability of obtaining a correct annotation

[8] Tanno et al., "Learning from Noisy Labels by Regularized Estimation of Annotator Confusion", 2019.

IAL

Multi-annotator Learning: Evaluation

Setup

- We exemplarily evaluate the test accuracy of multi-annotator learning approaches on two dataset
- Both datasets are annotated by real human crowdworkers
- Each crowdworker labeled only a subset such that the number of labels per sample varies

Datasets	`music`[9]	`label-me`[10]
data type	sound	image
samples [#]	1,000	1,188
classes [#]	10	8
labeling tool	Amazon Mechanical Turk	
annotators [#]	42	59
labels per sample [#]	4.2	2.5
labels per annotator [#]	67	43
fraction of false labels [%]	44.0	26.0
architecture	MLP	VGG-16

[9] Rodrigues, Pereira, and Ribeiro, "Learning from multiple annotators: Distinguishing good from random labelers", 2013

[10] Rodrigues and Pereira, "Deep Learning from Crowds", 2018

IAL

Multi-annotator Learning: Evaluation

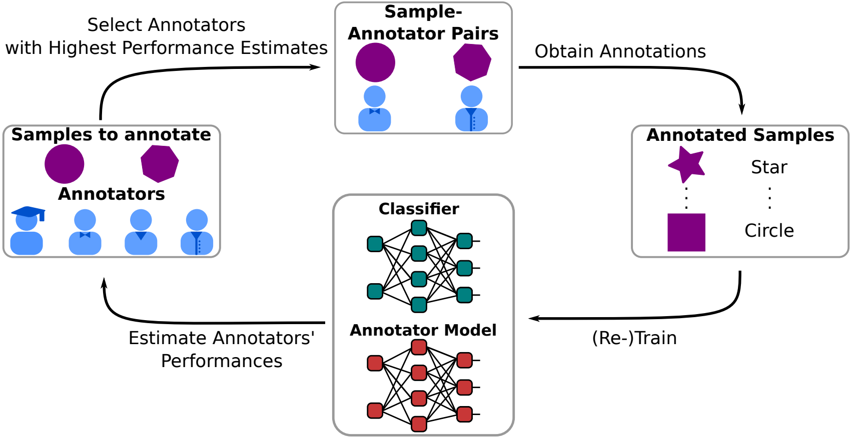
Results

Approaches	Authors	Venue	Year	`music`	`label-me`
Baselines					
`TrueLabels`	–	–	–	78.5±2.0	91.4±0.3
`MajorityVote`	–	–	–	64.6±4.5	81.0±1.5
Class-dependent Annotator Performance Modeling					
`CrowdLayer`	Rodrigues and Pereira	AAAI	2018	67.5±1.5	85.7±1.1
`TraceReg`	Tanno et al.	CVPR	2019	70.5±2.3	84.3±0.6
`UnionNet`	Wei et al.	TNNLS	2022	68.2±1.3	85.5±0.4
`CoNAL`	Chu et al.	AAAI	2021	70.8±3.1	86.6±0.4
Sample-dependent Annotator Performance Modeling					
`LIA`	Platanios et al.	arXiv	2020	65.8±2.3	81.3±1.0
`MaDL`	Herde, Huseljic, and Sick	TMLR	2023	74.3±1.8	86.7±0.4

Takeaway: Multi-annotator learning approaches estimating annotators' performances improve the test accuracy upon training with majority vote labels.

Multi-annotator Active Learning: Motivation

Active Learning Cycle



IAL

Multi-annotator Active Learning: Approach

Annotator Performance as Selection Criterion

- The sample's class probabilities estimated by the classification model, combined with the confusion matrix from the annotator, allow us to compute the **annotation correctness probability**:

$$\hat{p}_{\boldsymbol{\theta},\boldsymbol{\omega}}(\mathbf{x}, \mathbf{a}) = \sum_{c=1}^{C} \hat{p}_{\boldsymbol{\theta}}^{(c)}(\mathbf{x}) \cdot \hat{P}_{\boldsymbol{\omega}}^{(c,c)}(\mathbf{x}, \mathbf{a})$$

- By estimating the annotation correctness probabilities of each annotator per sample, we can select the one with the **highest performance**:

$$\mathbf{a}^* = \underset{\mathbf{a} \in \mathcal{A}}{\arg\max} \left(\hat{p}_{\boldsymbol{\theta},\boldsymbol{\omega}}(\mathbf{x}, \mathbf{a}) \right)$$

IAL

Multi-annotator Active Learning: Evaluation

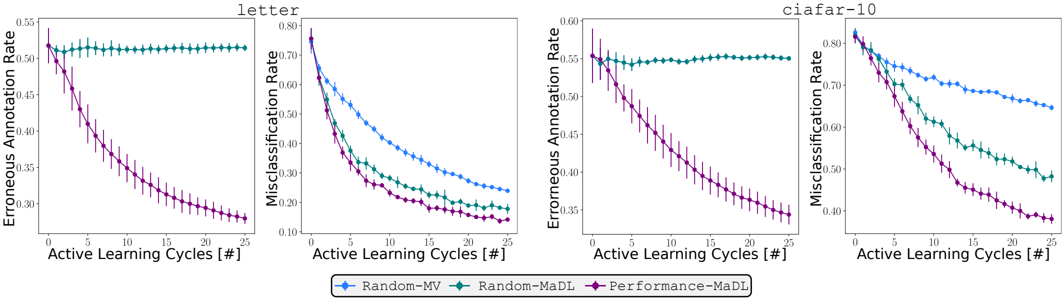
Setup

- Training of a ResNet-18 for `cifar-10` and a multi-layer perceptron for `letter`
- Ten annotators are simulated as adversarial, class-, or cluster-specialized
- 16 samples are randomly selected per annotator for initialization
- 256 samples are randomly selected per active learning cycle

Approaches	Training	Sample Selection	Annotator Selection
`Random-MV`	`MajorityVote`	random	random
`Random-MaDL`	`MaDL`	random	random
`Performance-MaDL`	`MaDL`	random	performance-based

Multi-annotator Active Learning: Evaluation

Results



Takeaway: Annotators' performances estimated by multi-annotator learning can be leveraged to improve the annotator selection.

Herde, Huseljic, Sick, Bretschneider, et al., "Who knows best? A Case Study on Intelligent Crowdworker Selection via Deep Learning", 2023

IAL

Multi-annotator Active Learning: Evaluation

Limitations and Outlook

Limitations:

- The results are restricted to real-world datasets with simulated error-prone annotators
- Samples are selected randomly for annotation
- The number of actively requested annotations per sample is fixed at 1

Outlook:

- Evaluation using datasets with class labels from error-prone humans
- Employing more advanced multi-annotator learning approaches
- Developing query strategies balancing the exploration-exploitation trade-off in the sample and annotator space

Lunch Break (13:00–14:00)

14:00–15:50	**Session 3: Keynote & Workshop Contributions**	
14:00–14:15	Deep Transfer Hashing for Adaptive Learning on Federated Streaming Data	M. Röder & F.-M. Schleif
14:15–14:35	General Reusability: Ensuring Long-Term Benefits of Deep Active Learning	P. Hahn, D. Huseljic, M. Herde & B. Sick
14:35–14:55	Suitability of Modern Neural Networks for Active and Transfer Learning in Surrogate-Assisted Black-Box Optimization	M. Holena & J. Koza
14:55–15:15	Amortized Active Learning for Nonparametric Functions	C.-Y. Li, M. Toussaint, B. Rakitsch & C. Zimmer
15:15–15:30	Towards Deep Active Learning in Avian Bioacoustics	L. Rauch, D. Huseljic, M. Wirth, J. Decke, B. Sick & C. Scholz
15:30–15:50	Active Learning with Physics-Informed Graph Neural Networks on Unstructured Meshes	J. Decke, A. Heinen, B. Sick & C. Gruhl

Coffee Break (15:50–16:20)

Coffee Break (15:50–16:20)

16:20–17:45	**Session 4: Workshop Contributions & Closing**	
16:20–16:40	Combining Large Language Model Classifications and Active Learning for Improved Technology-Assisted Review	M. P. Bron, B. Greijn, B. M. Coimbra, R. van de Schoot & A. Bagheri
16:40–17:00	Contextual kNN Ensemble Retrieval Approach for Semantic Postal Address Matching	E. M. Faraoun, N. Mellouli, M. Lamolle & S. Millot
17:00–17:45	Round Table & Closing	

Thank you!