

02-750 Week 4

Automation of Scientific Research

Aidan Jan

February 5, 2026

Quantifying Committee Disagreement (continued)

Let $\mathcal{M} = \{h_1, \dots, h_p\}$ be a diverse set of models (i.e., committee members)

Soft Vote Entropy

- Method 2: **Soft Vote Entropy** (for classification)

$$x_{SVE}^* = \arg \max_{x \in \mathcal{U}} - \sum_{y \in \mathcal{C}} P_{\mathcal{M}}(y|x) \log P_{\mathcal{M}}(y|x)$$

where $P_{\mathcal{M}}(y|x) = \frac{1}{|\mathcal{M}|} \sum_{h \in \mathcal{M}} P_h(y|x)$ is the consensus probability that the true label is $y \in \mathcal{C}$ and $P_h(y|x)$ is the probability that model h assigns label y for point $x \in \mathcal{U}$

- Soft vote entropy takes the confidence of each committee member into consideration.

KL Divergence

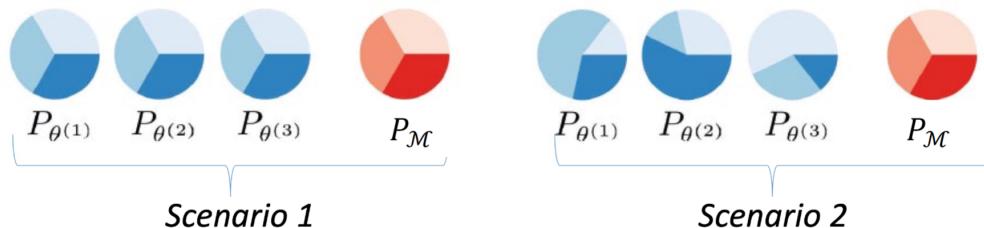
- Method 3: **Kullback-Leibler (KL) Divergence**

$$x_{KL}^* = \arg \max_{x \in \mathcal{U}} \frac{1}{|\mathcal{M}|} \sum_{h \in \mathcal{M}} D(P_h(y|x) || P_{\mathcal{M}}(y|x))$$

where $D(P_h(y|x) || P_{\mathcal{M}}(y|x)) = \sum_{y' \in \mathcal{C}} P_h(y'|x) \log \frac{P_h(y'|x)}{P_{\mathcal{M}}(y'|x)}$ and $P_{\mathcal{M}}(y|x) = \frac{1}{|\mathcal{M}|} \sum_{h \in \mathcal{M}} P_h(y|x)$ is the “consensus” probability that the true label is $y \in \mathcal{C}$ (as defined in the Soft Vote Entropy approach) for point $x \in \mathcal{U}$.

KL Divergence measures the **average divergence** of each committee member’s prediction(s) from the consensus distribution.

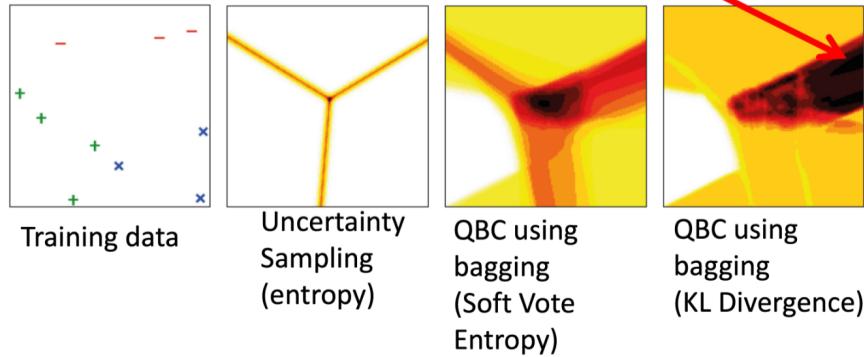
Comparing Soft Vote Entropy vs. KL Divergence



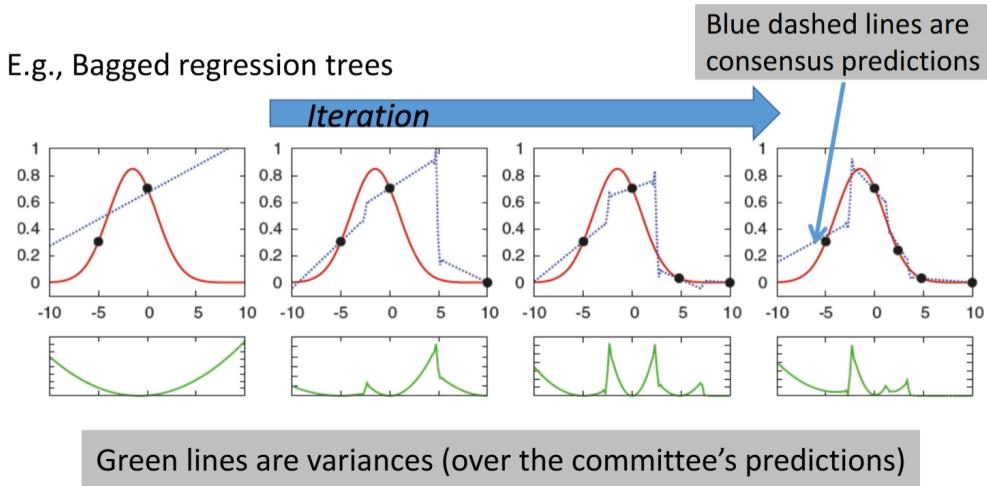
- Blue pie charts represent label probabilities for each of three committee members. Red pie charts are the consensus probabilities.
- Soft vote entropy cannot distinguish between these two scenarios, but the KL Divergence method can.

Example using logistic regression:

KL Divergence reflects regions of **high uncertainty and disagreement** among committee members



For regression, disagreement is typically measured as the variance among the predictions of the committee members



Expected Model Change

Given $D_L = \{(x_1, y_1), \dots, (x_n, y_n)\}$, many learning algorithms optimize parameters by iteratively following the gradient of an objective function: $\theta_{t+1} = \theta_t - \mu \nabla \mathcal{L}_\theta(D_L)$, where μ is the learning rate and $\nabla \mathcal{L}_\theta(D_L)$ is the gradient of the loss function. For example, mean squared error for linear regression:

- Model: $y = b_0 + b_1 x; \theta = (b_0, b_1)$
- Mean squared error: $\mathcal{L}_{\theta=(b_0, b_1)}(D_L) = \frac{1}{n} \sum_{i=1}^n [y_i - (b_0 + b_1 x_i)]^2$
- Gradient: $\nabla \mathcal{L}_\theta(D_L) = \left(\frac{\partial \mathcal{L}}{\partial b_0}, \frac{\partial \mathcal{L}}{\partial b_1} \right)$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial b_0} &= -\frac{2}{n} \sum_{i=1}^n (y_i - (b_0 + b_1 x_i)) \\ \frac{\partial \mathcal{L}}{\partial b_1} &= -\frac{2}{n} \sum_{i=1}^n x_i (y_i - (b_0 + b_1 x_i))\end{aligned}$$

- The **magnitude of the gradient** is proportional to the change in the model parameters during each iterative step, thus, we can *also* use it to select points.
- This approach selects instances that are more likely to significantly change the parameters.

Let $\nabla \mathcal{L}_\theta(D_{\mathcal{L}} \cup \{(x, \hat{y})\})$ be the gradient of the loss function if we were to add the “labeled” point (x, \hat{y}) to $D_{\mathcal{L}}$, where \hat{y} is a predicted label.

- Since we don’t know x ’s true label, we compute the **expected magnitude**.

$$x_{EGL}^* = \arg \max_{x \in \mathcal{U}} - \sum_{y \in \mathcal{C}} P_\theta(y|x) \|\nabla \mathcal{L}_\theta(D_{\mathcal{L}} \cup \{(x, y)\})\|$$

where $\|\cdot\|$ is some norm function.

- $P_\theta(y|x)$ is a class label prediction for the current $x \in \mathcal{U}$
- $\nabla \mathcal{L}_\theta(D_{\mathcal{L}} \cup \{(x, y)\})$ is the new gradient that would be obtained by adding the current instance, (x, y) to $D_{\mathcal{L}}$

Disadvantages

- Expected model change can be expensive, computationally, because the algorithm needs to compute the gradient for each (instance, label) pair
 - If the objective function is differentiable, the cost of computing the gradient is proportional to the dimensionality of the data.
 - If the objective function is not differentiable, you can use a derivative-free approach for estimating and following the gradient (ex. Nelder-Mead) but those tend to be expensive.
- That said, in the context of scientific research, the cost of performing these calculations may be much smaller than performing the experiments themselves.

Minimizing Expected Risk (in Machine Learning)

In machine learning, a **risk objective** (aka **decision rule**) defines what ‘best’ means. For example, expected loss, minimax loss, maximum likelihood, etc.

- Risk refers to the idea of expected generalization error
 - i.e., the model’s error on data it **was not** trained on
- Notice that none of the previous query selection strategies consider risk
 - Uncertainty sampling only considers the confidence of the current model
 - QBC only considers the degree of disagreement amongst the committee members
 - Expected model change only considers the size of change in parameters
- This is odd because generalization error is the only thing that matters
 - The next method attempts to select points that are predicted to reduce risk.
 - It does this by explicitly learning separate models, **each conditioned on one of the possible labels for each unlabeled instance**

For example, consider the **Expected 0-1 loss** (i.e., classification error). Here, we are assuming an unlabeled pool \mathcal{U}

$$x_{ER}^* = \arg \min_{x \in \mathcal{U}} \sum_{y \in \mathcal{C}} P_\theta(y|x) \left(\sum_{x' \in \mathcal{U}} 1 - P_{\theta+(x,y)}(\hat{y}|x') \right)$$

- $P_\theta(y|x)$ is the class label prediction for the current $x \in \mathcal{U}$.
- \hat{y} is the most likely label for x' under a new model trained on $D_{\mathcal{L}} \cup \{(x, y)\}$
- $P_{\theta+(x,y)}(\hat{y}|x')$ is the conditional probability of under the new label
- $1 - P_{\theta+(x,y)}(\hat{y}|x')$ is the expected 0-1 error for current instance $(x', \hat{y})_{x' \neq x}$
- The term in the parenthesis in the expected 0-1 loss over \mathcal{U} for a new model trained on $D_{\mathcal{L}} \cup \{(x, y)\}$

Note: like expected model change, we do not know the true label for each query instance ($x \in \mathcal{U}$), so we approximate using expectation over all possible labels ($y \in \mathfrak{C}$) under the current model θ . The objective here is to reduce the expected total number of incorrect predictions.

Minimizing the expected risk is **much more expensive** than any of the other query selection methods we have seen, because you need to train many models on each iteration

- A binary logistic regression model would require $O(UNG)$ time-complexity simply to choose the next query.
 - U - size of unlabeled pool (i.e., $|\mathcal{U}|$)
 - N - size of the current training set (i.e., $|D_{\mathcal{L}}|$)
 - G - number of gradient computations required by the optimization procedure until convergence
- Once again, in the context of scientific research, the cost of performing these calculations may be much smaller than performing the experiments themselves.

Density-based Sampling

- The query selection strategies we have studied so far are **myopic**, in the sense that they select instances without considering the relationships between the points in the domain/pool
- This could potentially lead to problems. For example, suppose a far outlier point that happens to be on the decision boundary between two labels. Is it really worth getting the label for that point if it is not representative of any other points (e.g., an outlier)?

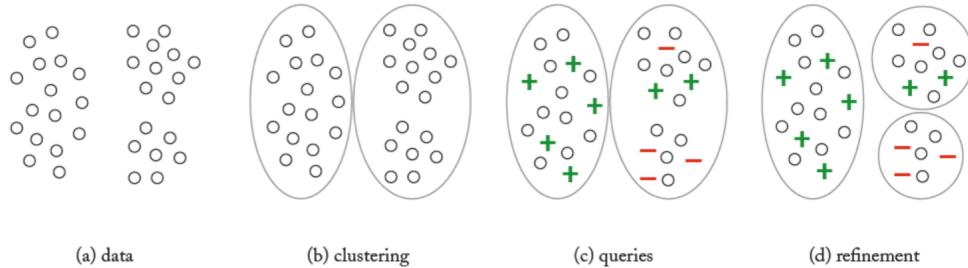
For example, consider the information density framework, which is a general weighting technique. Here, we are assuming an unlabeled pool \mathcal{U} .

$$x_{ID}^* = \arg \max_{x \in \mathcal{U}} \phi_A(x) \left(\frac{1}{|\mathcal{U}|} \sum_{x' \in \mathcal{U}} sim(x, x') \right)^\beta$$

- Here, $\phi_A(x)$ is a function that returns the utility of the point $x \in \mathcal{U}$.
- ϕ can be any query strategy that we have covered (e.g., uncertainty sampling, query by committee)
- Function $sim(x, x')$ returns some measure of similarity between point x and x' .
 - The term in the parenthesis in the **average similarity** of x to all other instances in \mathcal{U} , which is a measure of the local density around x .
 - β is a parameter that controls the relative importance of the density term.
- A variant of this approach might first cluster \mathcal{U} and compute average similarity to instances in the same cluster.

Exploiting Structure in the Data

- Density based sampling is an example of a query selection method that exploits the **structure** in an (unlabeled) data pool
- Later in the semester, we will study active learning algorithms that cluster the unlabeled samples, and then use the clusters to guide query selection.



Summary: Heuristic Query Selection Strategies

Method	Advantages	Disadvantages
Uncertainty Sampling	Fast and easy to implement	Myopic and may become overconfident
Query by Committee	Easy to implement	Myopic and requires multiple models
Expected Change	May accelerate convergence to best model	Multiple gradient calculations
Risk Minimization	Optimizes the objective we actually care about	Can be very expensive
Density-based	Not myopic	Requires $\Omega(n^2)$ work to compute $\binom{n}{2}$ pairwise similarities

There are many different query selection methods

- The ones we studied this week are merely heuristics
- We can't prove anything about them, with respect to statistical consistency or efficiency

Later, we will see several query selection methods that come with formal guarantees

Batch Mode Active Learning

- **Intuition:** Select a set of k unlabeled instances that maximize (or minimize) some criterion (e.g., Fisher information)
 - A simple strategy: Select the top k most informative instances
 - Some of the selected instances could be similar to each other, and therefore do not provide additional information for model updating
- **Goal:** Select instances that are informative such that each selected instances is different from the others and provides unique information
- **Challenge:** How to efficiently identify the subset of unlabeled instances under specified criterion?
 - Computationally complex to consider all combinations of potential instances
 - Do not know labels

Optimal Design of Experiments

- Batch - choose multiple instances in one group for experimentation (One or two batches for DOE.)

Is Active Learning an Appropriate Strategy?

Experiment execution time takes much, much longer than modeling and instance selection time.

- We have already improved efficiency of running experiments
- Miniaturization leads to more experimentation. (e.g., Use a 1536 well plate instead of a 384 well plate)
- Combining miniaturization with instance selection allows for lower per compound test time even if the total time spent is still lengthy!
- Need to choose compounds in batches to maximize efficiency!

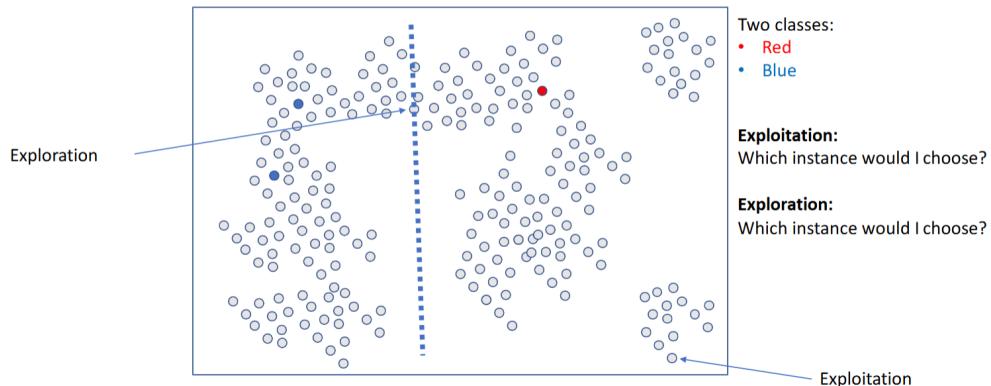
High-Level Query Selection Criteria: Exploration vs. Exploitation

Exploration

- Select designs that are from unexplored regions of the design space, \mathcal{X}
 - **Strength:** We gain knowledge that might help us build a better model (i.e., one that is better at directing us towards optimal designs)
 - **Weakness:** We may waste effort (opportunity loss)

Exploitation

- Select designs that are predicted to be optimal under the current model, h
 - **Strength:** Leverage the information we have
 - **Weakness:** We may get stuck in a local optimum (opportunity loss)



- What happens if our next 3 points are the ones on the line (e.g., all exploration)?
 - Not a great idea, since we only need one of those points to resolve the other two! We will waste two of the three experiments in that case.
- What happens if our next 3 points are all as far away as possible (e.g., all exploitation)?
 - If you do this, then you have absolutely no clue where the decision boundaries are!

Ideally, we want to pick some points for exploration and some points for exploitation.

Diversity-Based Sampling Strategies

- Cluster-based sampling - cluster the data and choose representative instances from each cluster
- How could you choose n instances in a **hierarchical clustering** mode?
 - Algorithm: Identify level of tree yielding n clusters. Choose randomly from those clusters.
- How could you choose n instances in a **k-means** mode?
 - Algorithm: Set $k = n$, and choose one instance from each cluster.

Discriminative Batch Mode Active Learning

Key features

- It presents a framework for “batch mode active learning” that applies the Fisher information matrix to measure the overall informativeness for a set of unlabeled instances
- Propose an efficient greedy algorithm that is based on the property of **submodular functions**
- Empirical studies show that the proposed batch mode active learning algorithm is more effective than the traditional algorithms for active learning.

Batch Mode Active Learning Set-up

Approach:

- Given that the Fisher information matrix represents the overall uncertainty of a classification model, the goal is to search for a set of instances that can most efficiently reduce the Fisher information matrix

Notation:

- $p(x)$ be the distribution of all unlabeled instances
- $q(x)$ be the distribution of unlabeled instances that are chosen for labeling
- α denotes the parameters of the classification model
- $I_p(\alpha)$ and $I_q(\alpha)$ denote the Fisher information matrix of the classification model for the distribution $p(x)$ and $q(x)$, respectively.
- $D = (x_1, \dots, x_n)$ be the unlabeled data
- $S = (x_1^5, \dots, x_k^5)$ be the subset of selected instances

The Fisher matrices $I_p(\alpha)$ and $I_q(\alpha)$ can be computed as:

$$I_p(\hat{\alpha}) = \frac{1}{n} \sum_{x \in \mathcal{D}} \pi(x)(1 - \pi(x))xx^T + \delta I_d$$

$$I_q(S, \hat{\alpha}) = \frac{1}{k} \sum_{x \in S} \pi(x)(1 - \pi(x))xx^T + \delta I_d$$

where

$$\pi(x) = p(-|x) = \frac{1}{1 + \exp(\hat{\alpha}^T x)}$$

- $\hat{\alpha}$ stands for the classification model that is estimated from the labeled instances
- I_d is the identity matrix of size $d \times d$

- $\delta << 1$ is the smoothing parameter
- δI_d is added to the estimation of $I_p(\hat{\alpha})$ and $I_q(\hat{\alpha})$ to prevent them from being singular matrices

The final optimization problem for batch mode active learning is formulated as follows:

$$S^* = \arg \min_{\substack{S \subseteq D \\ |S|=k}} \text{tr}(I_q(S, \hat{\alpha})^{-1} I_p(\alpha))$$

Challenge: The number of candidate sets for S is exponential in the number of unlabeled examples n

Batch Mode Active Learning: Algorithm

The key idea of this approach is to explore a general theorem about submodular functions (Newhauser et al., 1978)

- Consider the optimization problem that searches for a subset S with k elements to maximize a set function $f(S)$

```

Initialize S = empty set
For i = 1, 2, ..., k
    Compute x* = argmax_{x not in S} f(S + x) - f(S)
    Set S += x*

```

- If $f(S)$ is (i) a nondecreasing submodular function, and (ii) $f(\emptyset) = 0$, then the above greedy algorithm will guarantee a performance $(1 - \frac{1}{e}) f(S^*)$, where $S^* = \arg \max_{|S|=k} f(S)$ is the optimal subset

Meta-Learning for Batch Mode Active Learning

Quality Distribution Using the set of statistics for each unlabeled item, we can compute the probability of selecting an unlabeled item according to its quality as

$$p_{\text{quality}}(\tilde{x}_i) \propto \exp(q_i), \text{ where } q_i = f_q(\Pi(\{c_k\}_{k=1}^K, \tilde{x}_i)),$$

where f_q is a MLP with parameters q . This distribution independently maps the probability of each unlabeled item being selected based on a prediction of how useful the item will be to the existing classifier according to a learned function of item-classifier statistics.

Diversity Distribution The same set of statistics can also be used to compute a feature vector describing the unlabeled item to classifier relationship as

$$\phi_i = f_\phi(\Pi(\{c_k\}_{k=1}^K, \tilde{x}_i)),$$

where $\phi_i \in \mathbb{R}^{D'}$ and f_ϕ is a MLP with parameters ϕ . The goal of the diversity distribution is to increase the probability of selecting unlabeled items which are dissimilar from the items that already make up the set $\mathcal{A} = \{\tilde{x}_1, \dots, \tilde{x}_j\}$ where similarity is measured in terms of each item's corresponding feature vector. The probability of selecting an unlabeled item according to its diversity is then:

$$p_{\text{diversity}}(\tilde{x}_i | \mathcal{A}) \propto \exp(v(\phi_i)/\tau), \text{ where } v(\phi_i) = \min_{\tilde{x}_j \in \mathcal{A}} \{\sin \theta_{ij}\},$$

where θ_{ij} is the angle between feature vectors ϕ_i and ϕ_j and τ is a learned temperature parameter that allows us to control the flatness of this distribution. The probability of an item being picked increases as its feature vector is more orthogonal to feature vectors corresponding to items already having been added to the subset \mathcal{A} .

Product of Experts The final probability distribution over which unlabeled item to add to the subset $p(\tilde{x} | \mathcal{A})$ is attained as a product of experts model combining the distributions p_{quality} and $p_{\text{diversity}}$:

$$p(\tilde{x} | \mathcal{A}) \propto p_{\text{quality}}(\tilde{x}) \cdot p_{\text{diversity}}(\tilde{x} | \mathcal{A}) \cdot \mathbb{1}_{\tilde{x} \notin \mathcal{A}},$$

where the indicator variable enforces not having any support over an item that already belongs to \mathcal{A} .

Goal: Compute the probability of selecting an unlabeled item according to its quality

- $p_{\text{quality}}(\tilde{x})$ is the probability of unlabeled item being selected based on a prediction of how useful it would be to existing classifier

Goal: Increase probability of selecting unlabeled items which are dissimilar from the items already selected

- $p_{\text{diversity}}(\tilde{x} | \mathcal{A})$ is the probability of an item being picked increases as its feature vector is more orthogonal to feature vectors corresponding to the items already selected

Goal: Compute final probability distribution over which unlabeled item to add to the subset $p(\tilde{x} | \mathcal{A})$

- $p(\tilde{x} | \mathcal{A}) \propto p_{\text{quality}}(\tilde{x}) \cdot p_{\text{diversity}}(\tilde{x} | \mathcal{A}) \cdot \mathbb{1}_{\tilde{x} \notin \mathcal{A}}$

Summary of Batch Selection Strategies

- Batch mode strategies can improve the efficiency of scalable processes in an active learning context
- They mostly require consideration of exploration and exploitation in the design of the batches