02-750 Week 4
Automation of Scientific Research

Aidan Jan

February 3, 2026

Quantifying Committee Disagreement (continuued)

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

Soft Vote Entropy

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

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

  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 $y \in \mathfrak{C}$ 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

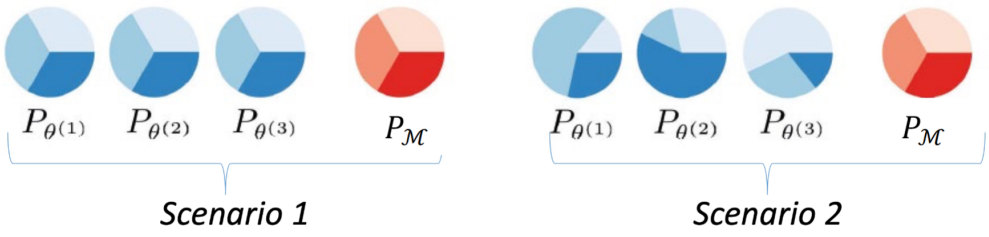
- Method 3: **Kullback-Leibler (KL) Divergence**

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

  where $D(P_h(y|x)||P_{\mathcal{M}}(y|x)) = \sum_{y' \in \mathfrak{C}} P_h(y'|x) \log \frac{P_h(y'|x)}{P_{\mathcal{M}}(y'|x)}$ 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 $y \in \mathfrak{C}$ (as defined in the Soft Vote Entropy approach) for point $x \in \mathcal{U}$.

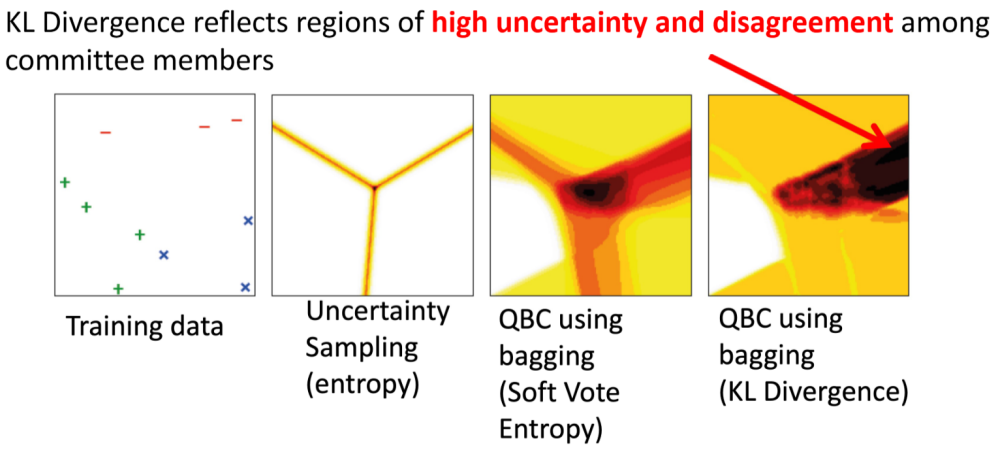
KL Divergence measures the **average divergence** of each committee member's prediction(s) from the consensus distribution.

Comparing Soft Vote Entropy vs. KL Divergence

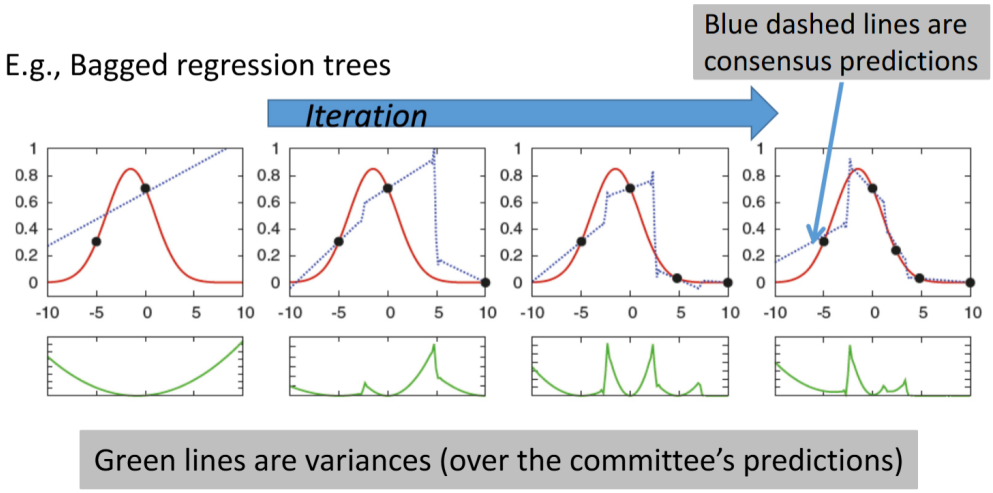


- Blue pie charts represent label probabilities for each of three committee members. Red pie charts are the consensus probabilities.
- Soft vote entropy cannot distinguish between these two scenarios, but the KL Divergence method can.

Example using logistic regression:



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



Expected Model Change

Given $D_L = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, 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: $\nabla \mathcal{L}_\theta(D_L) = \left(\frac{\partial \mathcal{L}}{\partial b_0}, \frac{\partial \mathcal{L}}{\partial b_1} \right)$

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

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

- 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**.

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

  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 selected 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 $\mathcal{U}$

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

- $P_\theta(y|x)$ 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)?

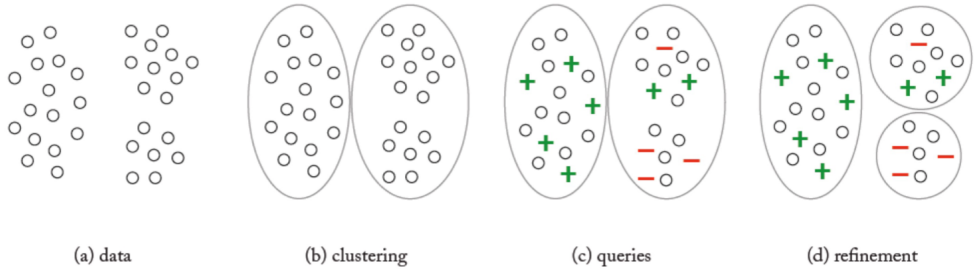
For example, consider the information density framework, which is a general weighting technique. Here, we are assuming an unlabeled pool $\mathcal{U}$.

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

- 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 approah 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	East 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