Active Learning

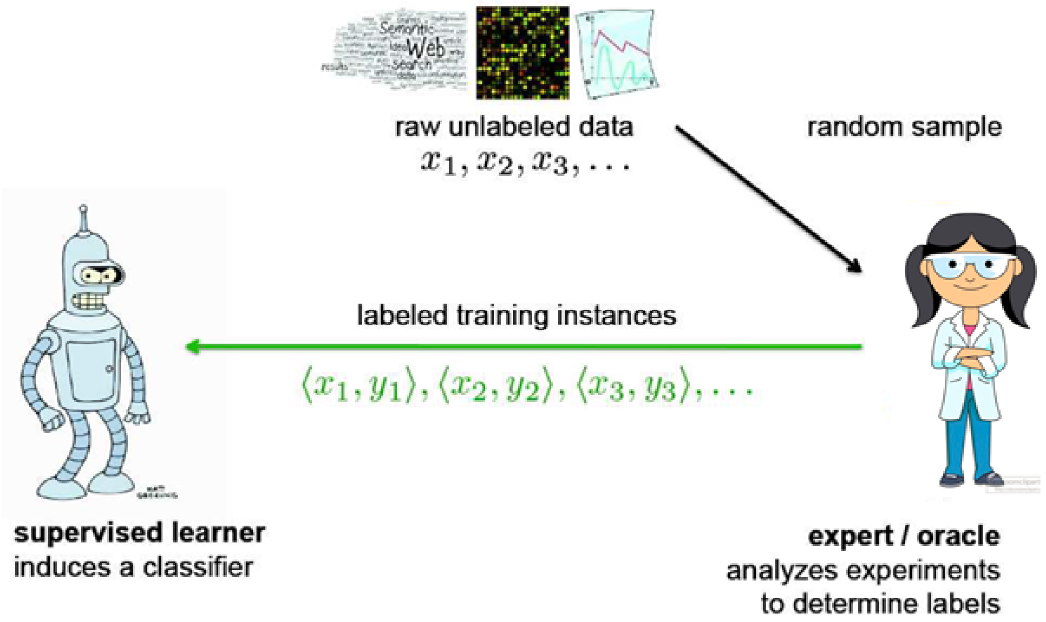
NASIM ZOLAKTAF

MACHINE LEARNING READING GROUP
THE UNIVERSITY OF BRITISH COLUMBIA
JULY 18, 2017

SLIDES ADAPTED FROM PIYUSH RAI

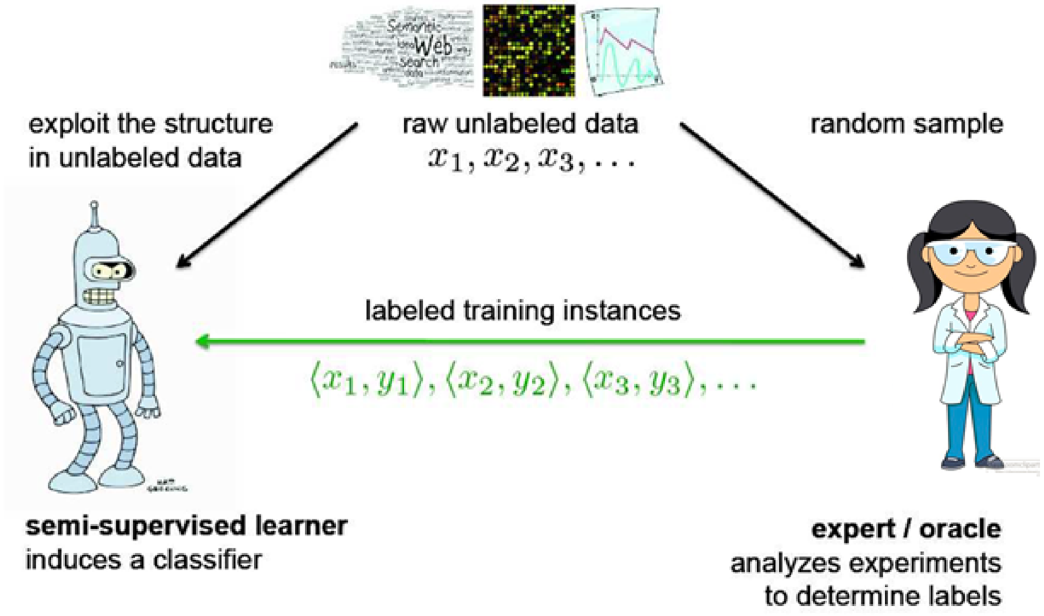
Supervised Learning

- Trains on labeled examples



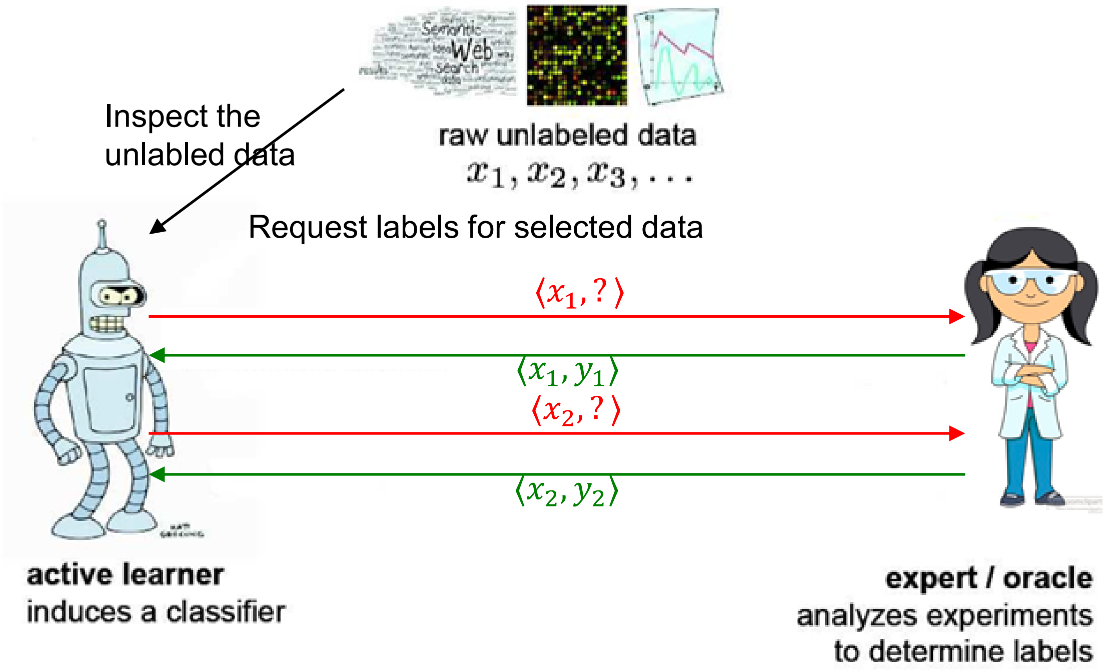
Semi-supervised Learning

- Trains on labeled and unlabeled examples



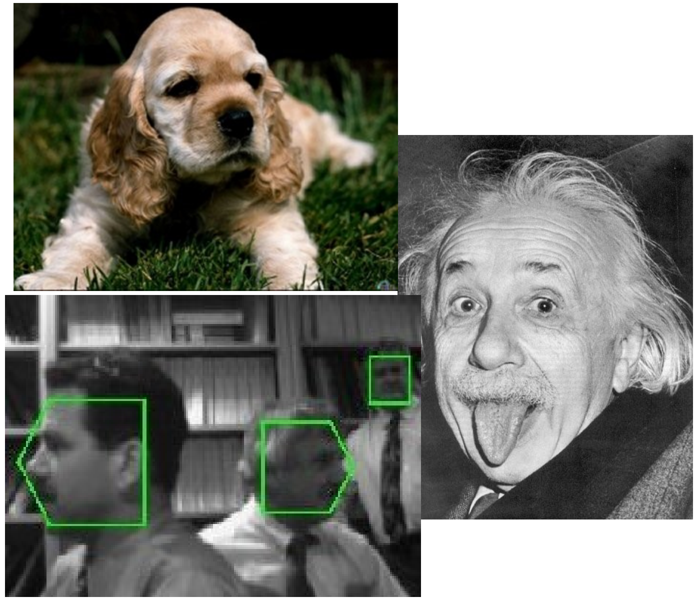
Active Learning

- Assumes some amount of initial labeled training data
- Incrementally requests labels for examples



Active Learning

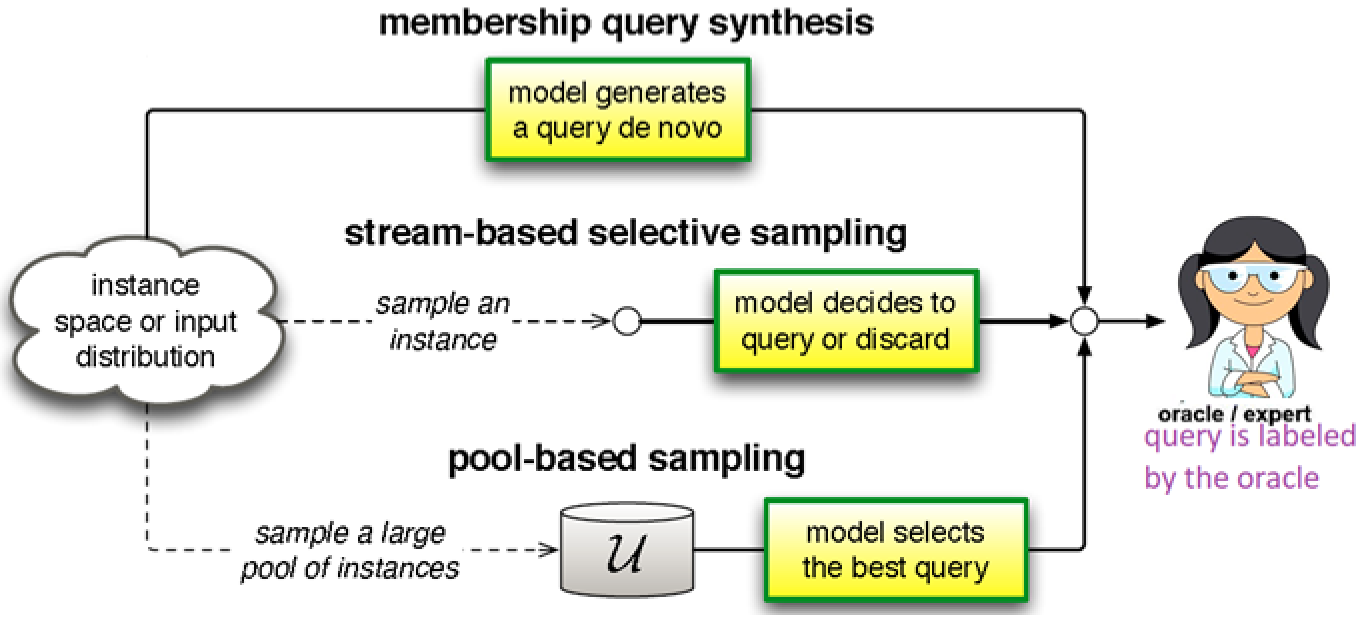
- Unlabeled data is abundant, but labels are time-consuming/expensive.
- Active learning is a useful model here.
  - Allows for intelligent choices of which examples to label.
- Goal: aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data.
- Applications
  - Speech Recognition
    - 10 mins to annotate words in 1 min of speech
    - 7 hrs to annotate phonemes of 1 minute speech
  - Image annotation
  - …



How Active Learning Operates

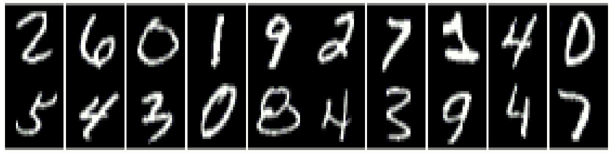
- Active learning proceeds in rounds
- Each round has a current model (learned using the labeled data seen so far)
- The current model is used to assess informativeness of unlabeled examples
  - .. using one of the **query selection strategies**
  - The most informative example(s) is/are selected
  - The labels are obtained (by the labeling oracle)
  - The (new) labeled example(s) is/are included in the training data
  - The model is re-trained using the new training data
- The process is repeated until we have budget left for getting labels

Active Learning Scenarios



Membership Query Synthesis

- Generate a query and request the label (query synthesis)
- Might be awkward for human annotators
  - For example, generate text/speech/image might be meaningless



- Real-world applications
  - Robot scientist: discover metabolic pathways in a yeast
    - An instance is a mixture of chemical solutions that constitute a growth medium, as well a particular yeast mutant
    - A label is whether or not the mutant thrived in the growth medium
    - Experiments are autonomously synthesized using inductive logic programming and physically performed with a laboratory robot
    - 3x $ decrease vs. cheapest next, and 100x $ decrease vs. random selection

Stream-Based Selective Sampling

- Assumption: unlabelled instance comes at no or minimal cost
- First sample an unlabeled instance
- Then decide whether to query its label or to ignore it
  - Informativeness measure: more informative instances are more likely to be queried
  - Region of uncertainty: only query instances that fall within it
- Real-world problems: part-of-speech tagging, sensor scheduling, word sense disambiguations, e.g., determining the meaning of bank based on context

Pool-Based Sampling

- Assumption: large pool of unlabled instances gathered at once
- Rank examples in order of informativeness
- Query the labels for the most informative examples
- Real-word problems: cancer diagnosis, text classification, image classification & retrieval, video classification & retrieval
- Stream-based vs pool-based active learning
  - The stream-based scenario scans through the data sequentially and makes query decisions individually, whereas the pool-based scenario evaluates and ranks the entire collection before selecting the best query
  - The pool-based scenario appears to be more common in applications
  - Sometimes stream-based active learning could be more appropriate. For example, when memory or processing power may be limited, as with mobile and embedded devices

Query Strategy Frameworks

- All types of active learning require a query selection strategy
- Examples:
  - Uncertainty Sampling
  - Query-By-Committee (QBC)
  - Expected Error Reduction
  - Variance Reduction

Uncertainty Sampling

- Query examples which the current model θ is the most uncertain about
- Various ways to measure uncertainty. For example:
  - Based on the distance from the hyperplane
  - Using the label probability $P_\theta(y|x)$ (for probabilistic models):
    - **Least Confident**:

$$x^*_{LC} = \text{argmax}_x 1 - P_\theta(\hat{y}|x)$$

      where $\hat{y} = \text{argmax}_y\ P_\theta(y|x)$

    - **Smallest Margin:**

$$x^*_{SM} = \text{argmin}_x P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)$$

      where $\hat{y}_1$ and $\hat{y}_2$ are the two most probable labels for x under the current model

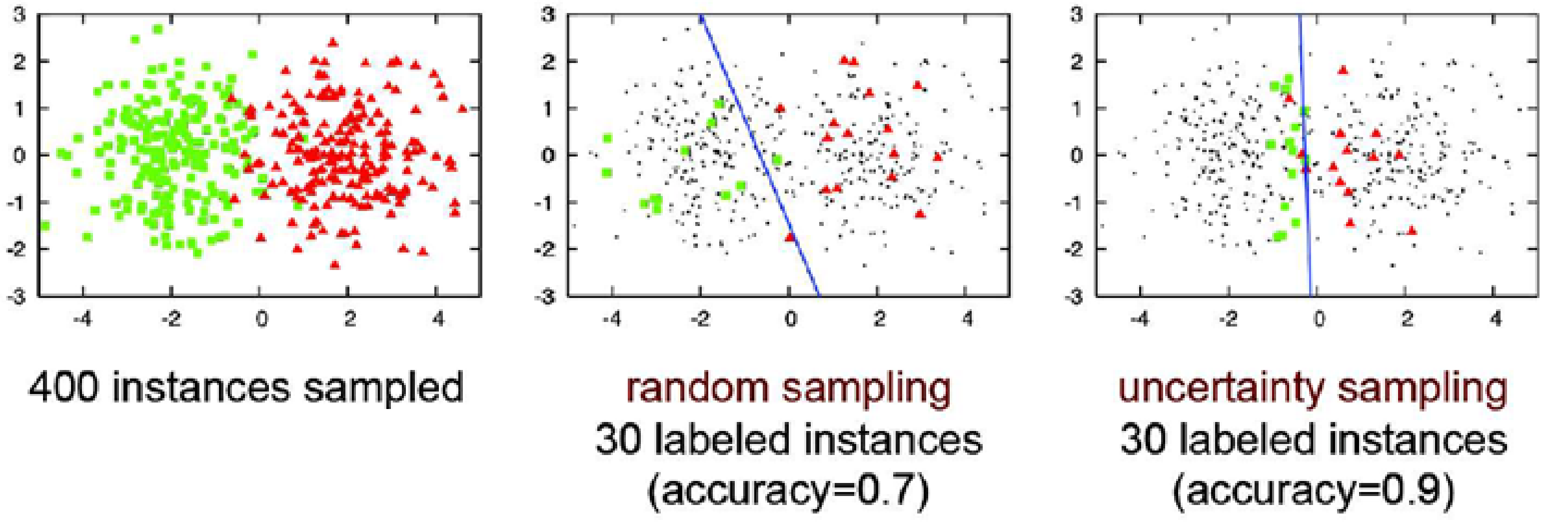
    - **Label Entropy:**

$$x^*_{LE} = \text{argmax}_x - \sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$$

      where y_i ranges over all possible labels

Uncertainty Sampling

- A simple illustration of uncertainty sampling based on the distance from the hyperplane



Query-By-Committee (QBC)

- QBC uses a committee of models $C = \{\theta^{(1)}, \ldots, \theta^{(C)}\}$
- All models trained using the currently available labeled data $\mathcal{L}$
- Different ways to construct committee, e.g., using bagging/boosting ensemble methods
- All models vote their predictions on the unlabeled pool
- The example(s) with maximum disagreement is/are chosen for labeling
- One way of measuring disagreement is the Vote Entropy (a QBC generalization of entropy-based uncertainty sampling)

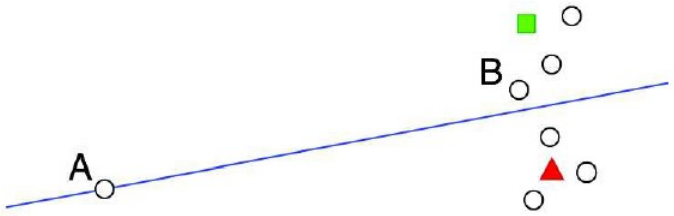
$$x^*_{VE} = \mathrm{argmax}_x - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}$$

  y_i ranges over all possible labels, $V(y_i)$ is the number of votes received to label y_i, C is the committee size

- Each model in the committee is re-trained after including the new example(s)

Effect of Outlier Examples

- Uncertainty Sampling or QBC may wrongly think an outlier to be an informative example
- Such examples won't really help (and can even be misleading)



- Other robust query selection methods exist to deal with outliers
- Idea: Instead of using the confidence of a model on an example, see how a labeled example affects the model itself
  - The example(s) that affects the model the most is probably the most informative

Expected Error Reduction

- Select example that reduces the expected generalization error the most, measured w.r.t. the remaining unlabeled examples (representative of the test distribution)
- Minimize the expected 0/1-loss

$$x^*_{0/1} = \underset{x}{\operatorname{argmin}} \sum_i P_\theta(y_i|x) \left(\sum_{u=1}^{U} 1 - P_{\theta^{+\langle x, y_i \rangle}}(\hat{y}|x^{(u)}) \right)$$

where $\theta^{+\langle x, y_i \rangle}$ refers to the new model after it has been re-trained with the training tuple $\langle x, y_i \rangle$ added to $\mathcal{L}$, U is the set of unlabled examples

- We do not know the true label for queries, so we approximate using expectation over all possible labels under the current model θ

Expected Error Reduction

- Computationally expensive for most tasks, e.g. $O(TM^{T+2}ULG)$ for a sequence labeling task using CRFs, where T is the length of input sequence and M is the number of labels
- Usually used for binary classification tasks.
- Binary logistic regression is $O(ULG)$ to choose the next query, where U is the set of unlabled examples, L is the size of the current training set $\mathcal{L}$, and G is the number of gradient computations for optimization convergence.
- Solution: use Monte Carlo sampling from the pool to reduce the U term, or approximate training techniques to reduce the L/G term, etc

Variance Reduction

- Select example(s) that reduces the model variance the most
- Minimizing the variance minimizes the future generalization error
  - $E[(\hat{y} - y)^2|x] = \text{noise} + \text{bias} + \text{variance}$
- The Fisher information sets a lower bound on the variance (Cramer-Rao inequality)
- Maximize Fisher information

Variance Reduction

- For neural networks, under certain assumptions, an expression for $\langle \tilde{\sigma}_{\hat{y}}^2 \rangle^{+x}$, which is the estimated mean output variance across the input distribution after the model has been re-trained on query x and its label, can be estimated efficiently in closed form so that actual model re-training is not required
- Variance reduction: $x_{VR}^* = \underset{x}{\operatorname{argmin}} \ \langle \tilde{\sigma}_{\hat{y}}^2 \rangle^{+x}.$
- Gradient methods can be used to search for the best possible query
- Example of query synthesis

Variance Reduction

- Estimating variance requires inverting a matrix $\rightarrow O(UK^3)$, where U is the size of unlabeled examples, and K is the number of parameters in the model.
- Impractical for large K, such as natural language processing tasks.
- Solution: use sampling approach based on Markov chains to reduce the U term, use principle component analysis for inverting the matrix, approximate matrix with its diagonal matrix, etc.

Other Query Selection Methods

- **Expected Model Change**
  - Select the example whose inclusion brings about the maximum change in the model (e.g., the gradient of the loss function w.r.t. the parameters)
- **Density Weighting**
  - Weight the informativeness of an example by its average similarity to the entire unlabeled pool of examples
  - An outlier will not get a substantial weight!

Thanks!

Questions?

References

- Settles, Burr. "Active learning literature survey." *University of Wisconsin, Madison* 52.55-66 (2010): 11.
- Rai, Piyush. (2017, July 17). *Active learning*. Retrieved from https://www.cs.utah.edu/~piyush/teaching/10-11-print.pdf.