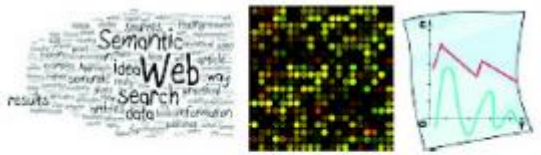
Probabilistic Approaches for Active Learning

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

Passive Learning

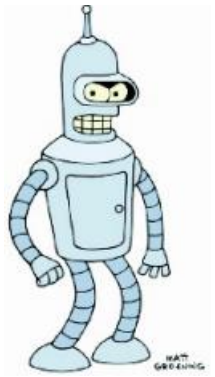
- Standard supervised learning is passive
  - Learner has no control over what labelled training examples it gets to learn from



raw unlabeled data

$x_1, x_2, x_3, \ldots$

Random Unlabeled Examples



Supervised Passive Learner

Labeled Training Examples

$(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots$

expert / oracle
analyzes experiments
to determine labels

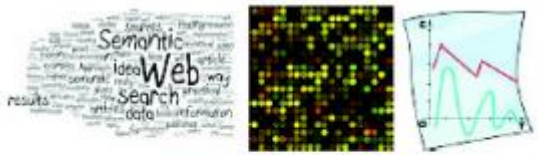
Active Learning

It is therefore also a sequential learning strategy (training data is not given all at once)

- In Active Learning, the learner can request specific labelled examples as it trains
  - In particular, examples that the learner thinks will be most useful to learn the underlying function

Will soon see what are some common notions of usefulness

Using the current model, identify the most useful example(s) from the unlabeled data pool



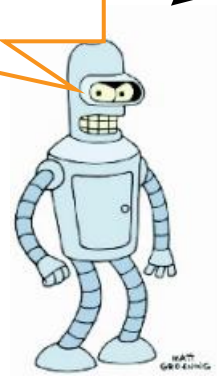
raw unlabeled data

$x_1, x_2, x_3, \ldots$

Although one example in each iteration is more common, labels of one or more than one examples can be queried ("batch" active learning")

Assumes some small amount of initial labelled training examples $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ are available to learn an initial model

More labelled training examples will be acquired "actively" to improve the initial model by retraining it using the updated training data $\mathcal{D} = \{\mathcal{D} \cup (x_*, y_*)\}$ and repeating the process until we get the desired accuracy or our budget exhausts



Query the expert for the true label of the selected unlabeled example, say x_1

$\langle x_1, ? \rangle$

$\langle x_1, y_1 \rangle$

$\langle x_2, ? \rangle$

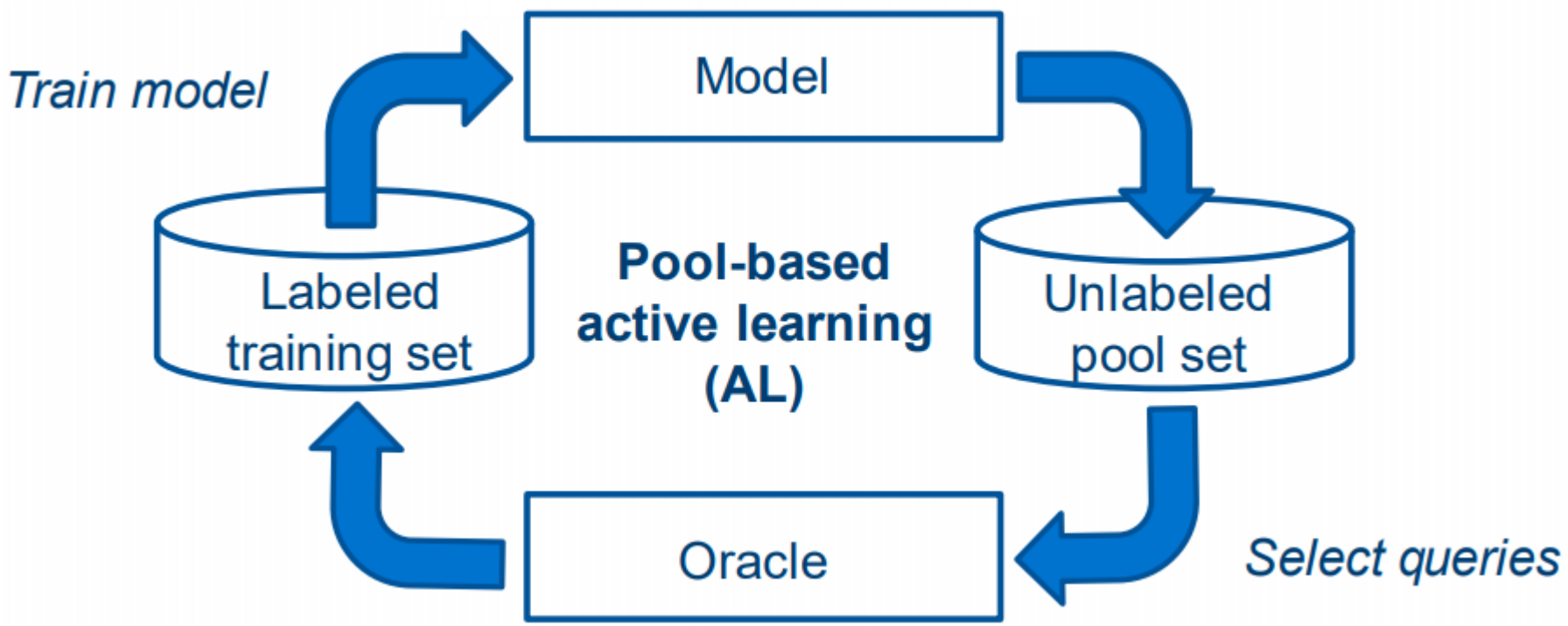
$\langle x_2, y_2 \rangle$

expert / oracle
analyzes experiments
to determine labels

Supervised Active Learner

Active Learning

- The figure below is another illustration of AL

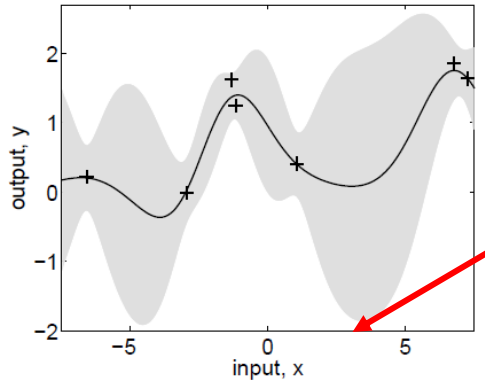


Measuring Usefulness in AL

Given the acquisition function, the most useful example can be selected from the pool as
$\hat{\boldsymbol{x}}_* = \text{argmax}_{x_* \in \mathcal{X}_{pool}} A(\boldsymbol{x}_* | p(\theta|\mathcal{D}))$

Note: We will use shorthand $A(\boldsymbol{x}_*)$

- Various ways to measure the usefulness of an unlabeled example $\boldsymbol{x}_*$
  - Defined by an "acquisition function" $A(\boldsymbol{x}_*)$ (high value for most useful unlabeled examples)
- Approach 1: For $\boldsymbol{x}_*$, look at uncertainty in output $\boldsymbol{y}_*$ predicted by the current model
  - Can use variance in the posterior predictive distribution: $A(\boldsymbol{x}_*) = \text{var}(\boldsymbol{y}_*)$

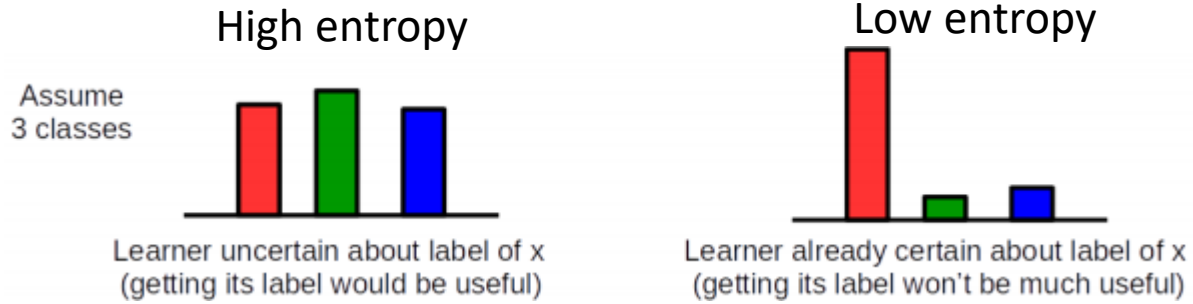


Most "useful/informative" since the model currently has the largest uncertainty for this input

$$\mathbb{H}(p) = -\int p(x) \log p(x) dx$$

AL using this criterion is called maximum entropy sampling

  - More generally, can use entropy of the posterior predictive distribution: $A(\boldsymbol{x}_*) = \mathbb{H}[p(\boldsymbol{y}_* | \boldsymbol{x}_*, \mathcal{D})]$



Note that this is the "marginal entropy" of the output distribution since the posterior predictive of the output is obtained by marginalizing over the posterior

Measuring Usefulness in AL

- Approach 2: Look at how much our model will improve if we add this unlabeled example with its true label, to our training set, and retrain the model
- Reduction in the model's uncertainty is one way to measure this improvement
  - Reduction in the entropy of the model's posterior distribution can be used for this

$$A(x_*) = \mathbb{H}[p(\theta|\mathcal{D})] - \mathbb{E}_{p(y_*|x_*,\mathcal{D})}\mathbb{H}[p(\theta|\mathcal{D} \cup (\boldsymbol{x}_*, y_*))]$$

Entropy of the current posterior

Need to use expectation here since y_* is not known

Entropy of the new posterior after including the new example in our training set

Often easier to work with entropy of the predictive distribution of y_* than the posterior distribution of θ since θ is usually much higher dimensional than y_*

$$= \mathbb{I}[\theta;\ y_*|\mathcal{D}, \boldsymbol{x}_*] \quad \text{(by definition, conditional mutual information of two r.v.s } \theta \text{ and } y_*\text{)}$$

$$= \mathbb{H}[p(y_*|\boldsymbol{x}_*, \mathcal{D})] - \mathbb{E}_{p(\theta|\mathcal{D})}\mathbb{H}[p(y_*|\boldsymbol{x}_*, \theta)] \quad \text{(due to symmetry of MI)}$$

This is the same marginal entropy term used in maximum entropy sampling – we seek inputs with high value of this quantity

Expected conditional entropy – we seek inputs with low value of this quantity

Basically, we also want to penalize inputs that have high uncertainty just due to the noise in the likelihood model.

Difference between the two terms will be large if the θ's have a large disagreement

Bayesian Active Learning by Disagreement (Houlsby et al, 2012)

Batch Active Learning

- Approaches we saw work by querying and adding one example at a time
- Expensive in practice since we have to retrain every time after including a new example
  - Especially true for deep learning models which are computationally expensive to train
- In practice, we want to use AL to jointly query the labels of $B > 1$ examples

$$(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_B) = \text{argmax}_{(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_B) \in \mathcal{X}_{pool}} A\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_B | p(\theta|\mathcal{D})\right)$$

- Difficult to construct such joint acquisition function and maximize them
- A greedy scheme is to simply select the B highest scoring points

$$A\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_B | p(\theta|\mathcal{D})\right) = \sum_{b=1}^{B} \mathbb{I}[\theta;\, y_b | \mathcal{D}, \boldsymbol{x}_b]$$

Some extensions of BALD use this scheme

- The above however is myopic and ignores correlations among the selected points
  - Recent work[1] has developed joint MI based acquisition functions for Batch BALD

$$\mathbb{H}[p(y_{1:B}|x_{1:B}, \mathcal{D})] - \mathbb{E}_{p(\theta|\mathcal{D})}\mathbb{H}[p(y_{1:B}|x_{1:B}, \theta)]$$

May check out the BatchBALD paper (referenced below) to see how this objective is defined, and how is compares with the greedy approach

[1] BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning (Kirsch et al, NeurIPS 2019)

Batch Active Learning

- A comparison of BALD and BatchBALD

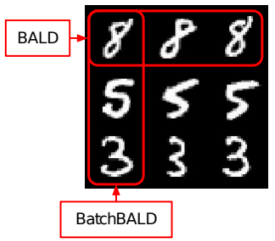


Figure 1: *Idealised acquisitions of BALD and BatchBALD.* If a dataset were to contain many (near) replicas for each data point, then BALD would select all replicas of a single informative data point at the expense of other informative data points, wasting data efficiency.

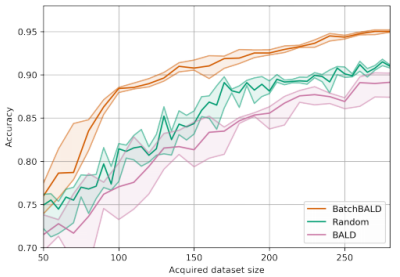
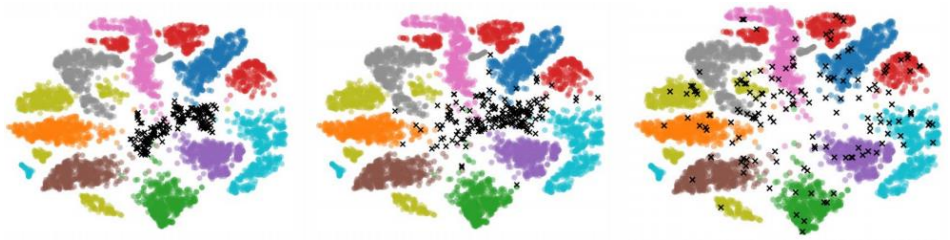


Figure 2: *Performance on* Repeated MNIST *with acquisition size 10.* See section 4.1 for further details. BatchBALD outperforms BALD while BALD performs worse than random acquisition due to the replications in the dataset.

- Other recent work has also proposed efficient ways for Batch AL
  - May see "Bayesian Batch Active Learning as Sparse Subset Approximation" (Pinsler et al, NeurIPS 2019)



Batch AL with MaxEnt | Batch AL with standard BALD | Batch AL with sparse subset approx (above paper)

Figure courtesy: BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning (Kirsch et al, NeurIPS 2019)

Active Learning: A Few Other Comments

- AL provides an efficient way to learn a model using very few labelled examples
  - Saves upon annotation costs especially when we need to pay human labellers
- Probabilistic/Bayesian approaches are natural for AL since we have estimates of the model's uncertainty and predictive uncertainty
  - Many ways to define acquisition functions using such quantities
- The AL techniques we have seen are applicable to a wide variety of models
  - Probabilistic linear models, nonlin. models based on GP, probabilistic deep learning models
- AL algorithms can also be design for unsupervised learning problems
  - Here, we need to decide usefulness without labels
  - The assumption is that unlabeled data is also difficult/costly to obtain

Coming Up

- Bayesian Optimization
  - Also based on the idea of sequential acquisition of data to optimize a function when we don't know the function but can only ask for function's values at a small number of points