Budgeted Batch Mode Active Learning with Generalized Cost and Utility Functions

Arvind Agarwal, Shashank Mujumdar, Nitin Gupta, Sameep Mehta
IBM Research, India

Active Learning

- Supervised learning algorithm requires labeled data which is expensive and time consuming.
- Active learning reduces the labelling cost by actively querying labels for the most valuable data points.
- An ideal active learning algorithm should select examples in a batch that are maximally useful and can be labeled within a budget.
- Traditional algorithms define utility of a single example rather than of a batch, without cost and budget considerations.
- In this work we propose:
  - A learning framework that actively selects optimal set of examples in a batch within a given budget, based on given utility and cost functions
  - A novel utility function based on the facility location problem that takes care of point utility, region utility and sample diversity.
  - A novel cost function that considers that the labelling cost of an example depends on the previously labelled examples.

Active Learning Framework

Objective: Maximize the utility while minimizing the cost which can be written as a knapsack problem in a budget setting

Non-batch Version:

$$\max \sum_{i=1}^{N} v_i \mathbf{x}_i$$
$$\mathbf{x}_i \in \{0, 1\}$$
$$\sum_{i=1}^{N} \mathbf{x}_i w_i \leq B$$

v_i = utility of the data point e.g. uncertainity
w_i = labelling cost
B = Total budget

NP Hard so heuristics are needed

Batch Version:

$$\max_{\mathcal{X}} f_u(\mathcal{X})$$
$$\mathcal{X} \subset \mathcal{U}$$
$$f_c(\mathcal{X}) \leq B$$

$f_u(\mathcal{X})$ = Utility function operating over the batch
$f_c(\mathcal{X})$ = labelling cost function operating over the batch
B = Total budget

- What is an appropriate utility function? It is usually non-linear.
- What is the cost function? Cost function is sequential in nature i.e., cost of example x_i depends on the all the previously seen examples including the current example
- How to solve this optimization problem?

Cost Function

$$\max_{\mathcal{X}} \quad f_u(\mathcal{X})$$
$$\mathcal{X} \subset \mathcal{U}$$
$$f_c(\mathcal{X}) \leq B$$

Cost is dependent on the previously labeled examples.

$$C(\mathbf{x}_i) = C_0(\mathbf{x}_i) + C(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{x}_{i-2} \ldots)$$

Infinite Memory Assumption

$$C(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{x}_{i-2} \ldots) = \min_{j=1 \ldots i-1} \delta(\mathbf{x}_i, \mathbf{x}_j)$$

For a given permutation:

$$C(X_\tau) = \sum_{i=1}^{|X_\tau|} C_0(\mathbf{x}_i) + C(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{x}_{i-2} \ldots)$$

Optimal Ordering

$$f_c(\mathcal{X}) = \Delta^* \leftarrow \min_{\tau \in \mathcal{T}} C(X_\tau)$$

Proof

Theorem 3.1. *Let Δ_{MST} be the solution to the minimum spanning tree for a complete graph $G(\mathcal{X}, \delta(\mathbf{x}_i, \mathbf{x}_j)\, \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X})$, then Δ_{MST} is the solution to (5).*

$$f_c(\mathcal{X}) = \Delta^* \leftarrow \min_{\tau \in \mathcal{T}} C(X_\tau)$$

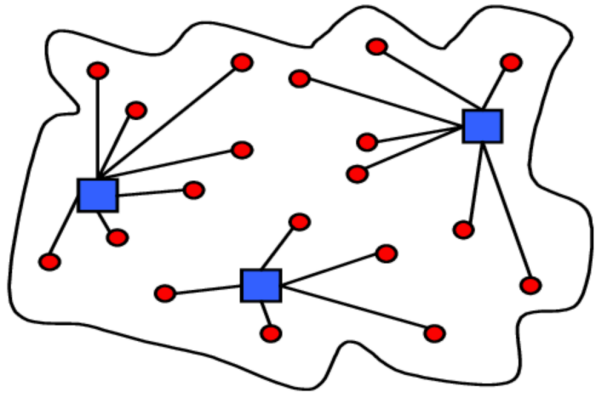
Proof. We prove this by contradiction. Lets assume that the solution to (5) is Δ^*, and $\Delta^* \neq \Delta_{MST}$. There are two possible scenarios: $\Delta^* > \Delta_{MST}$. Note that the minimum spanning tree algorithm gives us two outputs, cost of the tree and the tree itself. One can flatten out the tree and get an ordering of the points. Lets call that ordering τ_{MST}. Now let $\mathcal{T}$ be the set of all possible permutation for $\mathcal{X}$, then $\tau_{MST} \in \mathcal{T}$. Similar to the solution to the MST problem, let τ^* be the ordering from the optimal solution of Equation (5). Now by definition of the optimization, $C(X_{\tau^*}) < C(X_\tau) \forall \tau \in \mathcal{T}$ and since $\tau_{MST} \in \mathcal{T}$ implying $C(X_{\tau^*}) < C(X_{\tau_{MST}})$ or $\Delta^* < \Delta_{MST}$ which contradicts our hypothesis.

The second scenario is when $\Delta^* < \Delta_{MST}$. Let τ^* and τ_{MST} be the ordering corresponding to Δ^* and Δ_{MST} respectively, then our hypothesis states $C(X_{\tau^*}) < C(X_{\tau_{MST}}) \leq \Delta_{MST}$. As per the definition of the minimum spanning tree, all other spanning tree have a cost more than Δ_{MST}, and since $\tau*$ which includes all the nodes and edges from (3), is a spanning tree since it covers all the points and is tree by design so $\Delta_{MST} < C(X_{\tau^*})$ or $\Delta_{MST} < \Delta^*$ which is in contradiction. □

Minimum Spanning Tree

Utility Function

- The utility function should:
  - Give higher weight to individual samples that are important
  - Give higher weights to the region that is dense
  - Try to reduce the redundancy, or maximize the diversity, in order to cover the maximum region of the data space



- Facility Location Problem:
  - An optimization problem to select optimal set of facilities
  - Travelling distance is nothing but importance of each customer
  - It makes sense to have facility in dense regions
  - Want to select facilities which are farthest to each others (diversity)

- Applicability to Active Leaning:
  - Want to select facilities (or points to labels) which have the maximum utility or the value of the objective function.

$$f_u(\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_k) = \sum_{i=1}^{N} e_i \max_{\mathbf{x}_k} f_s(\mathbf{x}_k, \mathbf{x}_i)$$

How to Solve the Optimization Problem

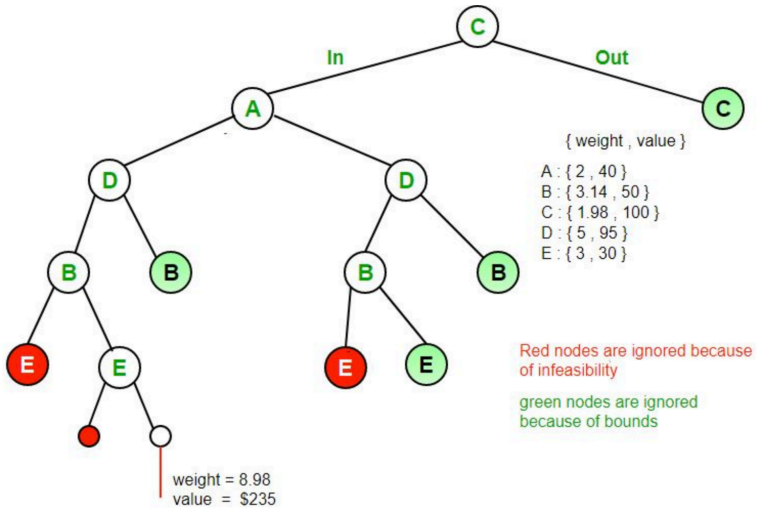
$$\max_{\mathcal{X}} \quad f_u(\mathcal{X})$$
$$\mathcal{X} \subset \mathcal{U}$$
$$f_c(\mathcal{X}) \leq B$$

- Solution to the non-batch version is known to be NP-hard, however a practical solution is available based on dynamic programming.

$$K_{i,b} = \max(v_{i-1} + K_{i-1,b-w_{i-1}}, K_{i-1,w})$$

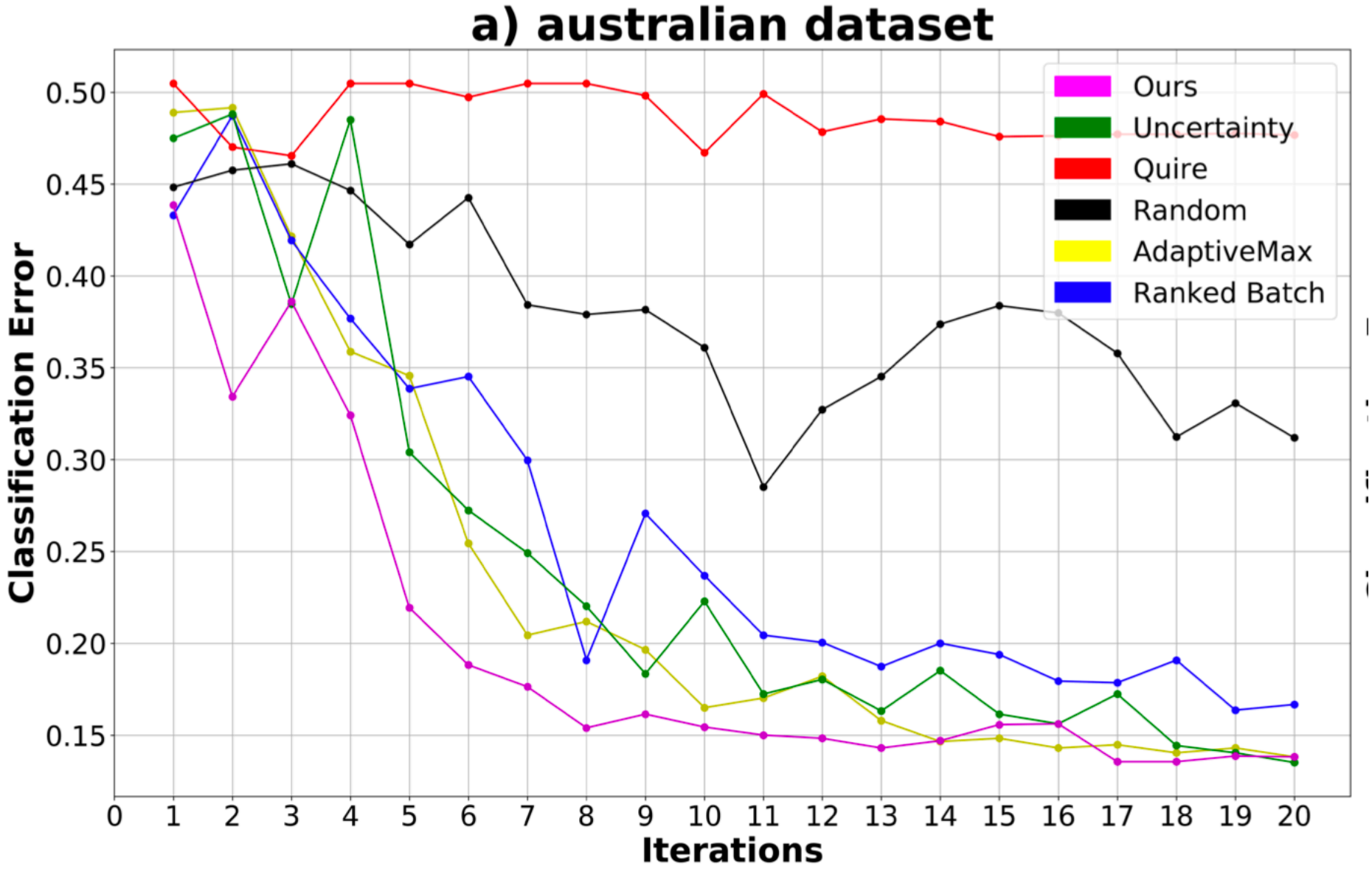
When the next point is included

When the next point NOT is included



- Brute force has the complexity of $O(2^n)$.
- DP has the complexity of $O(NB)$. Which is pseudo polynomial as it depends on the representation of B.

Representative Results (Australian Dataset)



Conclusion

- Proposed a novel and generic AL frame- work that selects the optimal batch within the budget constraint based on the given utility and cost functions.
- We also proposed a novel utility function based on the Facility Location problem.
- Proposed utility function has three important characteristics: (a) higher weights for important points, (b) higher weight for dense region, (c) Diversity of selected points.
- We also proposed a novel cost formulation, the optimal solution to which is the minimum spanning tree of the input batch.
- Experimental results on four datasets show that our approach outperforms the baseline algorithms especially in the initial iterations.