Common Active Learning Practices & Applications on Networked Data

De Liao, deliao2@illinois.edu
Qi Wang, qiwang12@illinois.edu
Doris Xin, dorxo@illinois.edu

Image courtesy of Andreas Krause
Why Active Learning?

• Labels are expensive to acquire in many classification tasks
  – E.g. medical experiments, lengthy documents with technical jargons.
• Unlabeled data is often readily available in large quantity
• Leverage the structure in the unlabeled data to minimize the number of labels required for small generalization error

“...an astonishing 8 in 10 [projects] were abandoned... Why such little success?... Clinical approval success rates have declined significantly.”
A Motivating Example

Figure 1: (a) A toy dataset of 400 instances evenly sampled from two class Gaussians. (b) A logistic regression model trained on a random subsample (of size 30) of the training set (accuracy = 0.7) (c) A logistic regression model trained on 30 actively queried datapoints via uncertainty sampling (accuracy = 0.9). [18]
Three Scenarios of Active Learning

• The learner may synthesize examples to query the oracle for labels based on domain knowledge of the learning task
• The learner may select examples to query from a stream
• The learner may examine a large pool of unlabeled examples and select a small subset for queries
Pool-based Active Learning

• **Full-sequential**: the learner selects a single example to query the oracle at each iteration

• **Batch-mode**: allows the learner to query for multiple examples in a given iteration

• **One-shot**: the learner is only allowed to query the oracle once (but with multiple instances if desired) at the beginning of the learning phase
Query Strategies

- Uncertainty Sampling
- Query-by-committee (QBC)
- Expectation Error Reduction
Query Strategies

- Uncertainty Sampling
- Query-by-committee (QBC)
- Expectation Error Reduction
Uncertainty Sampling

Select instances about which the learner is the least certain (LC). Given a label set Y

\[ x_{LC} = \arg \max_x 1 - P_\theta(\hat{y} | x), \text{ where } \hat{y} = \arg \max_{y \in Y} P_\theta(Y | x) \]

*Margin sampling*: difference between the top 2 most likely labels

\[ x_M = \arg \min_x P_\theta(\hat{y}_1 | x) - P_\theta(\hat{y}_2 | x) \]

*Shannon Entropy*:

\[ x_H = \arg \max_x - \sum_{y \in Y} P_\theta(y | x) \log P_\theta(y | x) \]
Query Strategies

- Uncertainty Sampling
- Query-by-committee (QBC)
- Expectation Error Reduction
Query-By-Committee (QBC)

- Committee of competing models (committee members)
- Select instances on which the members disagree the most
- Resolve instances in controversial regions in the *version space* (VS) in order to shrink the consistent version space with few examples
- Applications of QBC requires:
  - Constructing a committee covering different regions of the VS
  - *Disagreement* measure between committee members
Figure 3: Examples of version space for (a) linear classifier and (b) axis-parallel classifier in 2D. All hypotheses shown are consistent with the given dataset, i.e. in the version space, but different from each other.
Disagreement Measures

- **Vote Entropy**

\[
x_{VE} = \arg \max_x \frac{V(y)}{C} \sum_{y \in Y} \log \frac{V(y)}{C}
\]

Number of votes for label \(y\)

Size of committee

- **Average KL Divergence**

\[
x_{KL} = \arg \max_x \frac{1}{C} \sum_{i=1}^c \sum_{y \in Y} P_{\theta(i)}(y|x) \log \frac{P_{\theta(i)}(y|x)}{P_C(y|x)}
\]

\[
\frac{1}{C} \sum_{i=1}^c P_{\theta(i)}(y|x)
\]
QBC Example
Query Strategies

- Uncertainty Sampling
- Query-by-committee (QBC)
- Expectation Error Reduction
Expectation Error Reduction

• Instances that are the most likely to reduce generalization error

• Roy & McCallum ’01 pioneered the framework for text classification via naïve Bayes

• Guo & Schuurmans ’08 created an optimistic variant that biased the expectation towards the most likely label for computational convenience and used uncertainty sampling as a fall-back when the oracle returned a “surprising” label

• Can be generalized to reduce other measures beyond expected error, such as F1 and AUC.
Active Learning on Networked Data

Active Learning for Networked Data [Bilgic et al ‘10]

Active Learning for Node Classification in Assortative and Disassortative Networks [Moore et al ‘07]

Batch Mode Active Learning for Networked Data [Shi et al ‘12]
Active Learning on Networked Data

Active Learning for Networked Data [Bilgic et al ‘10]

Active Learning for Node Classification in Assortative and Disassortative Networks [Moore et al ‘07]

Batch Mode Active Learning for Networked Data [Shi et al ‘12]
Active Learning for Networked Data

• Clusters nodes via off-the-shelf graph clustering algorithms
• CC: collective classifier, which classifies the node based on its feature vector and labels for neighbors in aggregate
• CO: content-only classifier, which classifies the nodes based only on its feature vector
• Request labels for nodes where CC and CO don’t agree or make predictions that don’t match the observed label distribution in small batches ($k$)
Example CC & CO

numNbrsFeat1=a: 3
numNbrsFeat1=c: 0
numNbrsFeat2=b: 3
Disagreement Measure

\[
\text{Disagreement}(\mathcal{C}_j, \mathcal{L}) = \sum_{V_i \in \mathcal{C}_j \cap \mathcal{P}} \text{LD}(\mathcal{C}_j, \mathcal{L})
\]

**Assign** \(C_j\) with the majority class in the intersection of \(C_j\) and \(\mathcal{L}\)

**Set of labeled nodes**

**Pool of unlabeled nodes**

\[
\text{LD}(\mathcal{C}, \mathcal{L}) = H_{\mathcal{D}_i}(V_i)
\]

\[
\mathcal{D}_i = \{p_{i}^{h} \mid h \in \mathcal{S}_i\}
\]

**Fraction of classifiers that predicted** \(h\)

**Set of all categories predicted by all 3* classifiers**
Algorithm 1: ALFNET: Active Learning for Networked Data

**Input:** \( G = (\mathcal{V}, \mathcal{E}) \): the network, \( \mathcal{C}_O \): content-only learner, \( \mathcal{C}_C \): collective learner, \( k \): the batch size, \( B \): the budget

**Output:** \( \mathcal{L} \): the training set

1. \( \mathcal{L} \leftarrow \emptyset \)
2. \( \mathcal{C} \leftarrow \) Cluster the nodes \( \mathcal{V} \) of the network \( G \) into at least \( k \) clusters
3. \( \mathcal{C}^k \leftarrow \) Pick \( k \) clusters from \( \mathcal{C} \)
4. **foreach** Cluster \( \mathcal{C}_i \in \mathcal{C}^k \)
   5. \( V_j \leftarrow \) Pick an item from \( \mathcal{C}_i \)
   6. Add \( V_j \) to \( \mathcal{L} \)

while \( |\mathcal{L}| < B \)

8. Re-train \( \mathcal{C}_O \) and \( \mathcal{C}_C \)
9. **foreach** Cluster \( \mathcal{C}_i \in \mathcal{C} \)
   10. \( \text{score}(\mathcal{C}_i) \leftarrow \text{Disagreement}(\mathcal{C}_C, \mathcal{C}_O, \mathcal{C}_i, \mathcal{L}) \)
11. \( \mathcal{C}^k \leftarrow \) Pick \( k \) clusters based on the scores
12. **foreach** Cluster \( \mathcal{C}_i \in \mathcal{C}^k \)
13. \( V_j \leftarrow \) Pick an item from \( \mathcal{C}_i \cap \mathcal{P} \)
14. Add \( V_j \) to \( \mathcal{L} \)
15. Remove \( V_j \) from \( \mathcal{P} \)

Start with \( k \) clusters with one labeled node in each cluster

Pick an unlabeled node to query from each of the top \( k \) most uncertainty clusters
Experimental Results

• Used Cora and CiteSeer. Node represented by a binary word appearance vector (vocabulary set really small ~$10^3$) and citation links. Pruned away nodes that aren’t in the main connected component.

• Combined dimensionality reduction on the nodes
Comments
(from the POV of an adversarial reviewer)

• Homogeneous nodes
• Depends heavily on the performance of the chosen graph clustering algorithm
• Results do not show significant improvement
• Datasets are small. Would be interesting to see how the algorithm scales since it requires multiple iterations of training on the whole graph
Active Learning on Networked Data

Active Learning for Networked Data [Bilgic et al ‘10]

Active Learning for Node Classification in Assortative and Disassortative Networks [Moore et al ‘07]

Batch Mode Active Learning for Networked Data [Shi et al ‘12]
Active Learning for Node Classification in Assortative and Disassortative Networks

- Discovery of functional communities, i.e. a set of nodes that connects to the rest of the network in a similar way
  - Simply being connected to each other does not necessarily put two nodes into the same *functional* community
- Select nodes by estimating the mutual information (MI) between their labels and the join distribution of all other nodes via Gibbs Sampling
- AA: average number of nodes at which two independent samples of the Gibbs distribution agree

Pool-based  Uncertainty Sampling  Full-sequential
Mutual Information on Graphs

\[ \text{MI}(v) = I(v; G \setminus v) = H(G \setminus v) - H(G \setminus v \mid v) \]

\[ \text{MI}(v) = I(v; G \setminus v) = H(v) - H(v \mid G \setminus v) \]

\[ I(v; G \setminus v) = - \sum_{i=1}^{k} \langle P_i \rangle \ln \langle P_i \rangle + \left\langle \sum_{i=1}^{k} P_i \ln P_i \right\rangle \]

- Symmetry of MI
- Probability that node v has label i
- \(<*>: \text{average over the labels of other other nodes}\)
An Example
Experimental Results

- 3 datasets: social network of a karate club, common adjacent words in a Dickens novel, and a marine food web
- Compared with random, betweenness, degree
Comments
(from the POV of an adversarial reviewer)

• How MI alone captures intuitive exploration strategies on graphs is fascinating. Would’ve been nice to see explicit analysis on why this happens
• MI calculation is very expensive. $O(n^2)$ on every iteration
• Very small datasets, probably because of the lack of scalability
• Interesting to see the much simpler (and more scalable) algorithm proposed performs almost just as well
Active Learning on Networked Data

- Active Learning for Networked Data [Bilgic et al ‘10]
- Active Learning for Node Classification in Assortative and Disassortative Networks [Moore et al ‘07]
- Batch Mode Active Learning for Networked Data [Shi et al ‘12]
Batch Mode Active Learning for Networked Data*

*Fairly complex paper. We’ll present a select set of key insights gleaned from the paper

- Three criteria to maximize the informativeness of the set of nodes selected for querying
  - maximum uncertainty
  - maximum impact
  - minimum redundancy
- Objective function: linear combination of maximum uncertainty and maximum impact
- Random walk to compute expected labels for unlabeled datapoints
- Redundancy minimization was proven to be fully captured by maximum uncertainty and maximum impact

Pool-based  Uncertainty Sampling  Batch Mode
Objective Function

\[ Q(S) = \alpha C(S) + (1 - \alpha)H(S), \quad 0 \leq \alpha \leq 1 \]

**Uncertainty:**

\[ H(S) = \sum_{i \in S} H(i) = \sum_{i \in S} f_i \log \frac{1}{f_i} + (1 - f_i) \log \frac{1}{1 - f_i} \]

**Impact:**

\[ C(S) = \sum_{i \in U} (H(i))^\beta \left( \max_{j \in L \cup S} w_{ij} \right)^{1-\beta} \]

- Expectation of label for node \( i \) from random walk
- Similarity between node \( i \) and \( j \)
Experimental Results

• A parallelized version of the algorithm for scalability
• Synthetic Gaussian; real network datasets: Cora, CiteSeer, & WebKB
• Compared with Most Uncertainty(MU), Gaussian Fields (GF), Hybrid, k-means (K-M)

(a) accuracy in Cora
(b) accuracy in CiteSeer
(c) accuracy in WebKB
Comments

• Interesting that they were able to prove redundancy minimization is equivalent to maximizing a linear combination of uncertainty and impact of the set

• Multiple stages (random walk first, then uncertainty&impact maximization, then link information integration

• Both synthetic and real datasets

• Parallelization