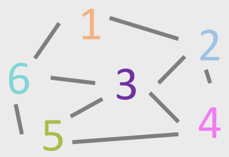
Active Learning with Label Comparisons



Gal Yona, Shay Moran, Gal Elidan, Amir Globerson (Google Research)

Label Comparisons

Pairwise label-comparison: given **x** and two candidate classes y1, y2, "is **x** more y1 than y2"?

$$A^f(\boldsymbol{x}, y_1, y_2) = \mathbf{1}[f_{y_1}(\boldsymbol{x}) > f_{y_2}(\boldsymbol{x})]$$

cf. **argmax supervision**:

$$A^f(\boldsymbol{x}) = \arg\max_i f_{y_i}(\boldsymbol{x})$$

Many cases where label-comparisons are natural & cognitively easier to provide.

Recent work (OpenAI) leveraged them to effectively align LLMs with user intent – but they used tens of thousands such comparisons, collected *heuristically*.

Fundamental question: Which comparisons to request for maximally useful supervision with minimal annotation cost?

This work: A theoretical perspective on this question.

Passive Learning

PAC learning, every example **x** is labeled with (i) argmax (ii) all $\binom{k}{2}$ label-comparisons (i.e. total order on classes).

Does this extra information make learning easier?

Negative result: In general, label-comparisons may not be helpful for passive learning.

Theorem: Learning linear classifier in $d = 1$ requires $\Omega(k/\epsilon)$ samples with *both forms of supervision*.

Active Learning (AL)

Learner draws unlabeled samples; decides which queries to ask the oracle. Performance is measured in terms of query complexity.

Basic observation: any AL that uses argmax queries can be simulated using comparisons.

Thus, we consider comparisons as helpful if the query complexity required to learn a class H is *strictly lower* than the query complexity required to simulate the best AL that uses argmax queries.

Theorem: Label-comparisons are helpful for actively learning linear classifier in $d = 1$!

Naïve (multi-class AL of H by simulating the best argmax AL algorithm)

- Learn Θ_2 with binary AL on the problem {1} vs {2,3} $\Rightarrow O(k \cdot \log 1/\epsilon)$ argmax q's
- Learn Θ_3 with binary AL on the problem {1,2} vs {3} $\Rightarrow O(k^2 \cdot \log 1/\epsilon)$ comparison q's

Smart (multi-class AL of H with a tailored algorithm)

Observation: Had we known the *order* of the classes, $O(k \cdot \log 1/\epsilon)$ comparisons suffice! (e.g. learn Θ_2 with binary AL on the problem {1} vs {2} using comparisons!)

And we can actually learn the order with $2k$ comparisons.

Practical algorithms

- **Label neighborhood graph**: y1, y2 neighboring if they share a decision boundary.
- In general, comparisons are useful when G^* is both **sparse** and can be learned with relatively few comparisons (e.g. H from before).
- We derive a practical algorithm, NbrGraphSGD, that uses a a graph G to guide requested comparisons and model updates.
- Experiments on synthetic & real data demonstrate the label neighborhood graph indeed plays an important role in AL efficacy, and NbrGraphSGD uses this structure.

Input: Label neighborhood graph G, buffer size R, steps T, confidence parameter τ, learning rate η, comparison oracle A^{f^*}.
Output: classifier $h(\cdot; \boldsymbol{W})$, number of comparisons q.

Initialize $\boldsymbol{W}^{(0)}$, $L = 0, q = 0, b = 0$.
for $t = 1, 2, \ldots, T$ **do**
  Sample $\boldsymbol{x} \sim \mathcal{D}$.
  Sample (i, j) uniformly from the edges of G.
  if $\left|h_i(\boldsymbol{x}; \boldsymbol{W}^{(t-1)}) - h_j(\boldsymbol{x}; \boldsymbol{W}^{(t-1)})\right| < \tau$ **then**
    Obtain oracle comparison $c = 2(A^{f^*}(\boldsymbol{x}, i, j) - 0.5)$
    $L \mathrel{+}= \log(1 + e^{-c(h_i(\boldsymbol{x};\boldsymbol{W}) - h_j(\boldsymbol{x};\boldsymbol{W}))})$.
    $q \mathrel{+}= 1, b \mathrel{+}= 1$.
  end if
  if $b \geq r$ **then**
    Update $\boldsymbol{W}^{(t)} \leftarrow \boldsymbol{W}^{(t-1)} - \eta \cdot \frac{\partial L}{\partial \boldsymbol{W}}$
    Clear buffer: $L = 0, b = 0$.
  end if
end for

Active Learning (AL)

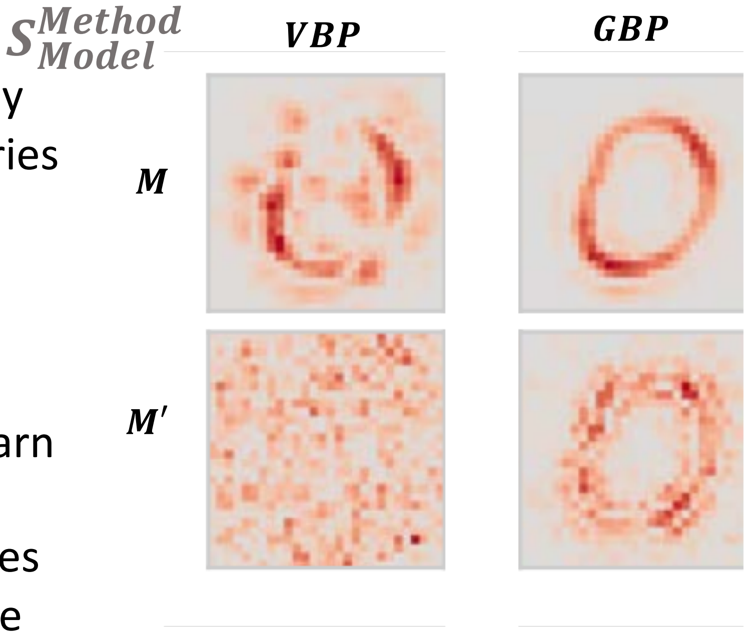
The learner draws unlabeled samples, and decides which queries to ask the oracle for (inc. no queries). Performance is measured in terms of query complexity.

Basic observation: any AL algo that uses argmax queries can be simulated using comparisons ($k-1$ label-comparisons suffice).

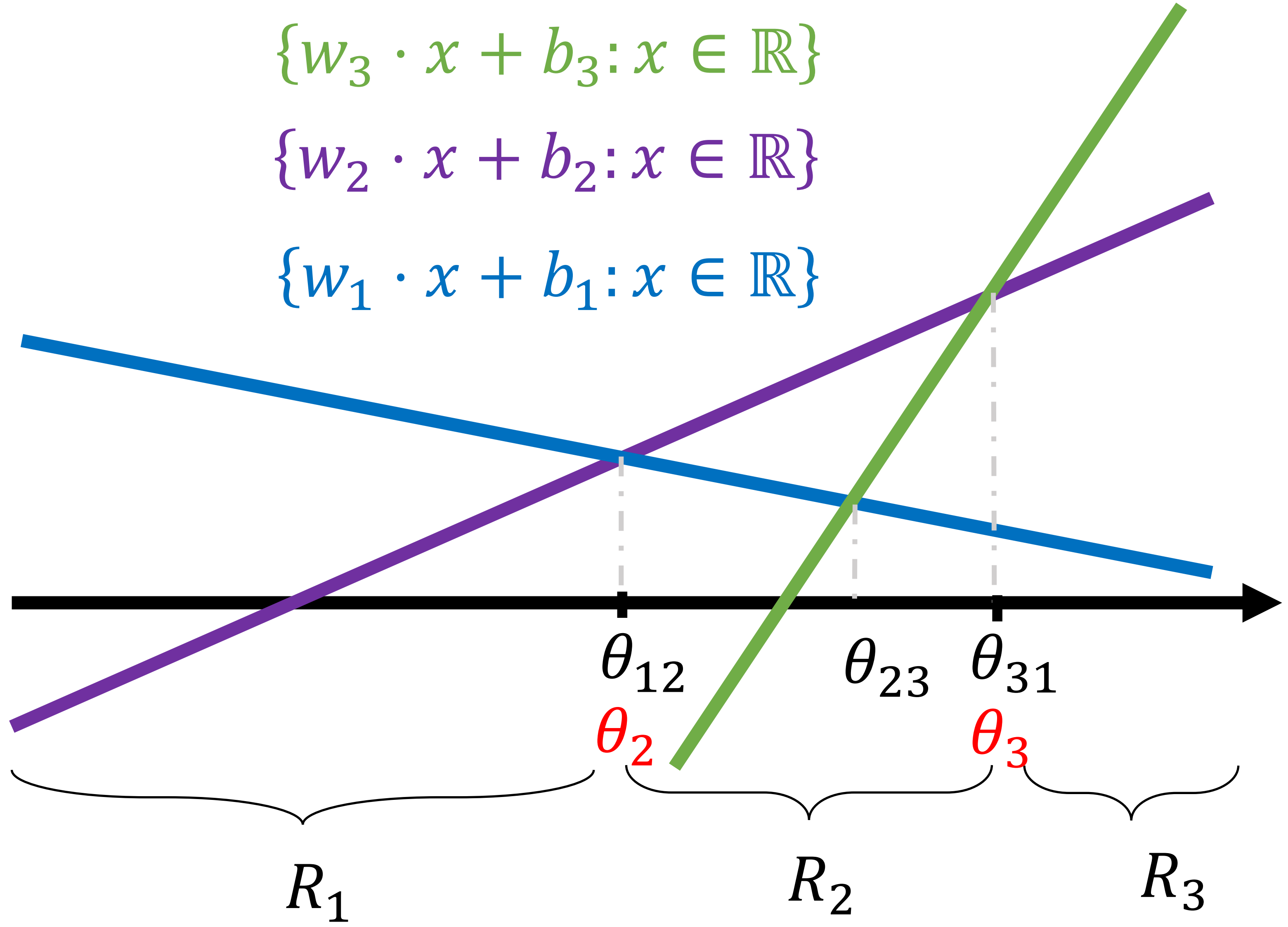
Thus, we consider comparisons as helpful if the query complexity required to learn a class H is *strictly lower* than the query complexity required to simulate the best AL that uses argmax queries.

Theorem: Label-comparisons are helpful for actively learning linear classifier in $d=1$!

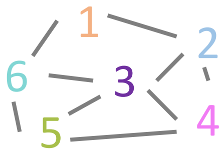
This provides a generic way to use the label-comparison oracle: simply request the label-comparison queries necessary for a “regular” active learner. We therefore say that comparisons are useful for active learning if the number of label-comparison queries required to learn a class H is strictly lower than the number of label-comparison queries required to simulate the best active learner that uses argmax queries to learn H.

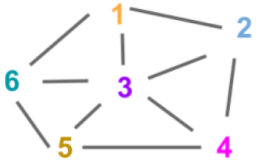


Analogy: Edge detection



$\{w_3 \cdot x + b_3 : x \in \mathbb{R}\}$
$\{w_2 \cdot x + b_2 : x \in \mathbb{R}\}$
$\{w_1 \cdot x + b_1 : x \in \mathbb{R}\}$
θ_{12}
θ_{23}
θ_{31}
θ_2
θ_3
R_1
R_2
R_3





- Learn Θ_2 with binary AL on the problem {1} vs {2,3} $\Rightarrow O(k \cdot \log 1/\epsilon)$ argmax queries
- Learn Θ_3 with binary AL on the problem {1,2} vs {3} $\Rightarrow O(k^2 \cdot \log 1/\epsilon)$ comparison queries

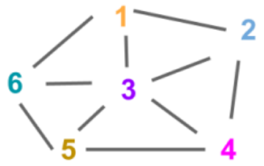
t label-comparisons will be useful when (i) the target neighborhood graph is sparse (has low degree), and (ii) it can be learned with relatively few labelcomparisons.

graph plays an important role in active learning efficacy, and that NbrGraphSGD can use this structure.

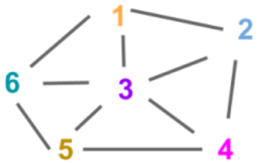
We use this to derive a practical algorithm, NbrGraphSGD:

Experiments on synthetic & real data demonstrate

- **Label neighborhood graph**: classes y1, y2 are neighboring if they share a decision boundary.
- In general, comparisons will be useful when the target label neighborhood is both **sparse** and can be learned with relatively few comparisons (e.g. H from before).
- We derive a practical algorithm, NbrGraphSGD, that uses a label neighborhood graph to guide the comparisons to request and update the model.
- Experiments on synthetic & real data demonstrate the label neighborhood graph indeed plays an important role in active learning efficacy, and NbrGraphSGD uses this structure.



t label-comparisons will be useful when (i) the target neighborhood graph is sparse (has low degree), and (ii) it can be learned with relatively few labelcomparisons.

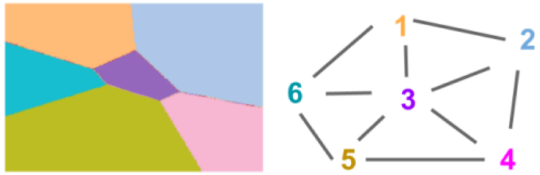


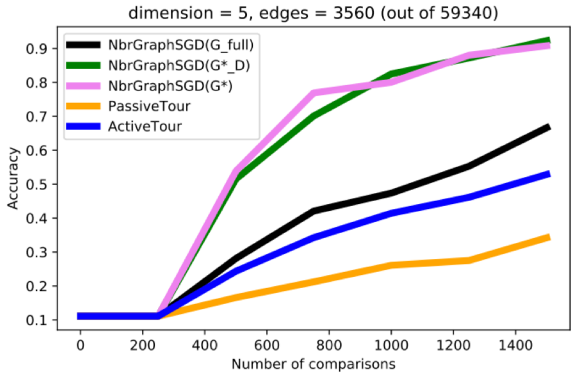
We use this to derive a practical algorithm, NbrGraphSGD:

Input: Label neighborhood graph G, buffer size R, steps T, confidence parameter τ, learning rate η, comparison oracle A^{f^*}.
Output: classifier $h(\cdot; \boldsymbol{W})$, number of comparisons q.

Initialize $\boldsymbol{W}^{(0)}$, $L = 0, q = 0, b = 0$.
for $t = 1, 2, \ldots, T$ **do**
  Sample $\boldsymbol{x} \sim \mathcal{D}$.
  Sample (i, j) uniformly from the edges of G.
  if $\left|h_i(\boldsymbol{x}; \boldsymbol{W}^{(t-1)}) - h_j(\boldsymbol{x}; \boldsymbol{W}^{(t-1)})\right| < \tau$ **then**
    Obtain oracle comparison $c = 2(A^{f^*}(\boldsymbol{x}, i, j) - 0.5)$
    $L \mathrel{+}= \log(1 + e^{-c(h_i(\boldsymbol{x};\boldsymbol{W}) - h_j(\boldsymbol{x};\boldsymbol{W}))})$.
    $q \mathrel{+}= 1$, $b \mathrel{+}= 1$.
  end if
  if $b \geq r$ **then**
    Update $\boldsymbol{W}^{(t)} \leftarrow \boldsymbol{W}^{(t-1)} - \eta \cdot \frac{\partial L}{\partial \boldsymbol{W}}$
    Clear buffer: $L = 0$, $b = 0$.
  end if
end for

Experiments on synthetic & real data demonstrate

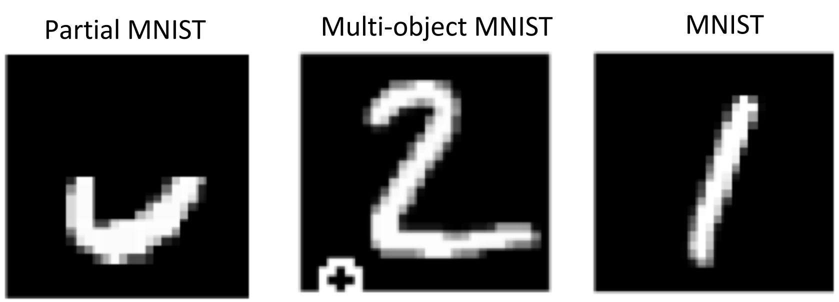


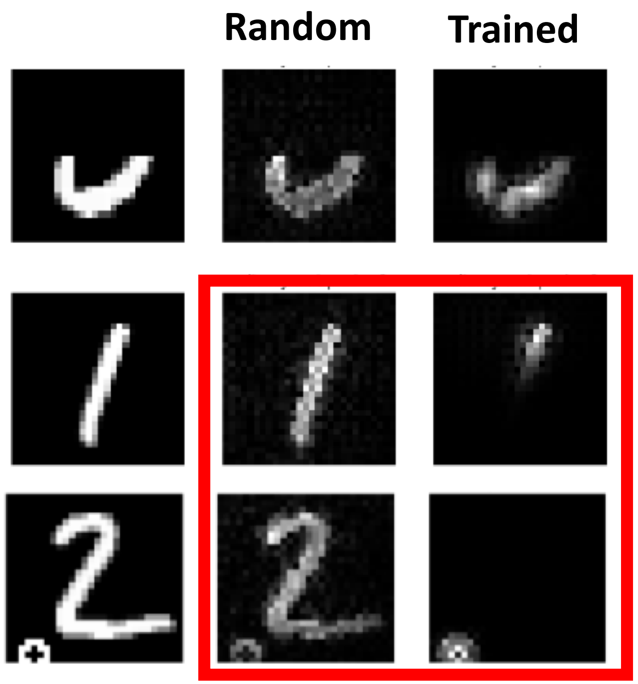


This Work

We design **custom tasks** intended to explicitly control for these (potentially) confounding factors

We show: saliency maps w.r.t the random model are **clearly different** than maps w.r.t the trained model, *for both VBP and GBP*





Discussion

Challenges current “wisdom” regarding saliency methods

- Is GBP really “worse” than VBP?
- Sanity check methodology not as useful in distinguishing between different methods

Moving forwards: comparing different methods beyond ad-hoc visual examination remains challenging!

Need proper benchmarks: Can semi synthetic datasets help?