Learning more from end-users and teachers

Oregon State University AI and EUSES Groups

Tom Dietterich

on behalf of

Alan Fern, Kshitij Judah, Saikat Roy, Joe Selman

Weng-Keen Wong, Ian Oberst, Shubumoy Das, Travis Moore, Simone Stumpf, Kevin McIntosh, Margaret Burnett

Research Space

	Supervised Learning	Imitation Learning	Reinforcement Learning
Current Methods	Label feedback	Demonstrations	Demonstrations
	Active learning for labels	Active learning via online action feedback	Active learning via online action feedback
	Equivalence queries and Membership Queries		
Novel Methods	Feature Labeling by end users [IUI 2011]	State Queries with $\perp$ responses [ICML Workshop 2010]	Practice & Critiques [AAAI 2010]
	Object Queries and Pairing Queries [ECML 2011]		

Label Feedback from End Users

- Setting:
  - Document classification (multi-class)
    - Features are words, n-grams, etc.
  - End user labels *features* as positive or negative for a class
  - Small data set; user-specific classes

Related Work

Supervised feature labeling algorithms:

1. SVM Method 1 [Raghavan and Allan 2007]
   - Scales relevant features by $\boldsymbol{a}$
   - Scales non-relevant features by $\boldsymbol{d}$
   - Where $a \geq d$
2. SVM Method 2 [Raghavan and Allan 2007]
   - Inserts pseudo-documents into the dataset
     pseudo-document: (0, 0, ..., r, ..., 0, class label)
   - Influences position of margin

Combined method will be called SVM-M1M2

Idea: Combine local learning algorithm with feature weights

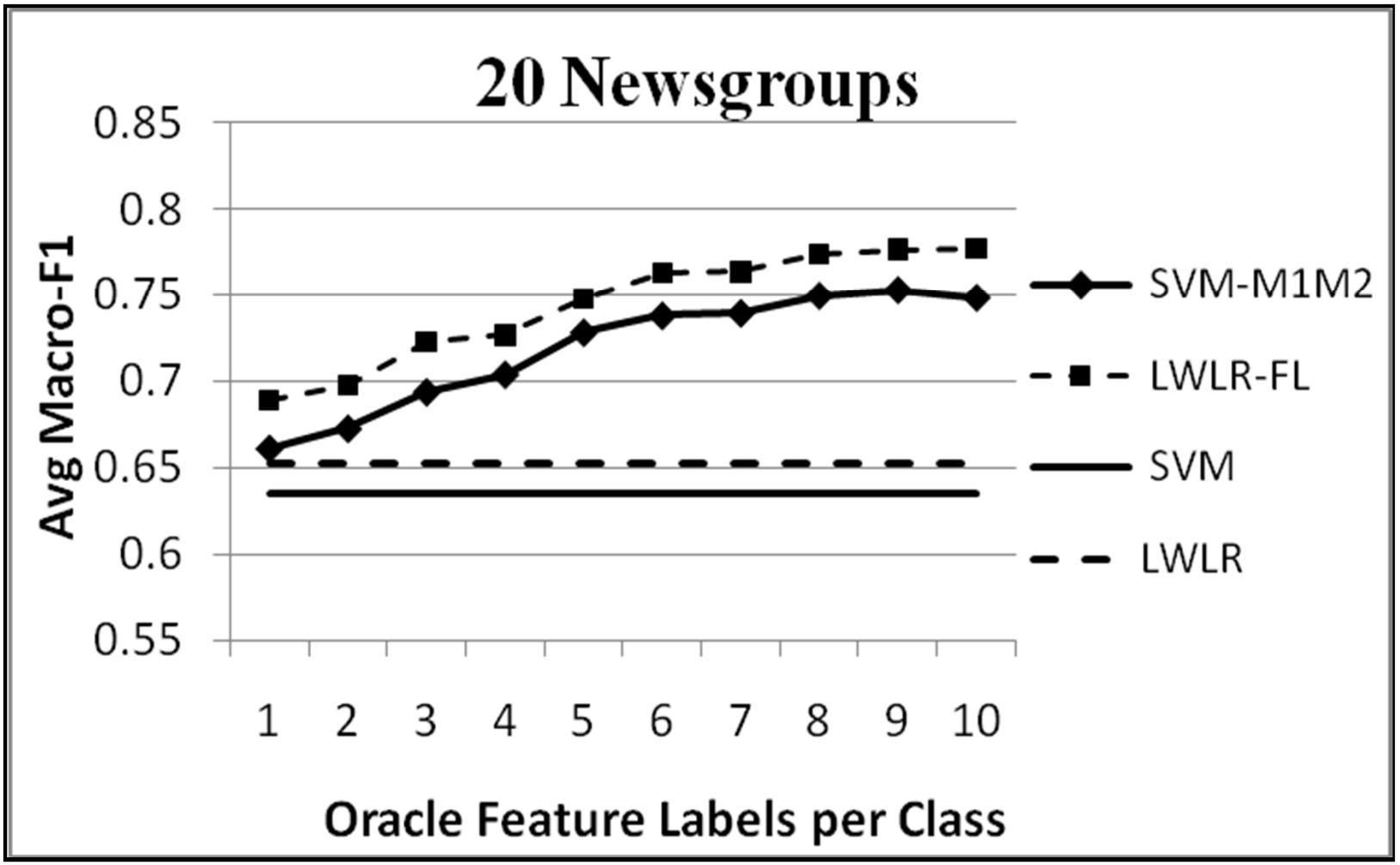
- Algorithm:
  - Locally-weighted logistic regression
  - Given query x_q assign weight $w_i = sim(x_q, x_i)$ to each training example x_i
  - Fit logistic regression to maximize weighted log likelihood
- Incorporating feature labels:
  - When training classifier for class k, if x_q and x_i share a feature labeled as positive for class k then make them "more similar"
  - If they share a feature labeled as positive for some other class, then make them "less similar"
- Hypothesis:
  - Local learning will prevent feature weights from over-generalizing beyond the local neighborhood

Experiments: Oracle Study

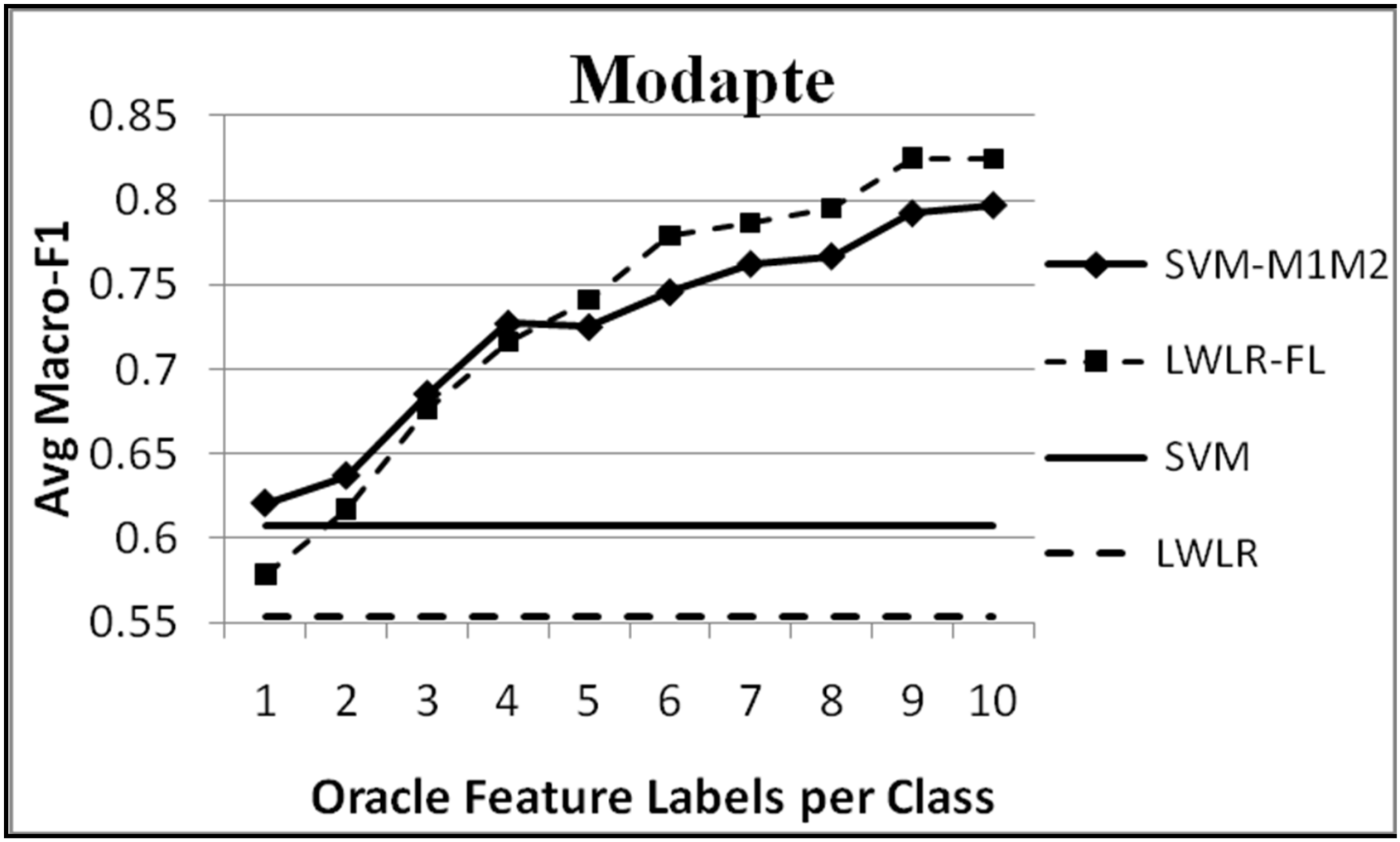
Oracle study: What happens if you can pick the "best" feature labels possible?

- Datasets
  - Balanced subset of 20 Newsgroups (4 classes)
  - Balanced subset of Modapte (4 classes)
  - Balanced subset of RCV1 (5 classes)
- Oracle feature labels:
  - 10 most informative features for each class (information gain computed over entire dataset)

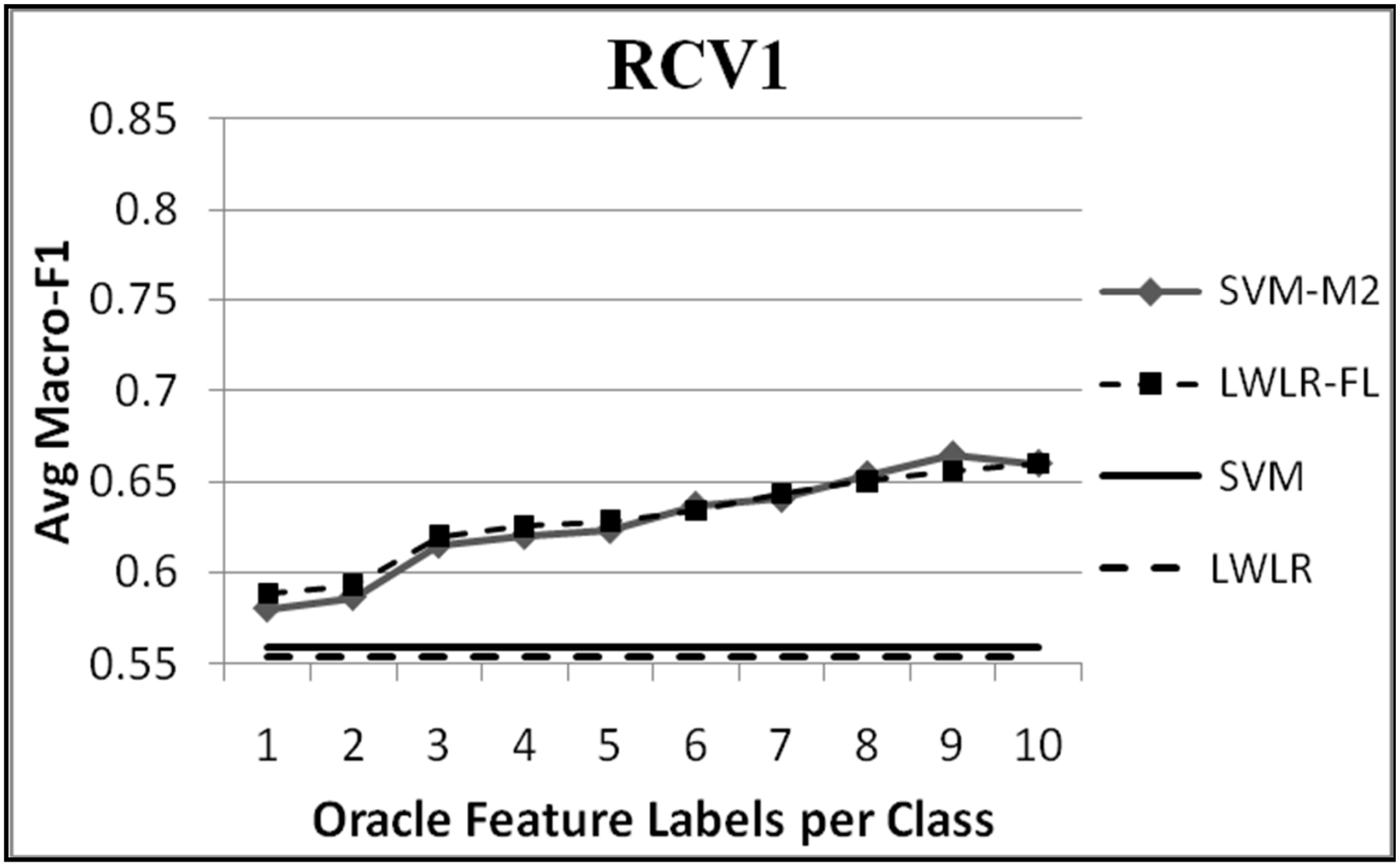
Results: Oracle Study



Results: Oracle Study



Results: Oracle Study



Results: Oracle Study

Summary

With oracle feature labels, LWLR-FL outperforms or matches the performance of SVM variants

Experiment: User Study

But what about real end users?

- How good are their feature labels?
- First user study of its kind:

  Statistical user study allowing end users to label any features

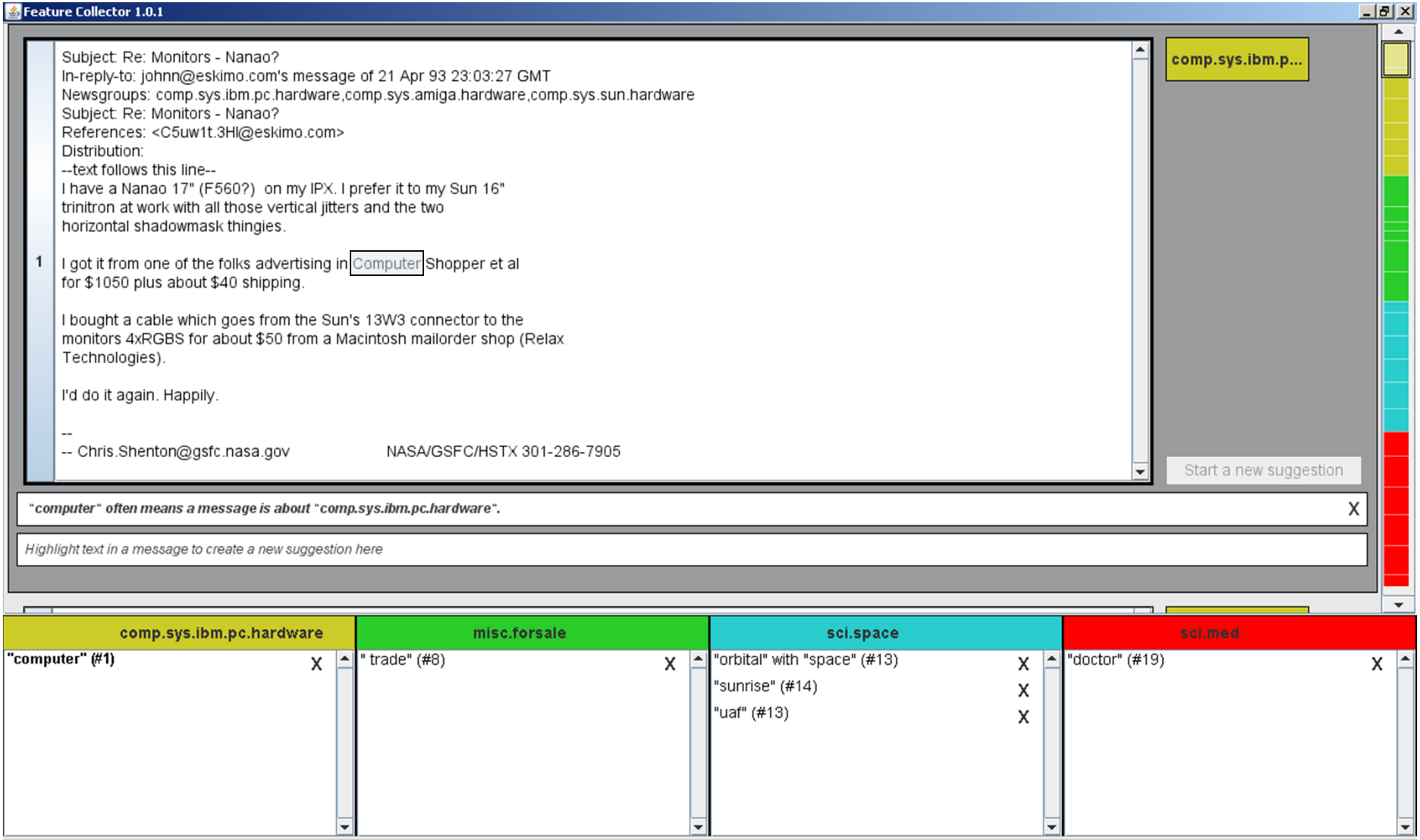
Results: User Study

- Presented 24 news articles from 4 Newsgroups: Computers, For Sale, Medicine, Outer Space
- Collected feature labels from 43 participants:
  - 24 male, 19 female
  - Non-CS background
- Experimental Setup
  - Features are unigrams
  - Training set: 24 instances
  - Validation set: 24 instances
  - Test set: remainder of data

User Study: Open-Ended Feature Set

- Participants allowed to highlight any text (including words and punctuation) that they thought was predictive of the newsgroup
- Separate results into two groups:
  - Existing: feature labels only on unigrams
  - All: feature labels on unigrams and any additional features highlighted by end users

Results: User study

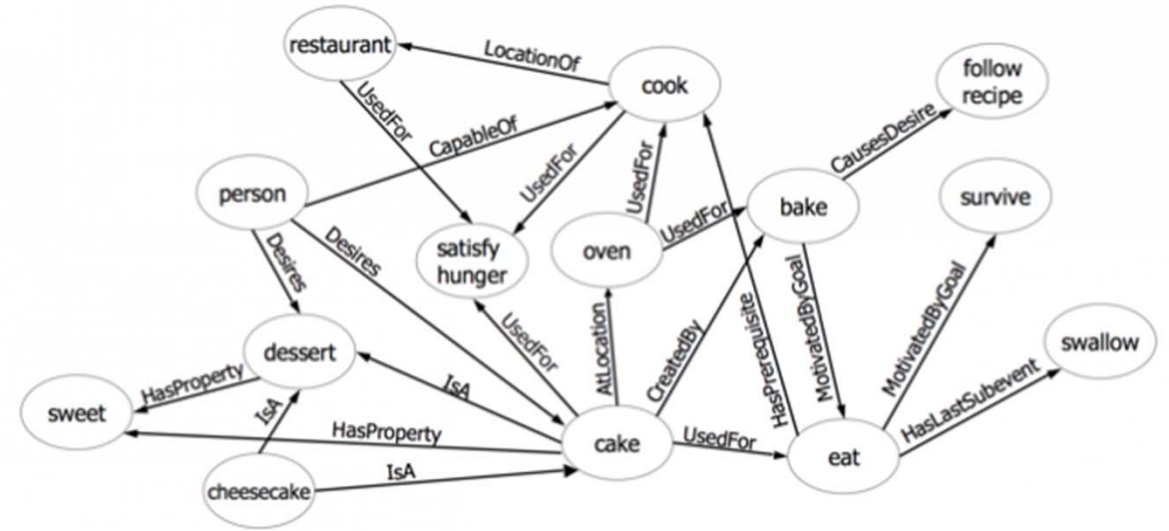


Results: User Study

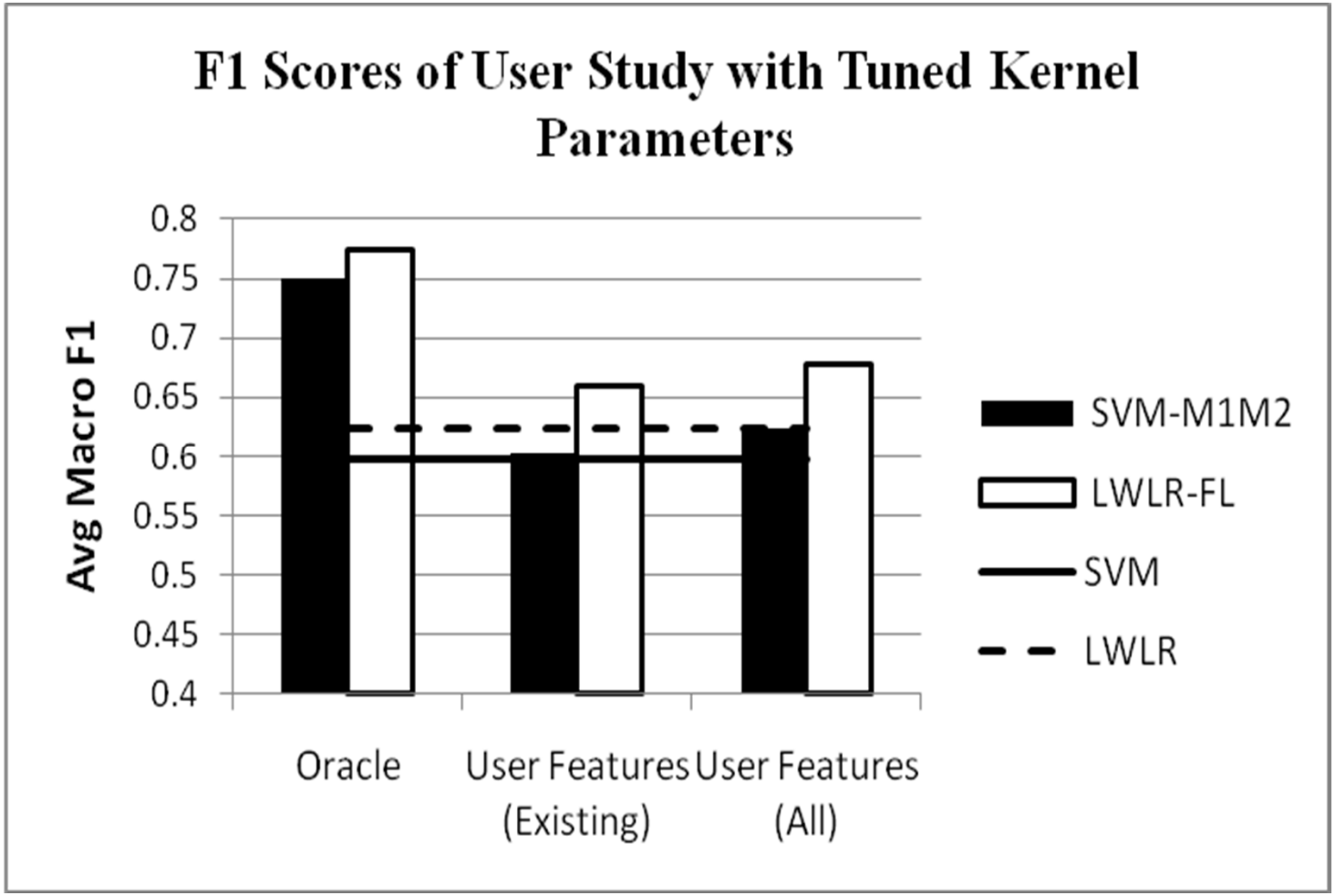
- End users introduced
  - non-continuous words ("cold" with "flu")
  - continuous phrases ("space shuttle")
  - features with punctuation ("for sale" with "$")
- Analysis of participants' features vs the oracle:
  - Lower average information gain (0.035 vs 0.078)
  - Higher average ConceptNet relatedness (0.308 vs 0.231)

Results: User Study

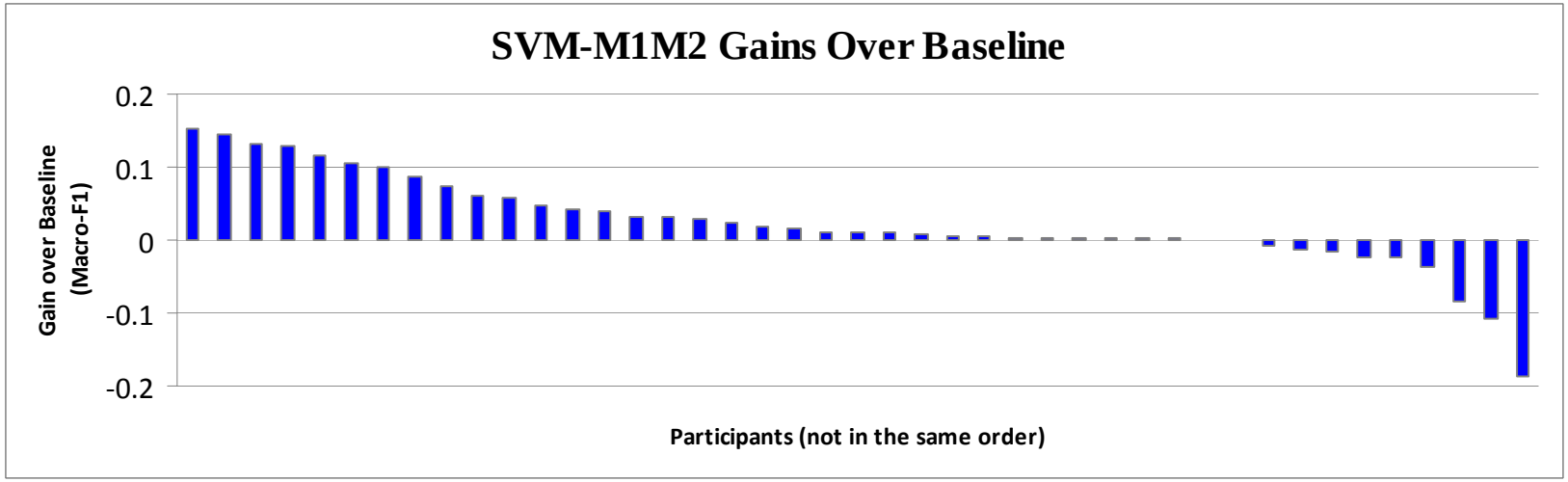
- Looked at relatedness from ConceptNet as an alternative to information gain
- End users picked features with higher average relatedness than oracle

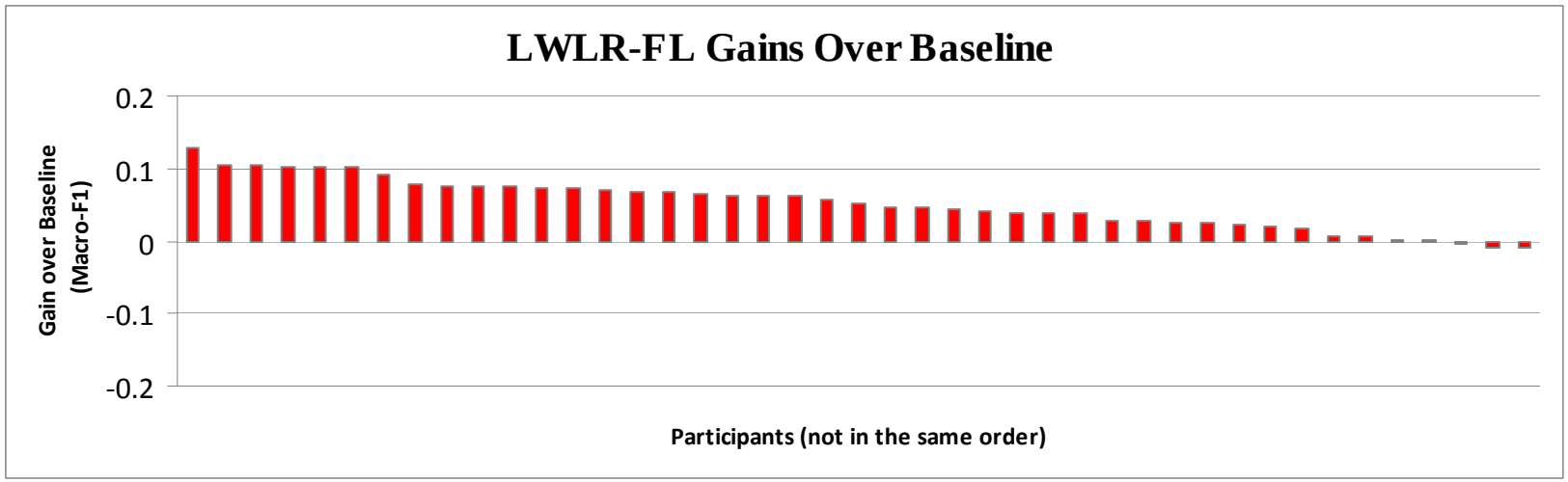


Results: User Study



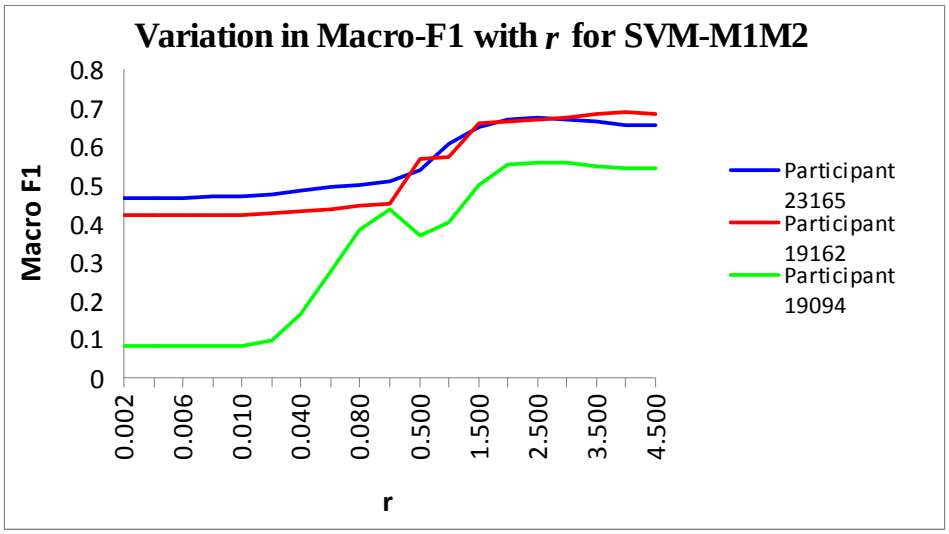
Results: User Study

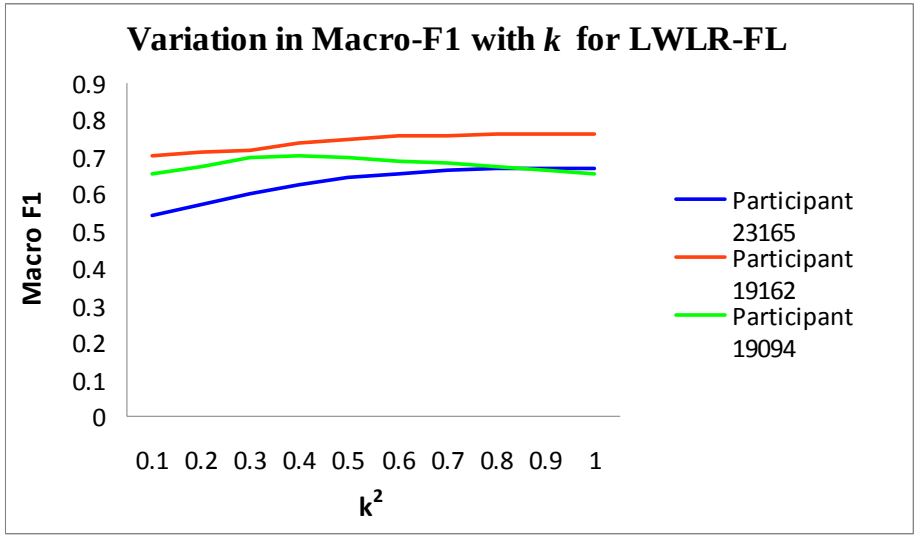




Results: User Study

Sensitivity Analysis





LWLR-FL is less sensitive to changes in key parameter

Results: User Study

Summary

- With real end-user feature labels, LWLR-FL outperforms SVM variants
- LWLR-FL is more robust to lower quality feature labels
- End users able to select features that have high relatedness to class label

Research Space

	Supervised Learning	Imitation Learning	Reinforcement Learning
Current Methods	Label feedback	Demonstrations	Demonstrations
	Active learning for labels	Active learning via online action feedback	Active learning via online action feedback
	Equivalence queries and Membership Queries		
Novel Methods	Feature Labeling by end users [IUI 2011]	State Queries with $\perp$ responses [ICML Workshop 2010]	Practice & Critiques [AAAI 2010]
	Object Queries and Pairing Queries [ECML 2011]		

Learning First-Order Theories using Object-Based Queries

- Goal:
  - Learn a first-order Horn theory
    - Set of Horn clauses
      - No functions
      - No constants (only variables)
  - A Horn theory covers a training example if it D-subsumes the example
    - Subsumption is required to be a one-to-one mapping
    - For example:
      - Theory: $P(X,Y), P(Y,Z) \Rightarrow Q(X,Z)$
      - D-subsumes $P(1,2), P(2,3) \Rightarrow Q(1,3)$
      - Does not D-subsume $P(a,b), P(b,b) \Rightarrow Q(a,b)$
    - Every theory under normal semantics has an equivalent theory that uses the new semantics

Previous Work

- Angluin et al. 1992:
  - Propositional Horn theories can be learned in polynomial time using Equivalence Queries and Membership Queries
  - Equivalence Query (EQ):
    - Ask teacher if theory T is equivalent to the correct theory
      - If No, returns a counter-example
  - Membership Query (MQ):
    - Ask teacher if example X is a positive example of the correct theory
- Reddy & Tadepalli, 1997:
  - Non-recursive function free first-order Horn definitions (single target predicate) can be learned in polynomial time using EQs and MQs
- Khardon, 1999
  - General first-order Horn theories can be learned in polynomial time using EQs and MQs (for fixed max size)

Shortcoming: MQs and EQs are unrealistic

- All of the algorithms make heavy use of MQs
- This can be unnatural for humans to answer
  - Teacher effort of labeling can be especially high
  - Often the examples asked about are created by the algorithm, and may not make sense in the real world
  - Each query only conveys a small amount of information

New Queries

- ROQ: Relevant Object Query
  - Given a positive example E, returns a minimal set of objects Q such that there exists a clause C in the true theory and a D-substitution Θ such that $C\Theta \subseteq E$

- Example for target concept $uncle(U, A)$
  - E: $father(a, b), father(a, c), spouse(a, d), brother(e, a), father(e, f), father(e, g), uncle(e, b), uncle(e, c), uncle(a, f), uncle(a, g), aunt(d, f), aunt(d, g)$
  - Q: $\{a, b, e\}$
- Clause: $uncle(U, A) \Leftarrow father(F, A), brother(U, F)$

New Queries

- PQ: Pairing Query
  - Given two positive examples E_1 and E_2, returns $false$ if there is no clause C in the true theory that covers both of them. Otherwise, it picks a clause C that covers both of them and returns a $1-1$ mapping of the objects in E_1 and E_2 where objects are mapped together if they correspond to the same variable in C
- Example:
  - E_1: $homeTeam(g1, dallas), awayTeam(g1, la),$ $scored(g1, dallas, 23), scored(g1, la, 15), leq(15,23),$ $winner(g1, dallas), loser(g1, la)$
  - E_2: $homeTeam(g2, portland), awayTeam(g2, ny),$ $scored(g2, portland, 34), scored(g2, ny, 32), leq(32,34),$ $winner(g2, portland), loser(g2, ny)$
  - Mapping: $\{dallas \leftrightarrow portland, la \leftrightarrow ny, 15 \leftrightarrow 32, 23 \leftrightarrow 34, g1 \leftrightarrow g2\}$

Results

- Result 1: By incorporating ROQs into Khardon's algorithm, the number of Membership Queries is greatly reduced, but not eliminated.
- Result 2: First-order Horn theories can be exactly learned in polynomial time using only PQs and EQs.

Next Steps

- Experimental test of how well users can answer each of these types of queries
- Theoretical studies of imperfect oracles
  - Try to model the kinds of errors teachers are likely to make

Research Space

	Supervised Learning	Imitation Learning	Reinforcement Learning
Current Methods	Label feedback	Demonstrations	Demonstrations
	Active learning for labels	Active learning via online action feedback	Active learning via online action feedback
	Equivalence queries and Membership Queries		
Novel Methods	Feature Labeling by end users [IUI 2011]	State Queries with ⊥ responses [ICML Workshop 2010]	Practice & Critiques [AAAI 2010]
	Object Queries and Pairing Queries [ECML 2011]		

Learning from Demonstrations and State Queries

- **Setting:**
  - Teacher has a policy π^T for selecting actions in a Markov Decision Problem (MDP) with states S and actions A
  - Learner has access to a simulator for the dynamics of the MDP:

    $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$ // next state distribution

    $s_0 \sim P(s_0)$ // start state distribution
  - Teacher provides *training trajectories* (demonstrations)
    - $\langle (s_1^1, a_1^1), \ldots, (s_H^1, a_H^1) \rangle$
    - $\langle (s_1^2, a_1^2), \ldots, (s_H^2, a_H^2) \rangle$
  - Learner's Goal: Learn the Teacher's policy over the first H steps
  - Note: No reward function!

State Queries

- The Learner can ask the Teacher state queries:
  - Learner: "What action should be performed in state s?"
  - Teacher:
    - If π^T visits state s with non-zero probability, then return action $a = \pi^T(s)$
    - Else, return $\perp$, which means "bad state"

Queries that result in Bad State

- Model cases where the Teacher doesn't know what to do
  - Teacher is not reliable in such cases
- Avoid unnecessary complexity in the learned policy
  - The Teacher's policy π^T doesn't need to model such cases, which can make learning the Learner's policy easier

Proposed Method: Extension of Bayesian Active Learning

- Space of hypotheses $\{\pi_1, \ldots, \pi_i, \ldots\}$
- Space of Teacher responses $X = \{a_1, \ldots, a_n, \perp\}$
- Demonstrations + query answers = D
- Posterior distribution $P(\pi_i|D)$
- $P(s, x|D)$ = posterior probability of those hypotheses that would respond to state query s with x
  - $P(s, a|D) = \sum_i P(\pi_i|D) I[\![P(s|\pi_i) > 0]\!] I[\![a = \pi_i(s)]\!]$
  - $P(s, \perp |D) = \sum_i P(\pi_i|D) I[\![P(s|\pi_i) = 0]\!]$
- Intuition: Student should learn to predict the Teacher's query responses (including Bad State responses)

Query Rule

- Choose s to maximize

$$VE(s) = -\sum_x P(s, x|D) \log P(s, x|D)$$

- Greedy reduction in our uncertainty about $P(s, x|D)$
- Let $P(a|s, D) = \sum_i P(\pi_i|D) I[\![\pi_i(s) = a]\!]$
- Let $\alpha(s) = \sum_i P(\pi_i|D)\, I[\![P(s|\pi_i) > 0]\!]$
- Then

$$VE(s) = \alpha(s) H(a|s, D) + H\big(\alpha(s), 1 - \alpha(s)\big)$$

[**Posterior prob. of target policy visiting *s*** $\times$ **Uncertainty over action choices at *s***] $+$ **Bonus that is maximized when** $\alpha(s) = 0.5$

A Practical Algorithm: Imitation Query-By-Committee (IQBC)

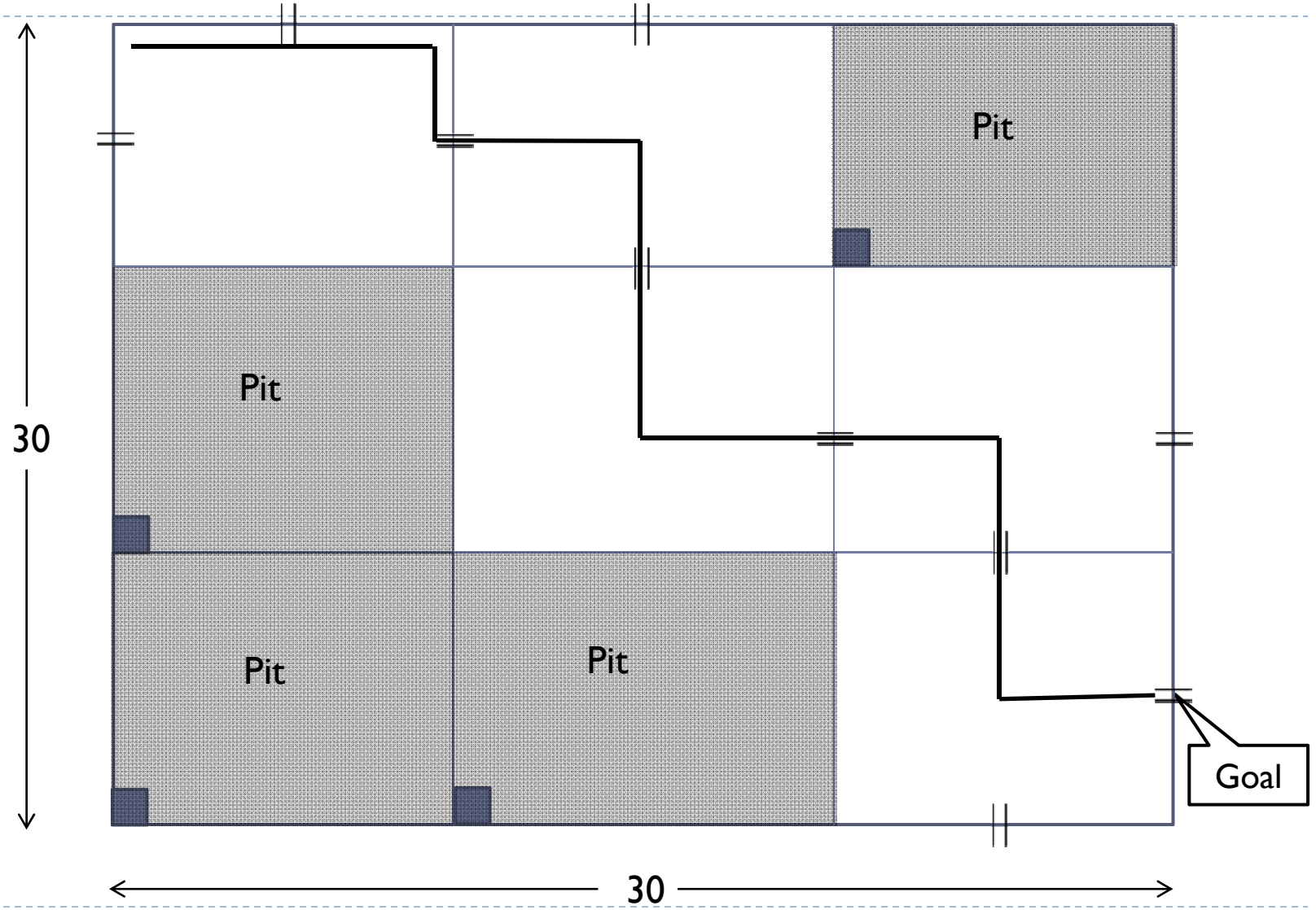
- Treat demonstrations and non-$\perp$ Teacher responses as training examples for supervised learning
- Represent each policy as a multi-class classifier the predicts the action to take in each state
- Approximate the posterior distribution over π by a committee learned using bagging

- This does not make any use of $\perp$ responses during learning
- Query Rule requires computing probability that each policy π_i visits each state s
  - Sample Average approximation (Pegasus-approach) for stochastic MDPs
- Only query in states visited by at least one π_i

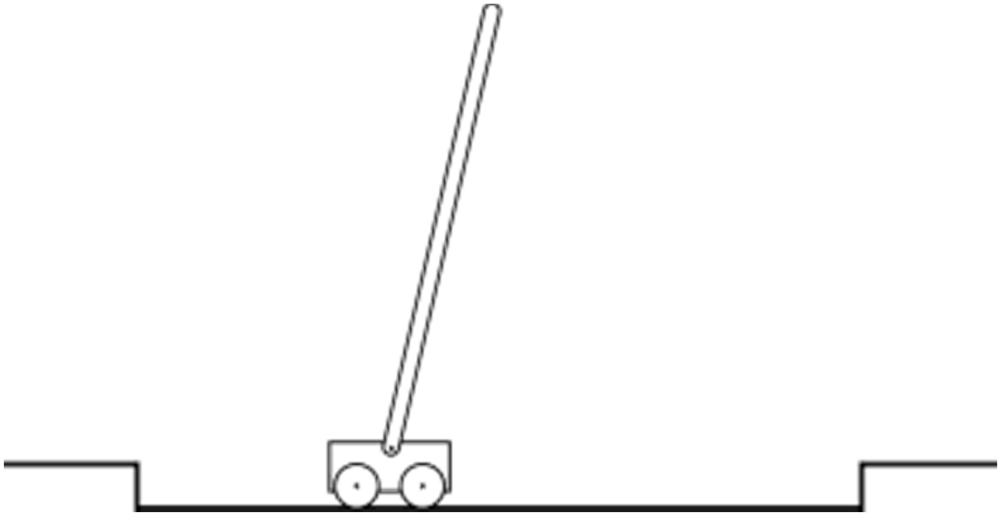
Experimental Test of the Method

- Domains:
  - Grid world with pits
  - Cart-pole
- Algorithms:
  - IQBC
  - Random: Selects stats to query uniformly at random from S
  - Standard QBC (SQBC): Ignores $I[\![P(s|\pi_i) > 0]\!]$
  - Passive imitation learning (Passive): Execute learned policy and ask teacher what to do in each state
  - Confidence based autonomy (CBA) (Chernova & Veloso, JAIR 2009):
    - Executes policy until confidence falls below an automatically-adjusted threshold, then query Teacher

Grid World With Pits



Cart Pole

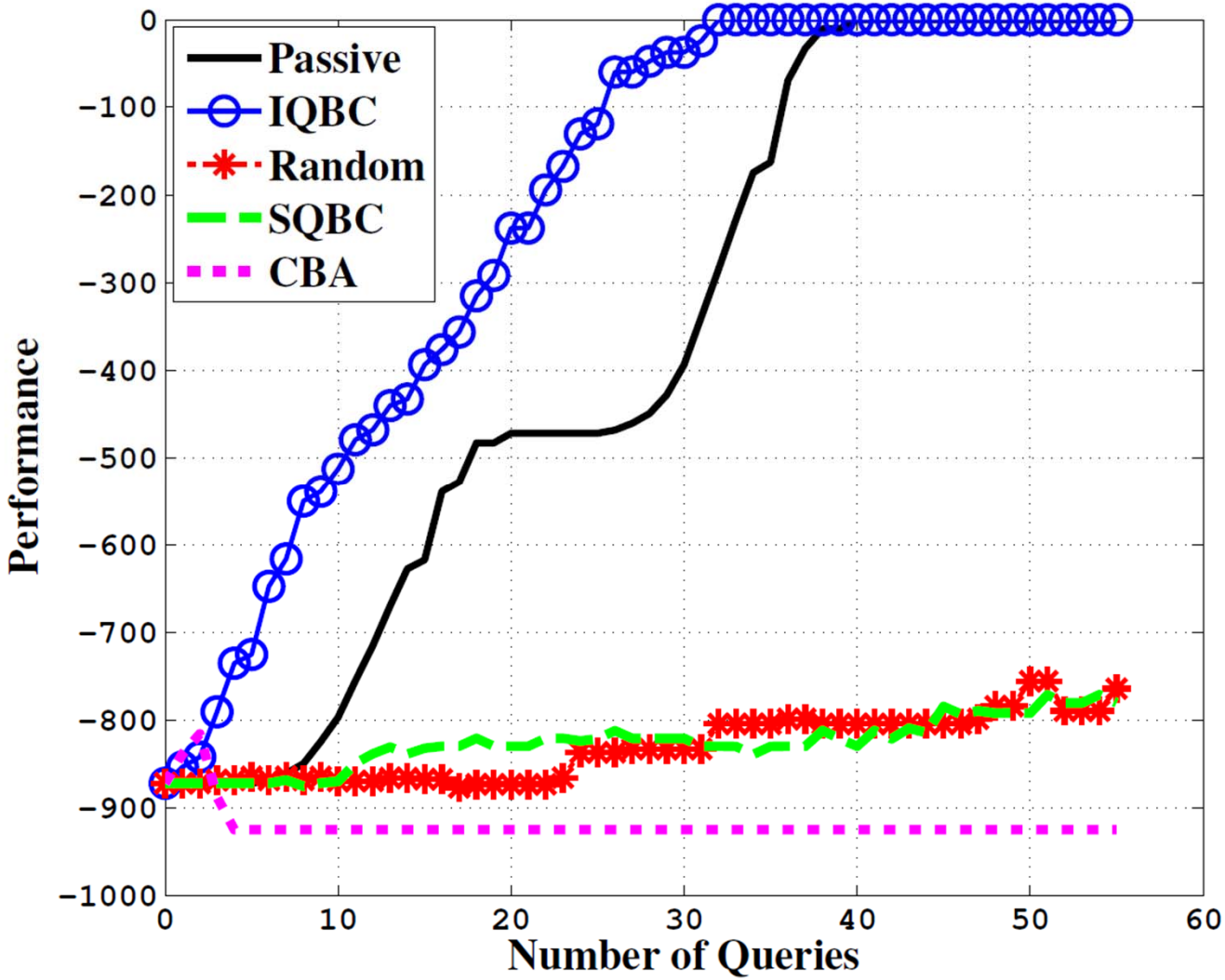


- State: $(x, \dot{x}, \theta, \dot{\theta})$
- Actions: {Left, Right}
- Bounds on cart position: $none$
- Bounds on pole angle: $[-90,90]$

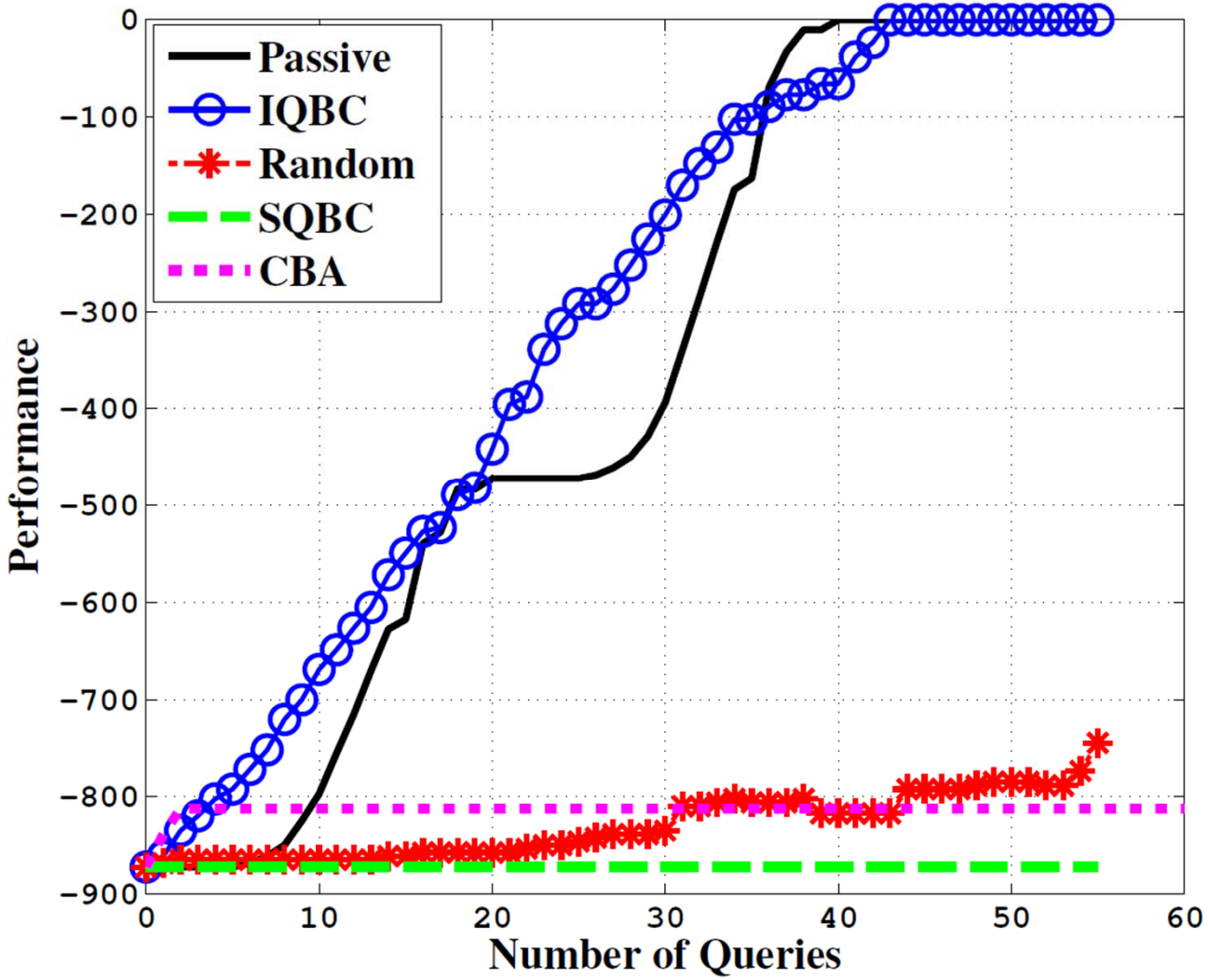
Teacher Types

- "Generous": always responds with an action $a \in A$
- "Strict": declares states >2 steps away from states visited by π^T as bad states

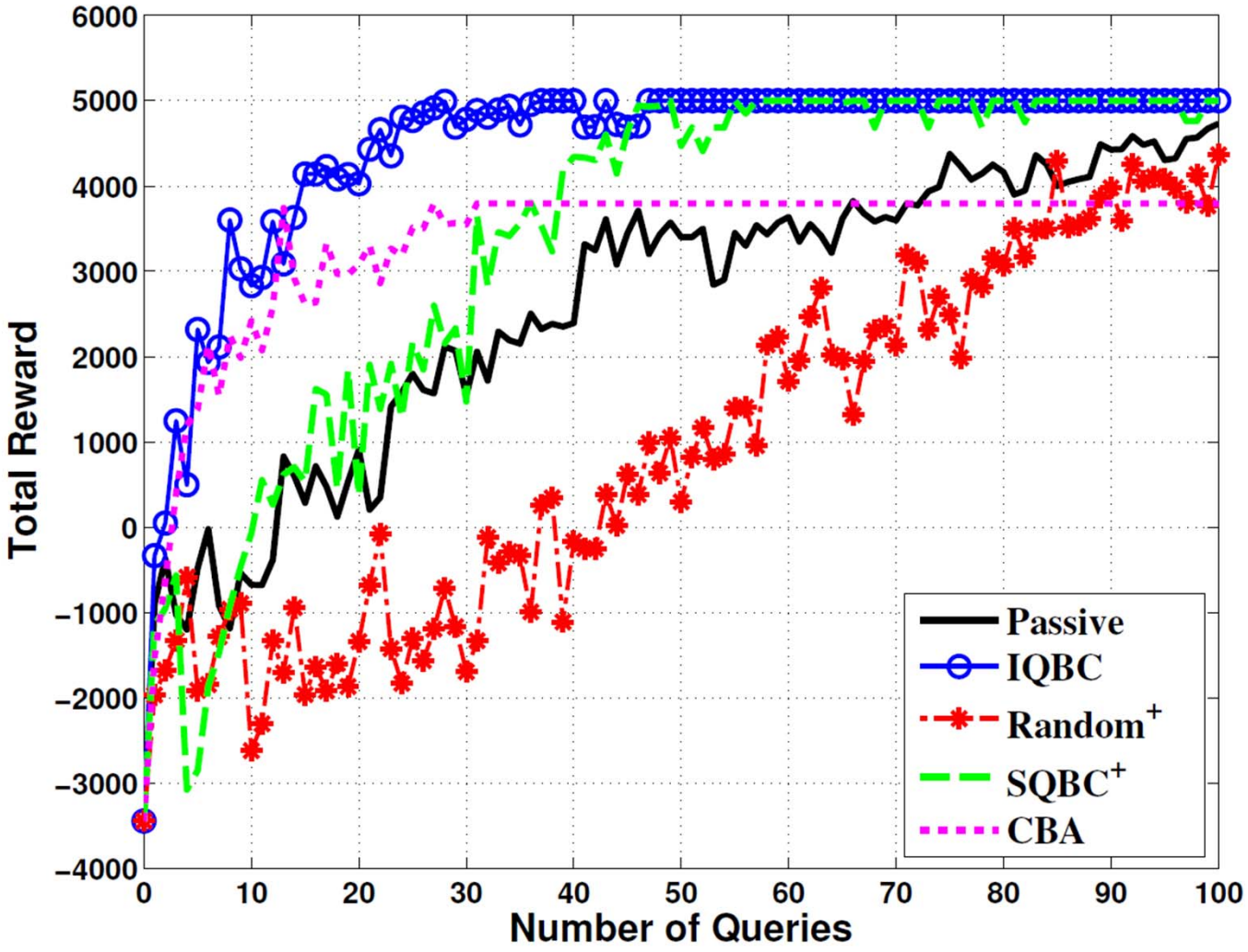
Grid World With Pits: Generous Teacher



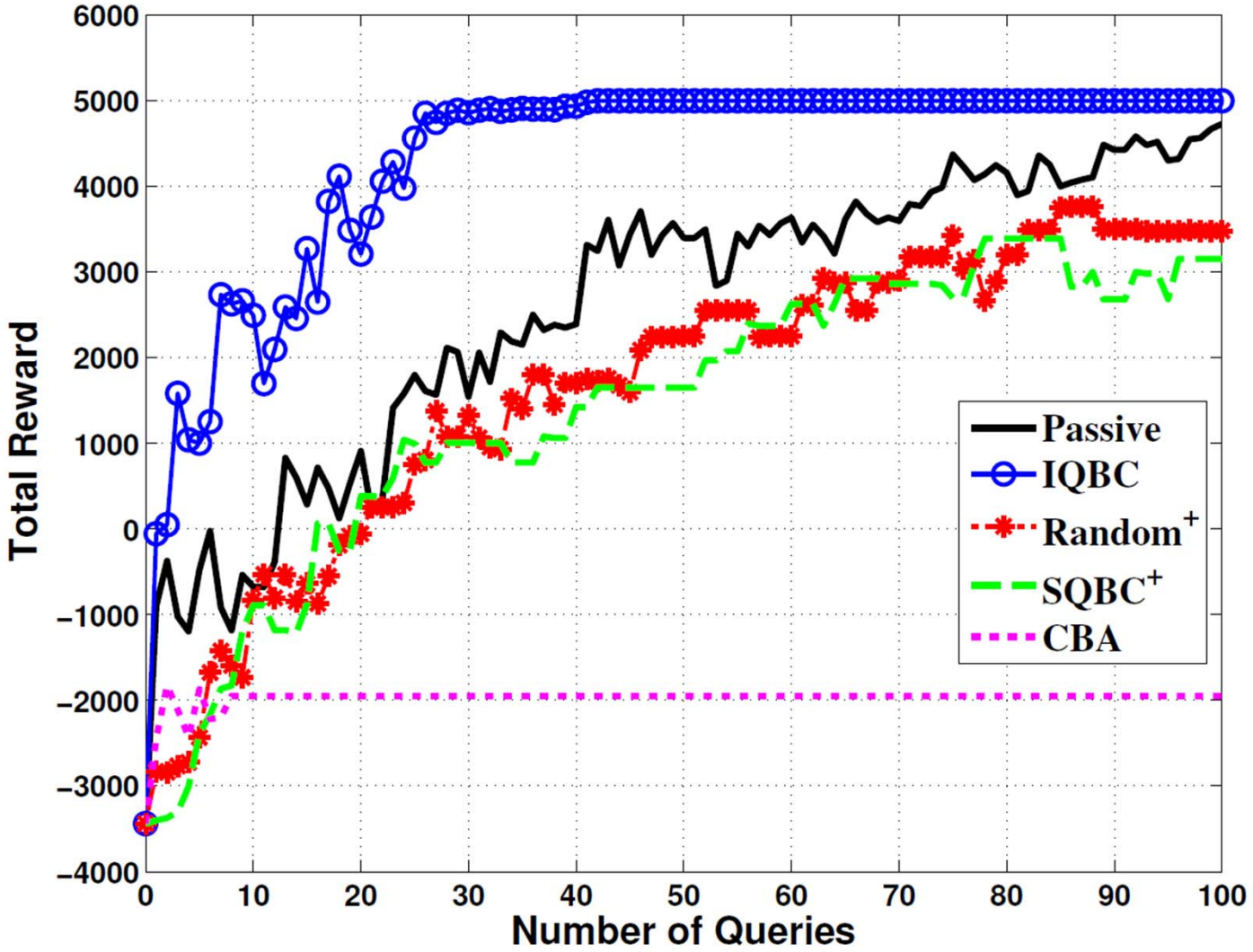
Grid World With Pits: Strict Teacher



Cart Pole: Generous Teacher



Cart Pole: Strict Teacher



Conclusions

- IQBC outperforms previous active learning algorithms for Imitation Learning
  - It is important to take the MDP dynamics into consideration when choosing states to query
  - The certainty threshold in CBA is very sensitive and can easily lead to premature convergence

Next Steps

- Incorporate ⊥ feedback when learning policies
- Consider the "mental cost" to the Teacher of understanding the query state s
  - Perhaps present short state sequences ("scenarios") and ask the Teacher to provide correct actions and/or ⊥ feedback for each state
- Conduct user studies to test the hypothesis that ⊥ feedback is easier to provide

Research Space

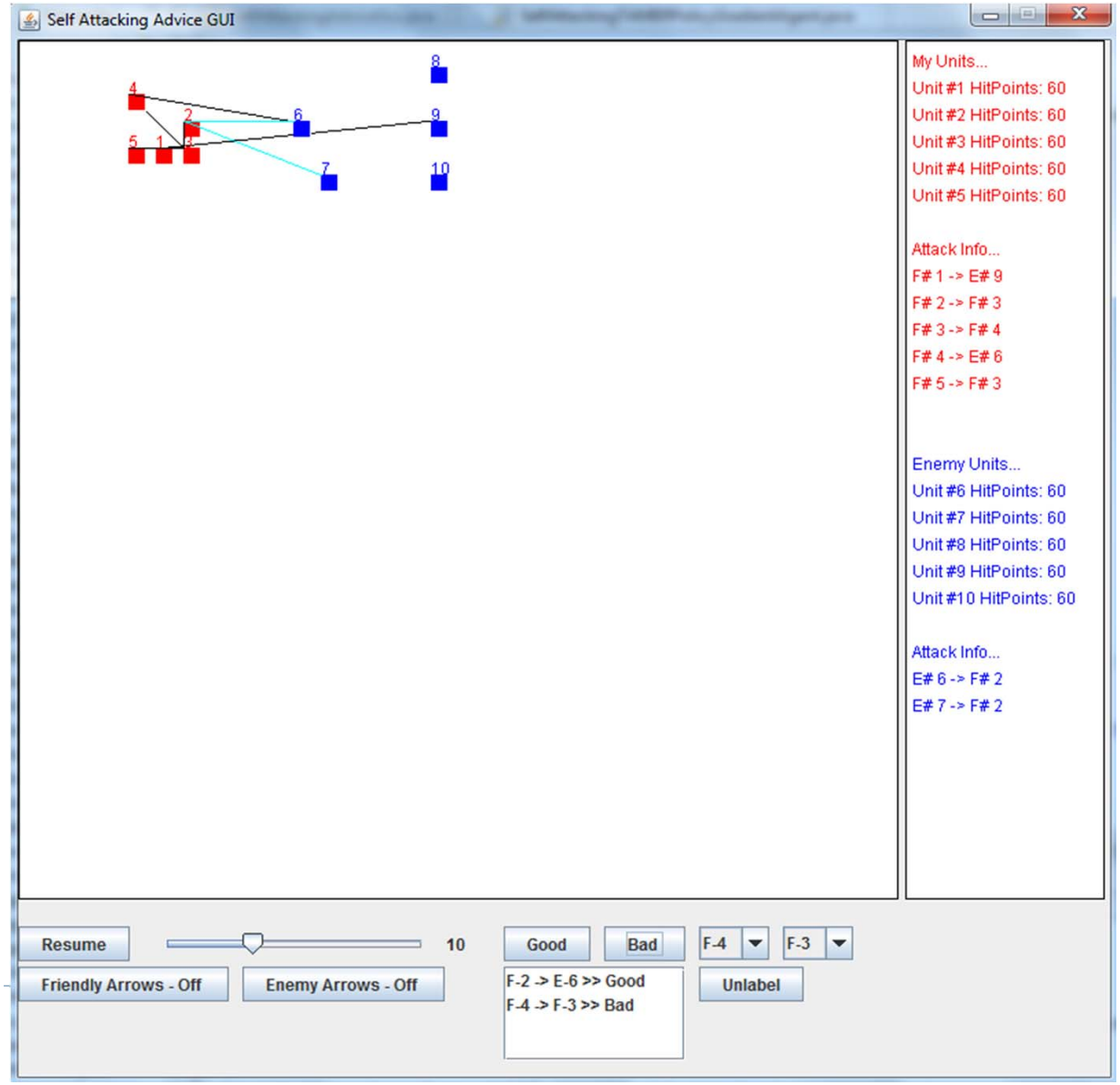
	Supervised Learning	Imitation Learning	Reinforcement Learning
Current Methods	Label feedback	Demonstrations	Demonstrations
	Active learning for labels	Active learning via online action feedback	Active learning via online action feedback
	Equivalence queries and Membership Queries		
Novel Methods	Feature Labeling by end users [IUI 2011]	State Queries with ⊥ responses [ICML Workshop 2010]	Practice & Critiques [AAAI 2010]
	Object Queries and Pairing Queries [ECML 2011]		

Reinforcement Learning from Critiquing and Practice

- Setting:
  - Standard reinforcement learning setting
    - Learner has access to an MDP and can learn via standard exploration policies
  - From time to time, the Learner can show the Teacher a trajectory from its current best policy π_t
  - Teacher can choose any state or states along the trajectory and provide feedback of the form of
    - Good actions in state s: A_{good}
    - Bad actions in state s: A_{bad}
    - Feedback: $\langle s, A_{good}, A_{bad} \rangle$
    - Either A_{good} or A_{bad} can be empty

Application Problem: Tactical Battles in Wargus

- Wargus: Open Source version of Warcraft II
- We provide a GUI that allows the Teacher to scroll backwards/forwards in the game and find states to critique



Learning Algorithm

- Assume space of policies parameterized by Θ
- Let
  - C = Critiquing examples
  - T = Observed $\langle s_t, a_t, r_t, s_{t+1} \rangle$ tuples along the Learner's exploratory trajectories
- Find Θ to maximize
  - $J(\Theta, C, T) = \lambda U(\Theta, T) + (1 - \lambda) L(\theta, C)$
  - where
    - $U(\Theta, T)$ is the estimated expected return of policy π_Θ
      - Evaluated via off-policy importance sampling [Peshkin & Shelton, 2002]
    - $L(\Theta, C)$ is the log likelihood of the Teacher's critiques under π_Θ
    - $L(\Theta, C) = \sum_i \log\left(1 + \pi_\Theta\left(A_{good}(s_i)\right) - \pi_\Theta(A_{bad}(s_i))\right)$
    - λ is a parameter that trades off the two criteria

Experimental Setup

Map 1

Map 2

- Our Domain: Micro-management in tactical battles in the Real Time Strategy (RTS) game of Wargus.
- 5 friendly footmen against a group of 5 enemy footmen (Wargus AI).
- Two battle maps, which differed only in the initial placement of the units.
- Both maps had winning strategies for the friendly team and are of roughly the same difficulty.

Experimental Details

- RL agent
  - Log-linear model over 27 hand-coded features
  - Choose action for each unit every 20 game cycles
  - Same policy applied to all units (independently)
- Each Practice Phase
  - Generate 10 trajectories
  - With probability 0.8: choose action according to π_{Θ}
  - With probability 0.2: choose action according to $\pi_{\alpha\Theta}$
    - Where α shrinks the weights, which causes the policy to become more random

User Study

- Goal is to evaluate three systems
  - Pure Supervised = no practice session ($\lambda = 0$)
  - Pure RL = no critiques ($\lambda = 1$)
  - Combined = includes practice and critiques ($\lambda = 0.3$)

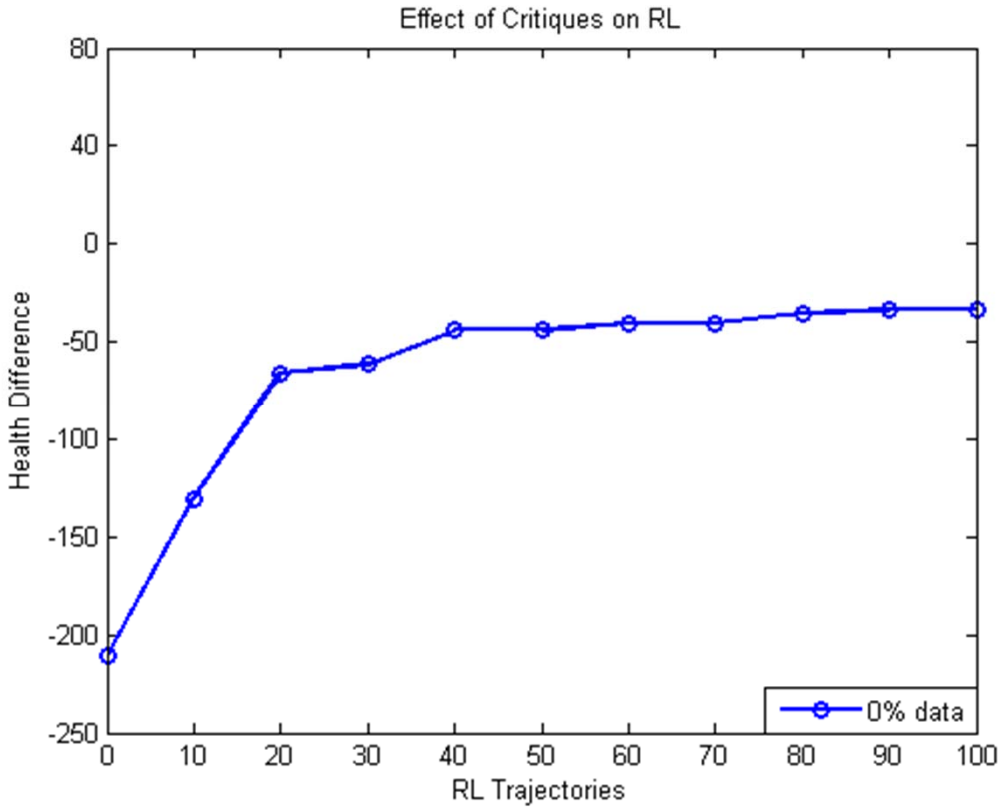
- The user study involved 10 end-users
  - 6 with CS background
  - 4 no CS background

- Each user trained both the supervised and combined systems
  - 30 minutes total for supervised
  - 60 minutes for combined (30 minutes of practice)

Simulated Learning Curves

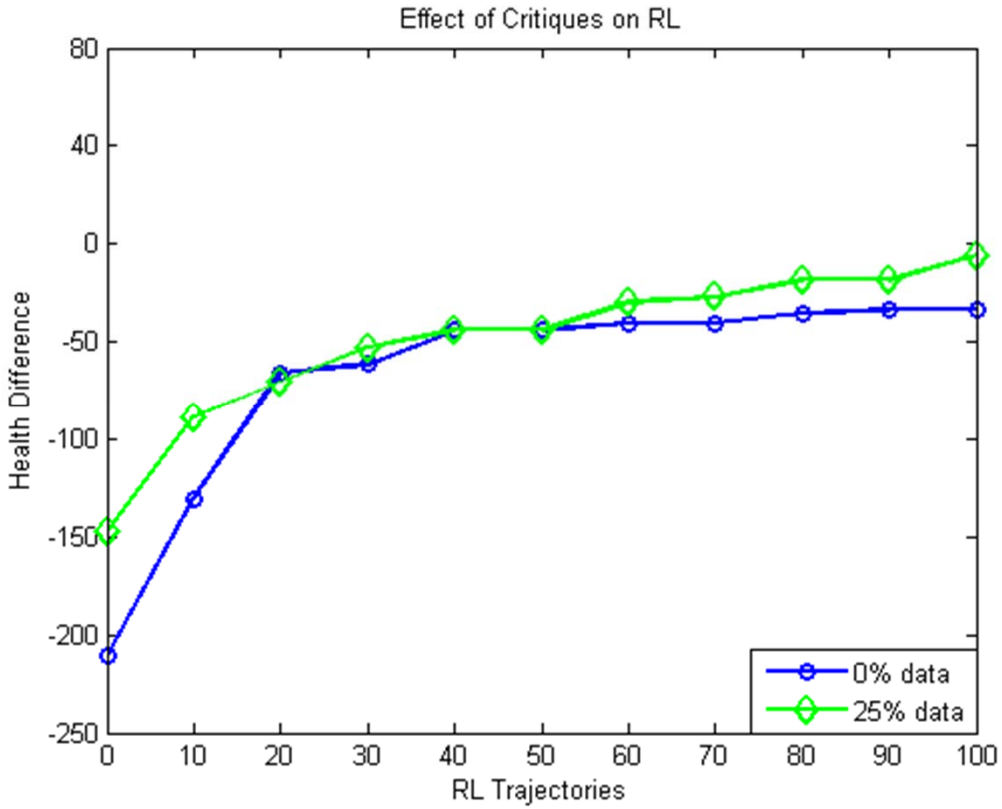
- After user study, selected the worst- and best-performing users on each map when training the Combined system
- Total Critique data:
  - User#1: 36, User#2: 91, User#3: 115, User#4: 33.
- For each user:
  - divide critique data into 4 segments containing 25%, 50%, 75%, and 100% of the data
- Evaluate the Combined system varying both the amount of practice and the amount of critique data

Simulated Experiments: Benefit of Critiques from User #1



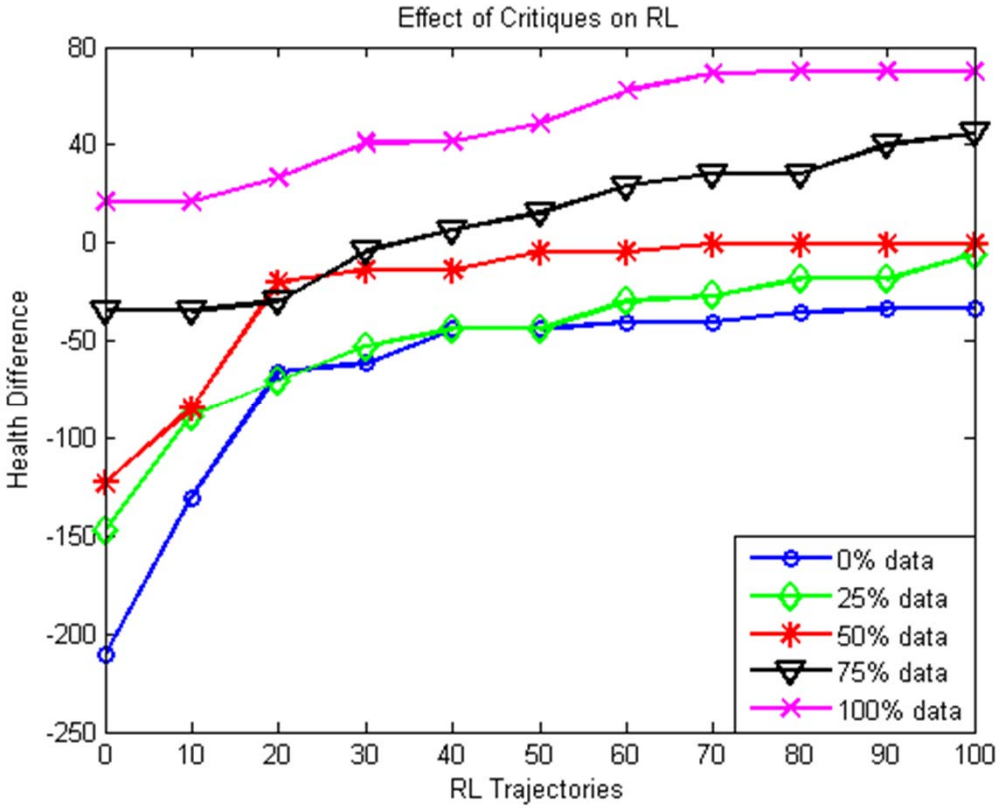
- RL is unable to learn a winning policy (i.e. achieve a positive value).

Simulated Experiments: Benefit of Critiques from User #1



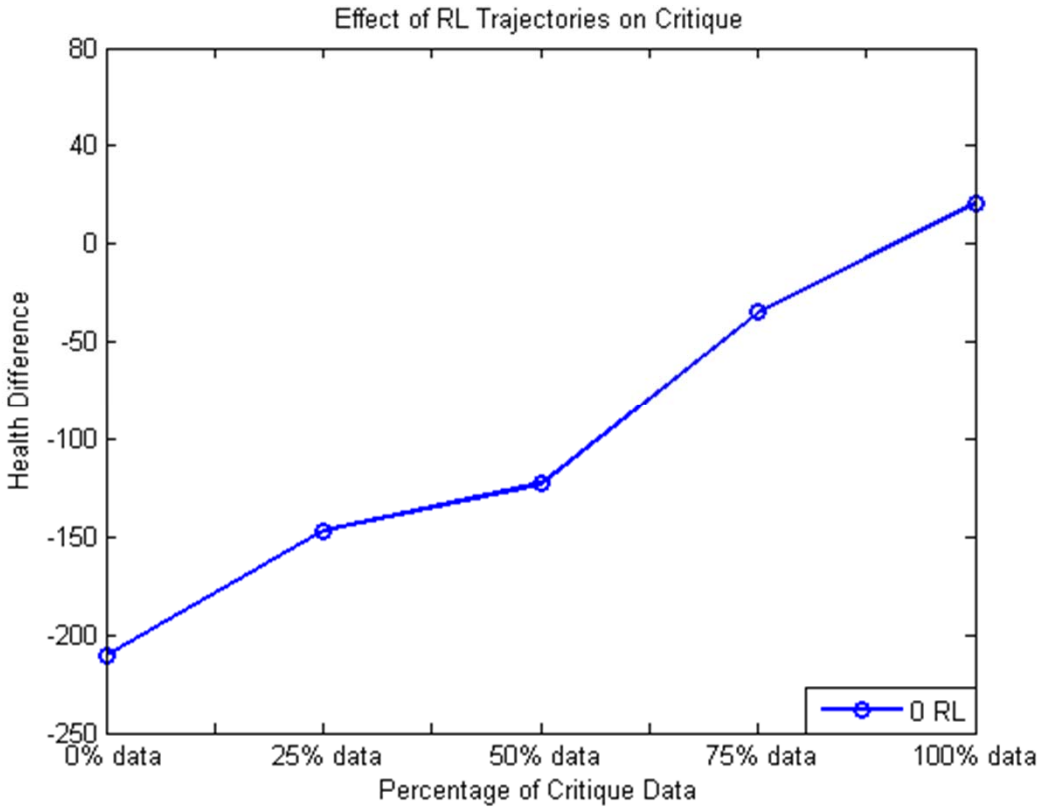
- With more critiques performance increases a little bit.

Simulated Experiments: Benefit of Critiques from User #1



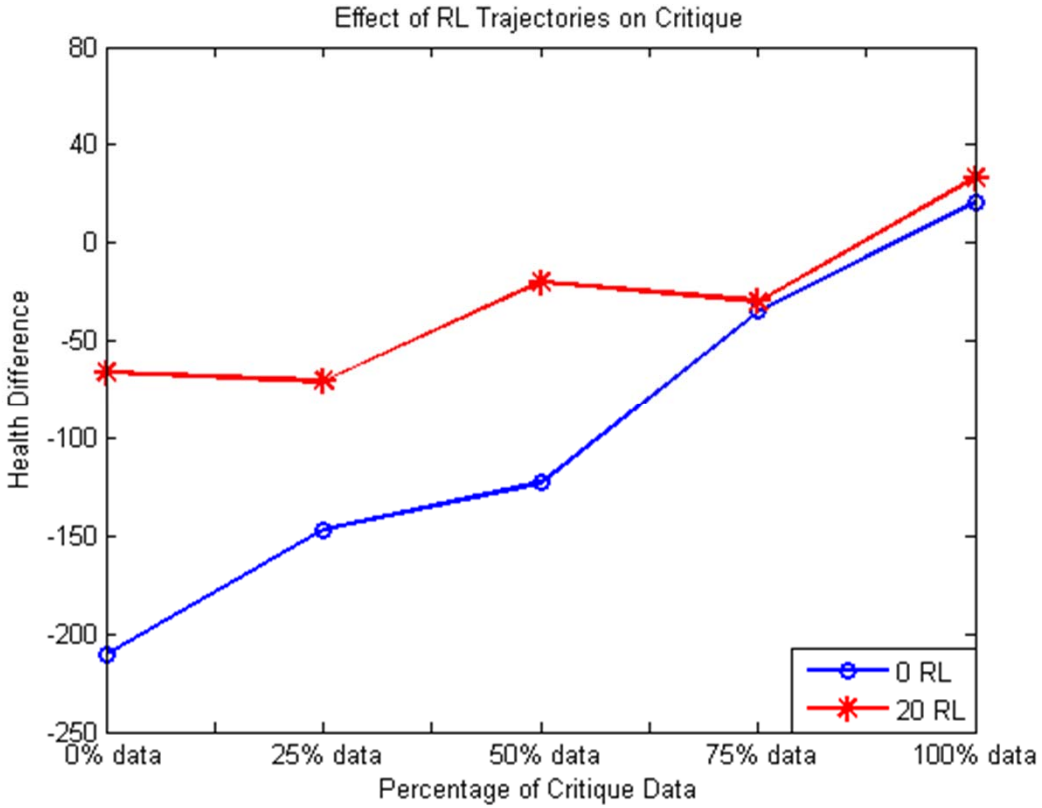
- As the amount of critique data increases, the performance improves for a fixed number of practice episodes.
- RL did not go past 12 health difference on any map even after 500 trajectories.

Simulated Experiments: Benefit of Practice for User #1



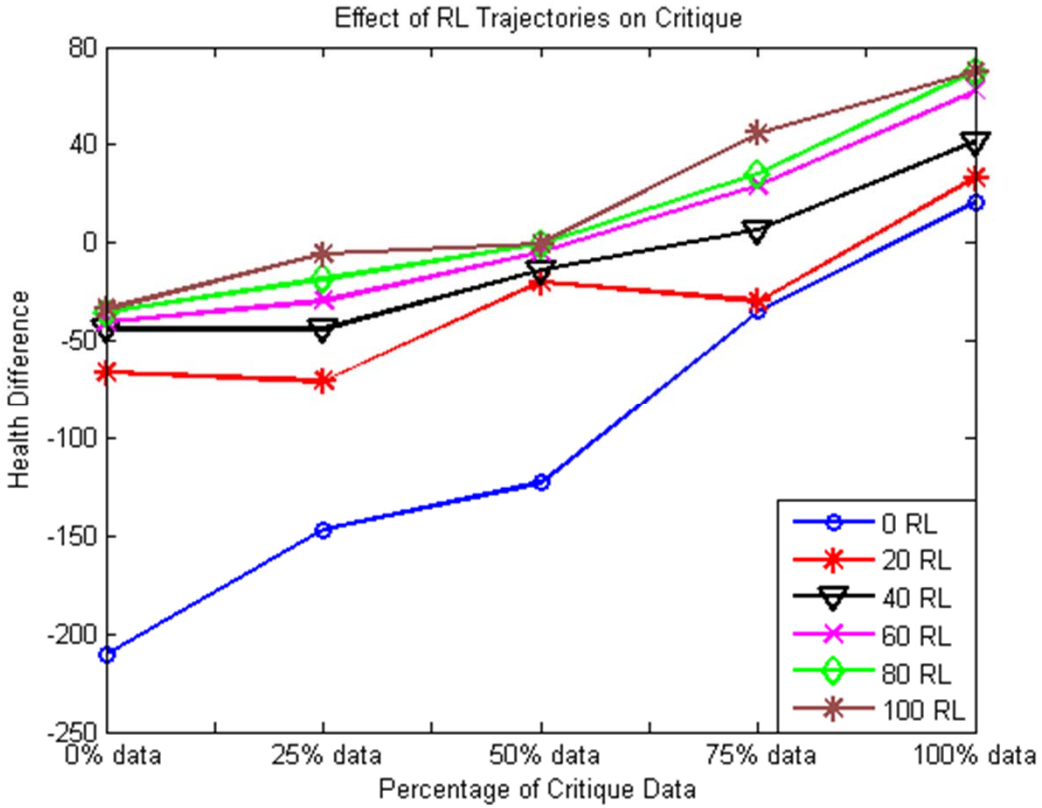
- Even with no practice, the critique data was sufficient to outperform RL.
- RL did not go past 12 health difference.

Simulated Experiments: Benefit of Practice for User #1



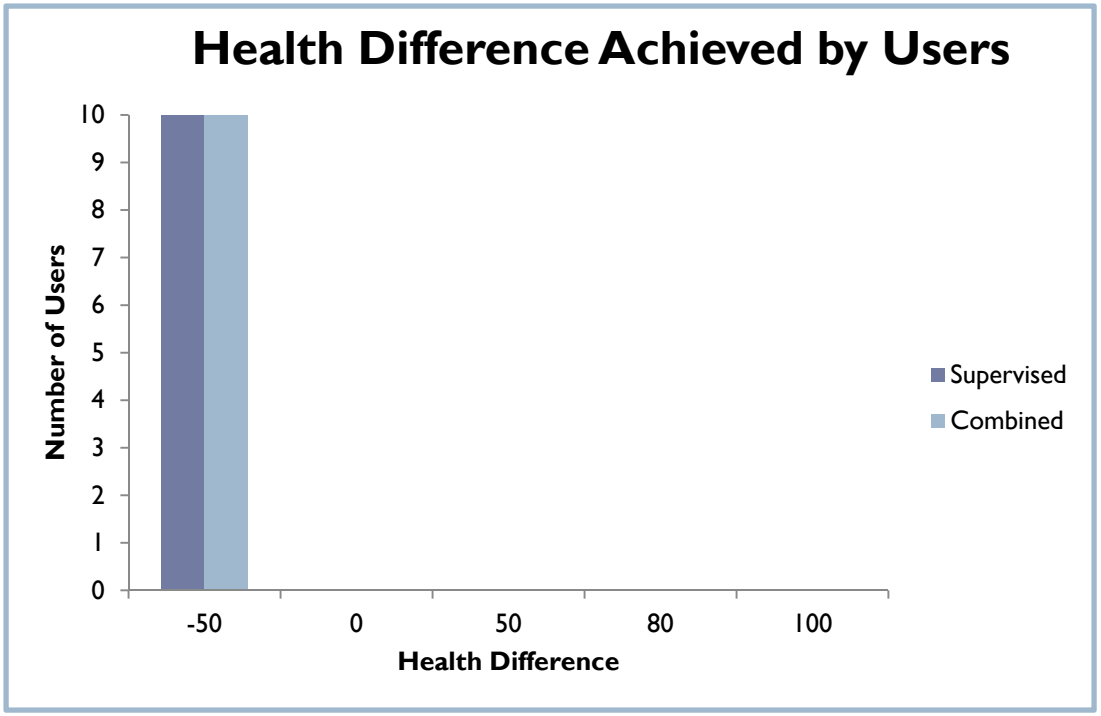
- With more practice performance increases too.

Simulated Experiments: Benefit of Practice for User #1

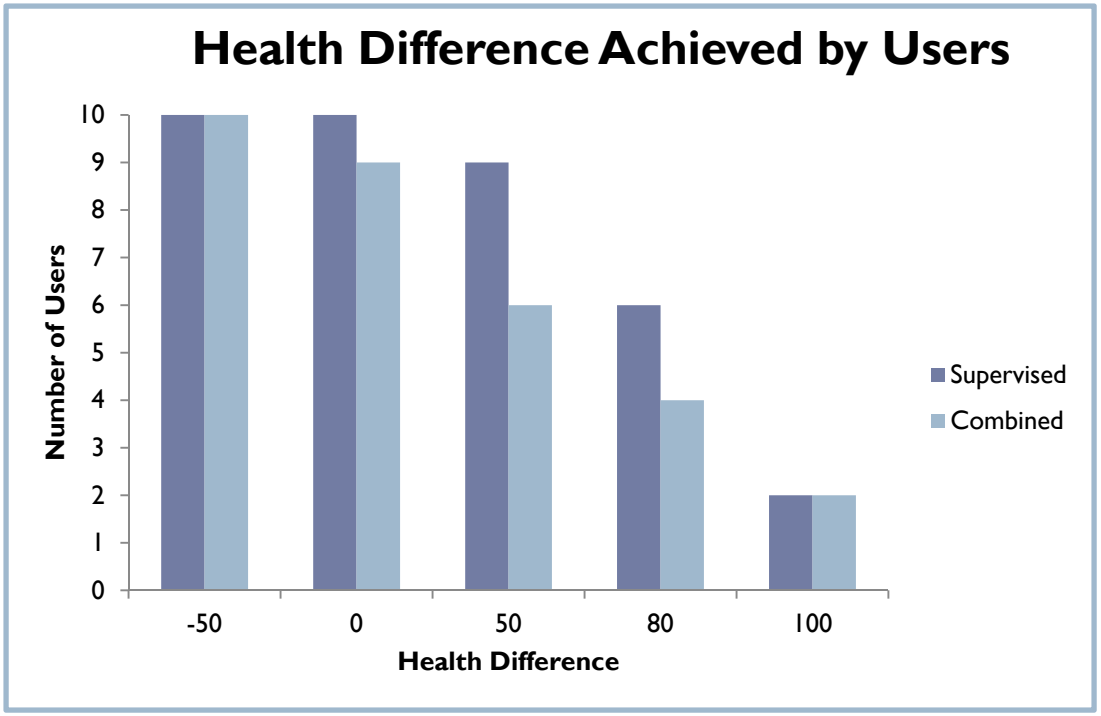


❑ Our approach is able to leverage practice episodes in order to improve the effectiveness on a given amount of critique data.

Results of User Study

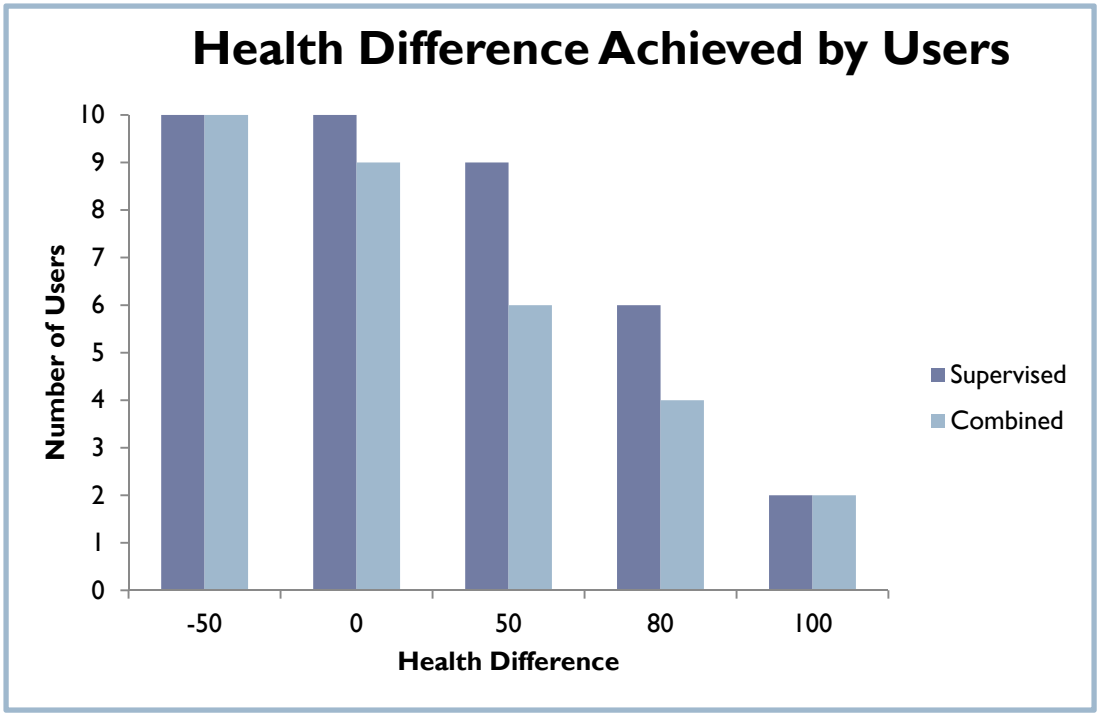


Results of User Study



- Pure RL did not go past 12 health difference on any map even after 500 trajectories.

Results of User Study



- Users were slightly more successful using the purely supervised method (no practice)

Conclusions

- Combining Teacher critiques with practice has potential to speed learning
- User study did not achieve this potential
  - Insufficient practice
  - Users complained that the combined system "ignored them"

Summary

	Supervised Learning	Imitation Learning	Reinforcement Learning
Current Methods	Label feedback	Demonstrations	Demonstrations
	Active learning for labels	Active learning via online action feedback	Active learning via online action feedback
	Equivalence queries and Membership Queries		
Novel Methods	Feature Labeling by end users [IUI 2011]	State Queries with ⊥ responses [ICML Workshop 2010]	Practice & Critiques [AAAI 2010]
	Object Queries and Pairing Queries [ECML 2011]		

Summary

- End-users can reliably label features, and these can be exploited by local learning algorithms to speed up learning
- Horn Clause Theories can be learned exactly in polynomial time using the more-realistic Object Relevance and Pairing Queries
- Imitation Query-by-Committee is more effective than existing methods for learning a Teacher's policy in an MDP\R
- Combining RL with Critiquing feedback shows promise of speeding up reinforcement learning

Questions?