Penn

PRiML

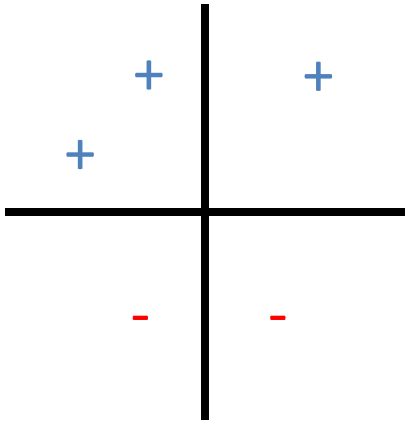
Penn Research in Machine Learning

Semi-Supervised Learning with Adversarially Missing Label Information

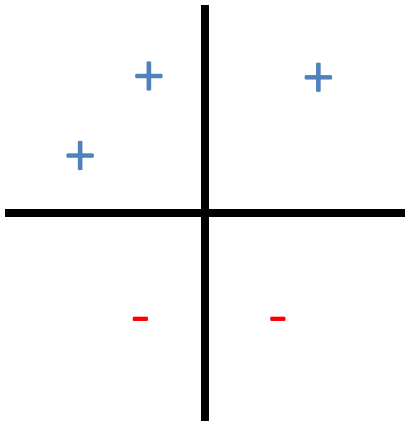
Umar Syed and Ben Taskar

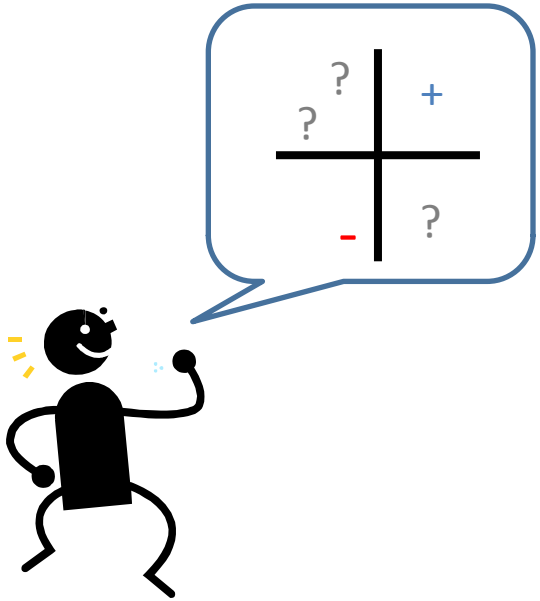
NIPS 2010

- Labeler examines training data …

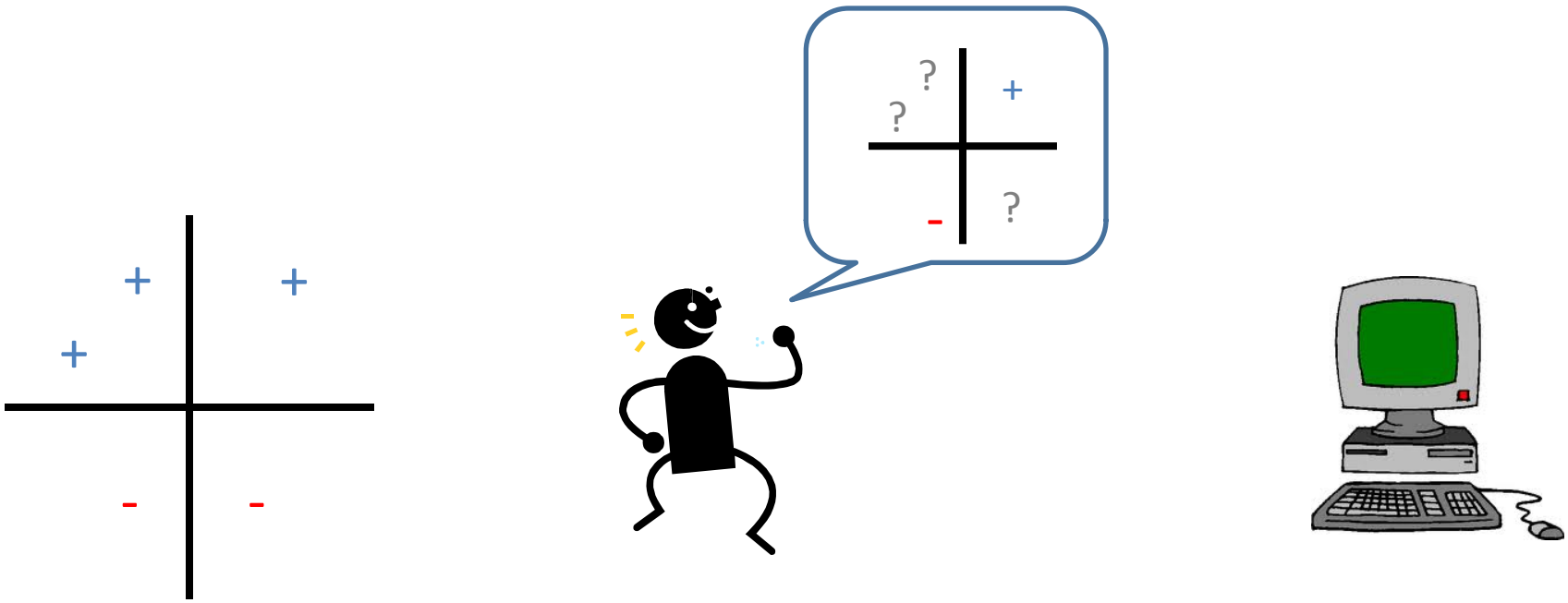


- Labeler examines training data … reveals <u>some</u> labels to learning algorithm.



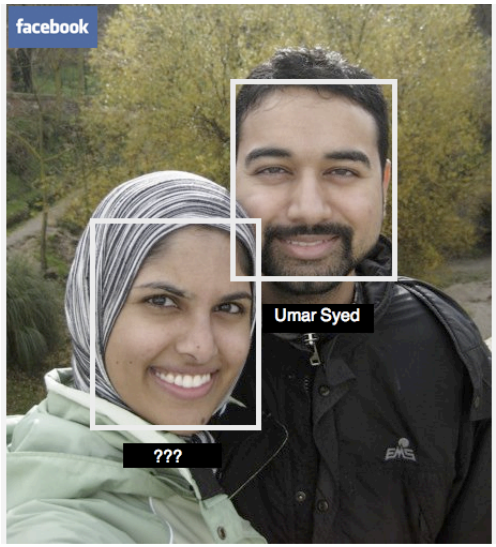


- Labeler examines training data … reveals some labels to learning algorithm.

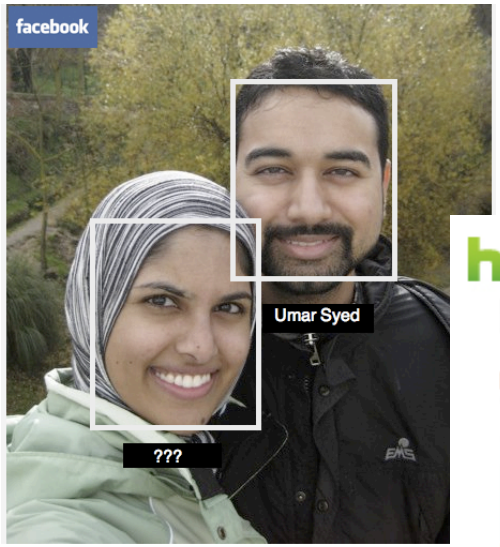


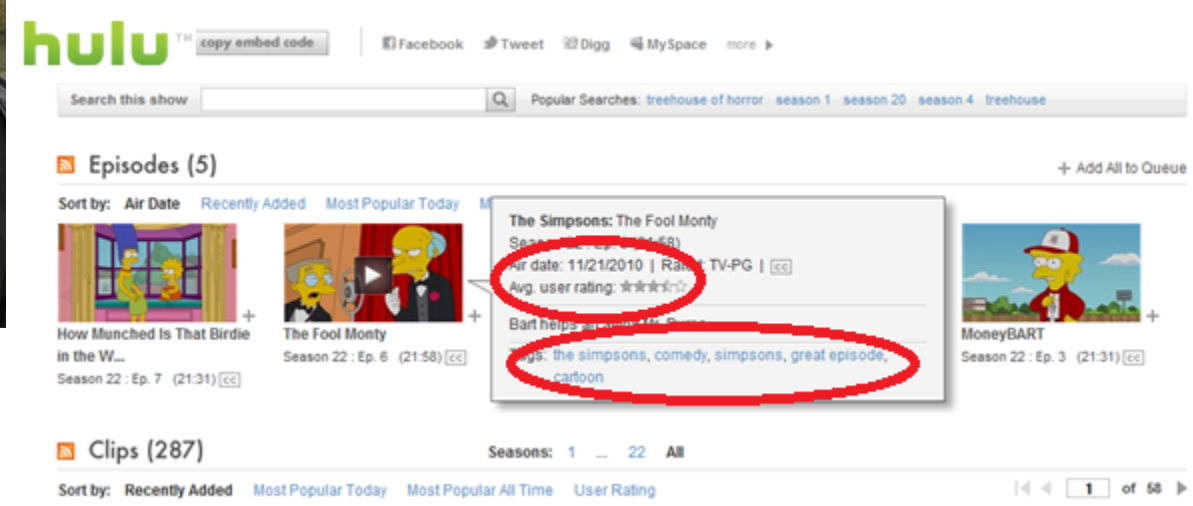
- Typical assumption: Labeled examples chosen randomly.

- “Naturally occurring” labeling is <u>not</u> random.

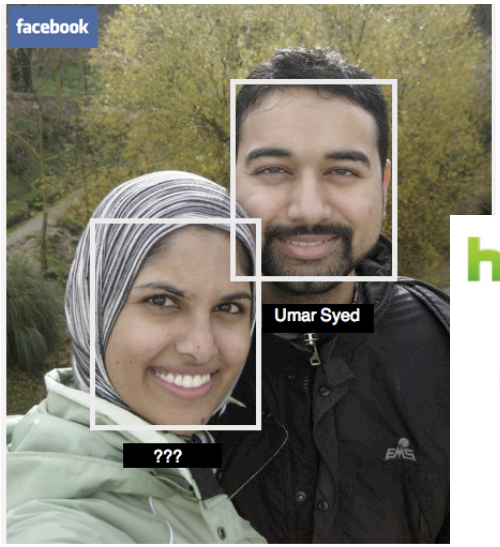


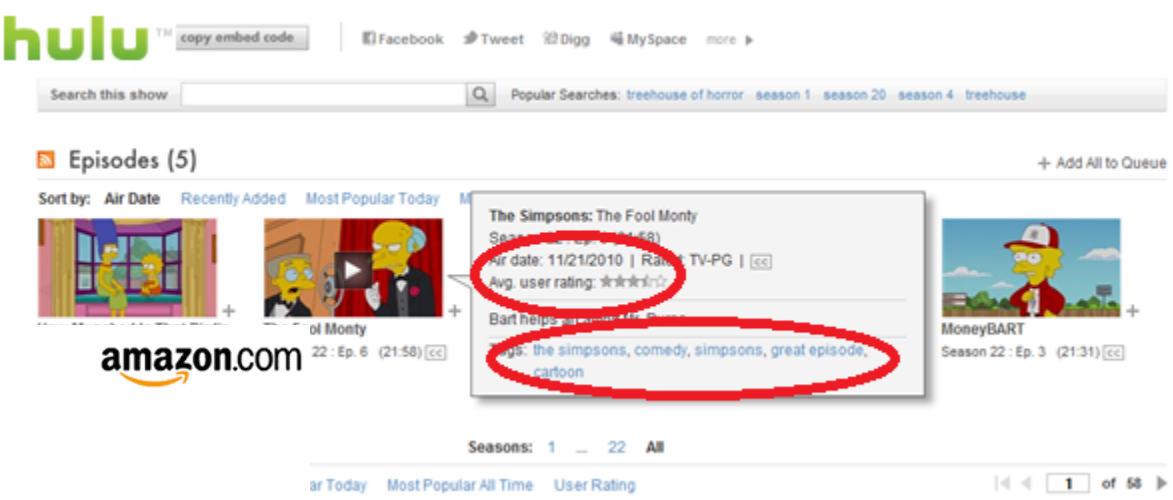
- “Naturally occurring” labeling is <u>not</u> random.





- “Naturally occurring” labeling is <u>not</u> random.





amazon.com

Most Helpful Customer Reviews

20 of 22 people found the following review helpful:

Excelent reference both for theory and practice, March 2, 2006

By J. J. Arrieta-Camacho "Wannabe Rocket Scientist" (Pittsburgh, PA USA) - See all my

REAL NAME

This review is from: **Convex Optimization (Hardcover)**

The book provides sound theoretical basis in a non-intimidating way. It also presents many exa reader understand and relate his or her specific needs to general convex optimization problems really good compromise between theory and practice: it can please the more mathematics-orie definitions, and bibliography; as well as the more application-oriented with examples, implemen The authors have been very generous in allowing the free download of the full book from their

Help other customers find the most helpful reviews

Was this review helpful to you? Yes No

Report abuse | Permalink

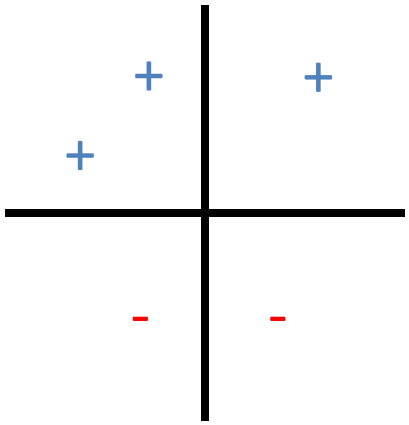
Comment

Our Framework

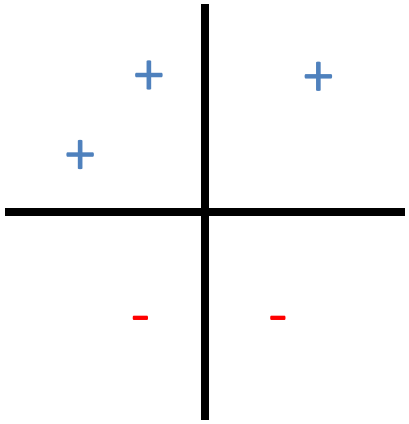
- Semi-supervised learning, but label information is adversarially missing.

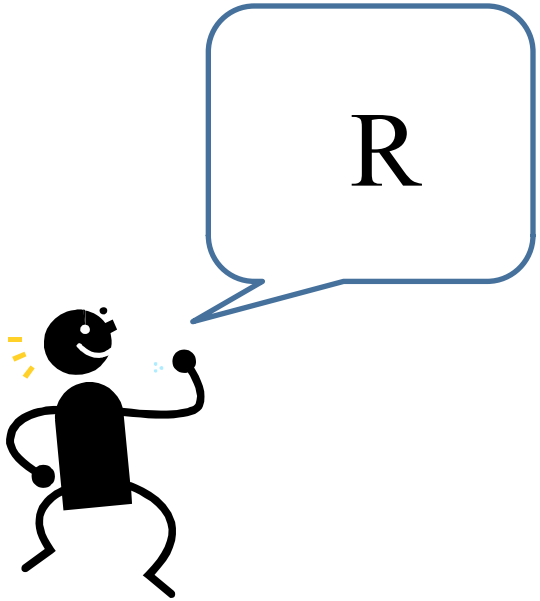
- Allows for local and global label information.

- **Idea:** Labeler examines training data $(\mathbf{x}, \mathbf{y})$ …

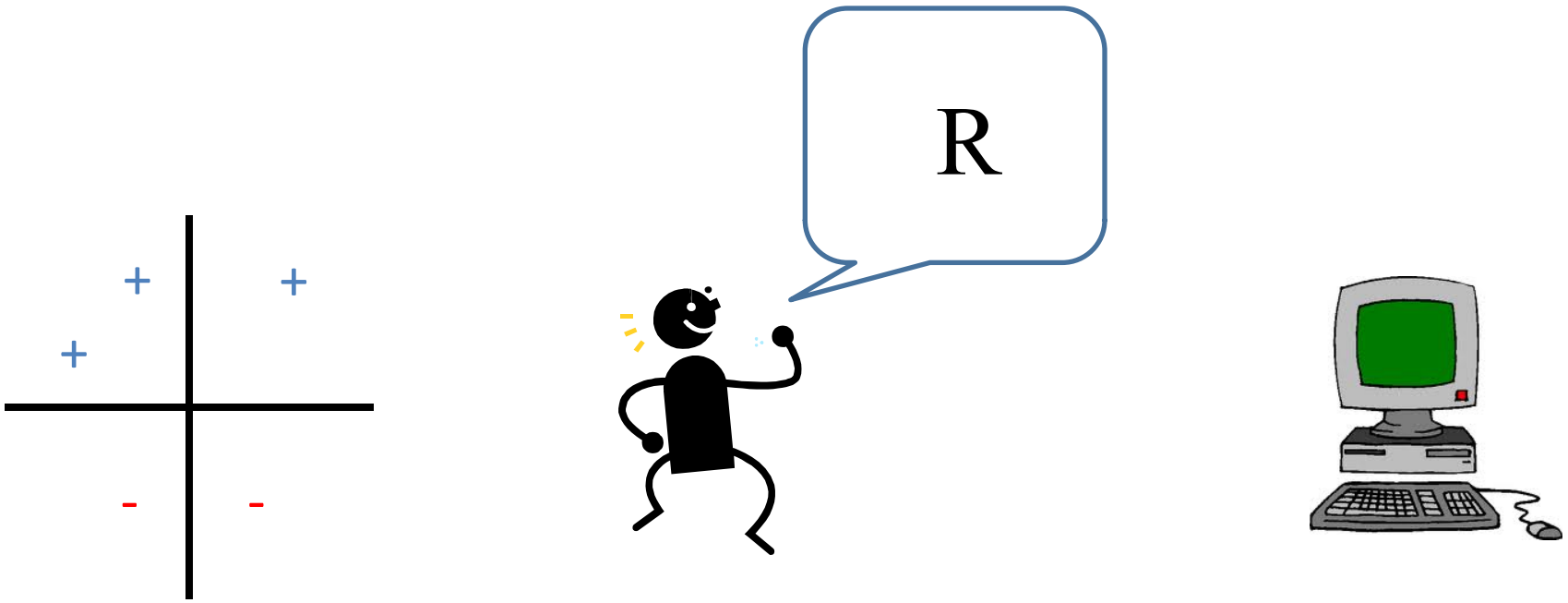


- **Idea:** Labeler examines training data $(\mathbf{x}, \mathbf{y})$ … reveals *label regularizer* function R.



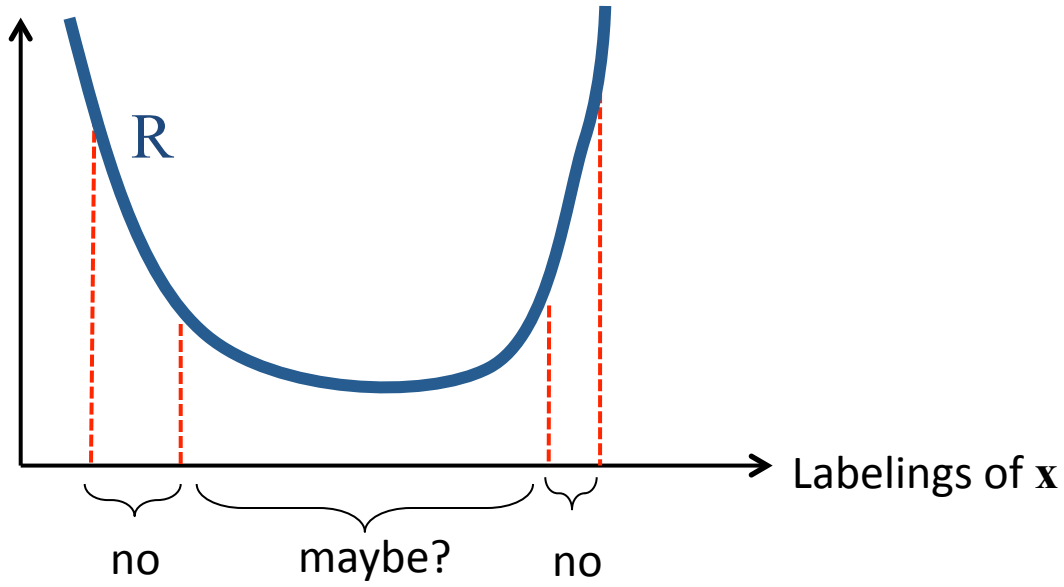


- **Idea:** Labeler examines training data $(\mathbf{x}, \mathbf{y})$ … reveals *label regularizer* function R.

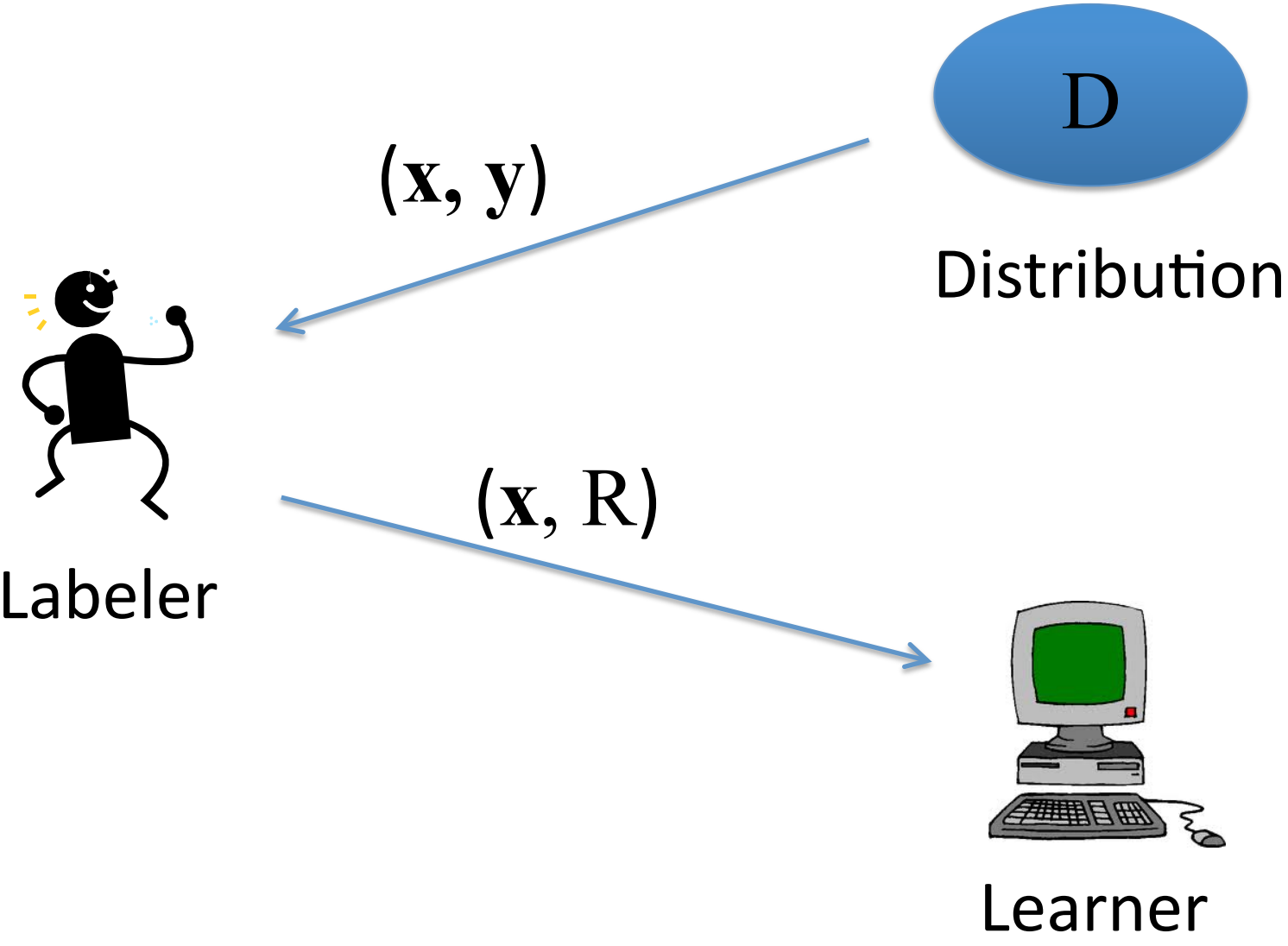


- R chosen <u>arbitrarily</u> from a known function family.
- R encodes all label information about training data.

- $\mathrm{R}(\mathbf{y})$ ≈ large ⇒ $\mathbf{y}$ likely not true labeling of $\mathbf{x}$.
- $\mathrm{R}(\mathbf{y})$ ≈ small ⇒ $\mathbf{y}$ may be true labeling of $\mathbf{x}$.



Overview of our Framework



Contributions

- Nearly tight upper/lower bounds on true loss under worst-case assumptions.
- Efficient learning algorithm that minimizes (convex) upper bound.
- Experiments show that algorithm is robust against "unhelpful" labelers.

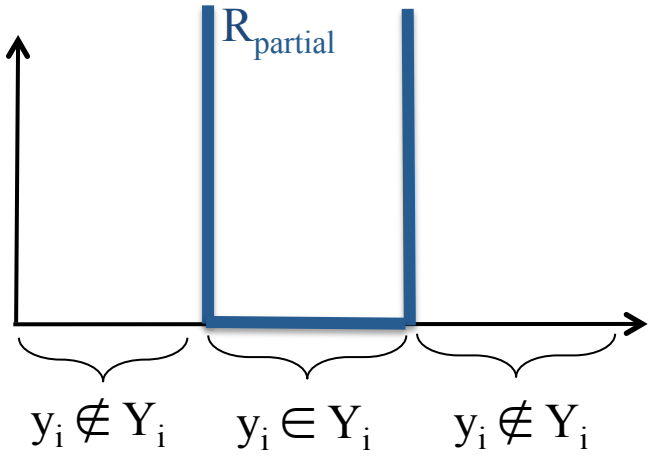
Related Work

- Compatibility functions (Balcan & Blum 05)
  - Similar to label regularizers, but not adversarially chosen.

- Malicious label noise (Kalai et al 05, Klivans et al 09)
  - Our analysis is valid for this setting, but very loose (more later).

Label Regularizer: Partial Label Sets

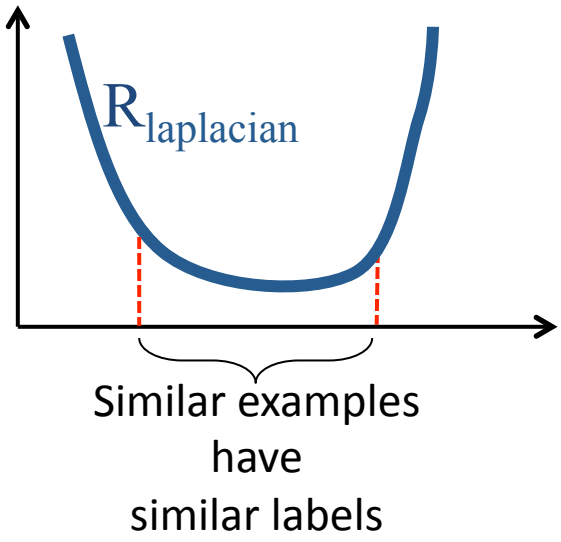
- Labeler reveals label set Y_i per training example x_i:



$$R_{partial}(\mathbf{y}) = \begin{cases} 0 \text{ if } y_i \in Y_i \text{ for all i.} \\ \infty \text{ otherwise.} \end{cases}$$

Label Regularizer: Graph Laplacian

- Labeler reveals similarity score w_{ij} for each training example pair (x_i, x_j):

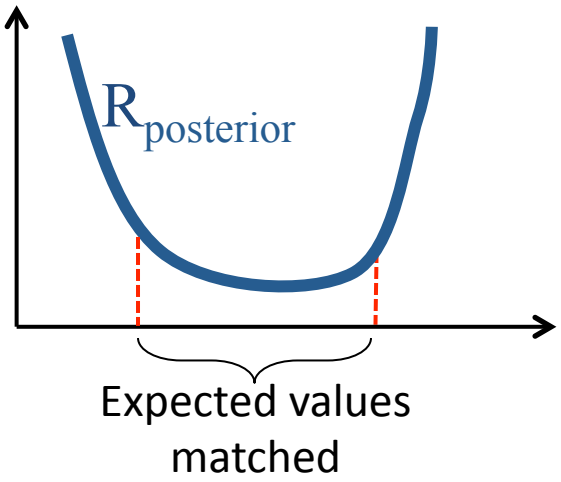


$$R_{laplacian}(\mathbf{q}) = \Sigma_y \Sigma_{i,j}\ w_{ij}\ (q_i(y) - q_j(y))^2$$

(**q** = Set of label distributions, one per training example.)

Label Regularizer: Posterior Expectations

- Labeler reveals expected values $\mathbf{b}$ of features $\mathbf{f}$ under true posterior distribution.



$$R_{posterior}(\mathbf{q}) = (E_{\mathbf{x},\mathbf{q}}[\mathbf{f}] - \mathbf{b})^2$$

($\mathbf{q}$ = Set of label distributions, one per training example.)

- Label regularizers can also be combined.

- For example, labeler might reveal partial label sets and similarity information:

$$R_{\text{partial}}(\mathbf{q}) + R_{\text{laplacian}}(\mathbf{q})$$

- We use this type of label regularizer in our experiments.

Upper Bound

- Let $(\mathbf{x}, \mathbf{y}) \sim \mathrm{D}$ be the training set of m examples.
- Let $\mathrm{L}(\theta)$ be the loss of parameter θ (e.g., hinge, log).
- Let R be the label regularizer.

- With probability $1 - \delta$:

$$\mathrm{E}_{\mathrm{D}}[\mathrm{L}(\theta)] \leq \max_{\mathbf{q}} (\mathrm{E}_{\mathbf{x},\mathbf{q}}[\mathrm{L}(\theta)] - \mathrm{R}(\mathbf{q})) + \mathrm{R}(\mathbf{y}) + \mathrm{O}\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)$$

- $(\mathbf{x}, \mathbf{y})$ = Training set (size m)
- D = Distribution
- $\mathrm{L}(\theta)$ = Loss
- R = Label regularizer

Upper Bound

- With probability $1 - \delta$:

$$E_D[L(\theta)] \leq \max_{\mathbf{q}} \left(E_{\mathbf{x},\mathbf{q}}[L(\theta)] - R(\mathbf{q})\right) + R(\mathbf{y}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)$$

- $(\mathbf{x}, \mathbf{y})$ = Training set (size m)
- D = Distribution
- $L(\theta)$ = Loss
- R = Label regularizer

Upper Bound

- With probability $1 - \delta$:

$$\underbrace{E_D[L(\theta)]}_{\text{Expected loss}} \leq \underbrace{\max_{\mathbf{q}} \left(E_{\mathbf{x},\mathbf{q}}[L(\theta)] - R(\mathbf{q})\right)}_{\#1} + \underbrace{R(\mathbf{y})}_{\#2} + \underbrace{O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)}_{\#3}$$

- $(\mathbf{x}, \mathbf{y})$ = Training set (size m)
- D = Distribution
- $L(\theta)$ = Loss
- R = Label regularizer

Upper Bound

- With probability $1 - \delta$:

$$E_D[L(\theta)] \leq \underbrace{\max_{\mathbf{q}} \left(E_{\mathbf{x},\mathbf{q}}[L(\theta)] - R(\mathbf{q})\right)} + R(\mathbf{y}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)$$

#1: Large when R is ambiguous (has many minima)

- $(\mathbf{x}, \mathbf{y})$ = Training set (size m)
- D = Distribution
- $L(\theta)$ = Loss
- R = Label regularizer

Upper Bound

- With probability $1 - \delta$:

$$E_D[L(\theta)] \leq \max_{\mathbf{q}} \left(E_{\mathbf{x},\mathbf{q}}[L(\theta)] - R(\mathbf{q})\right) + R(\mathbf{y}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)$$

#2: Large when R is misleading (penalizes true labeling)

- $(\mathbf{x}, \mathbf{y})$ = Training set (size m)
- D = Distribution
- $L(\theta)$ = Loss
- R = Label regularizer

Upper Bound

- With probability $1 - \delta$:

$$E_D[L(\theta)] \leq \max_{\mathbf{q}} \left(E_{\mathbf{x},\mathbf{q}}[L(\theta)] - R(\mathbf{q})\right) + R(\mathbf{y}) + \underbrace{O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)}$$

#3: Uniform convergence

- $(\mathbf{x}, \mathbf{y})$ = Training set (size m)
- D = Distribution
- $L(\theta)$ = Loss
- R = Label regularizer

Lower Bound

- Nearly matching lower bound*, except for a gap:

$$R(\mathbf{y}) - \min_{\mathbf{y}'} R(\mathbf{y}')$$

  i.e., large gap when R is misleading.

- ∴ Bounds loose in presence of malicious label noise.

* Technical assumptions required; paper has details.

GAME Algorithm

- Idea: Minimize our upper bound while controlling norm of θ:

$$\theta^* = \arg\min_\theta \max_q \left(\text{Upper Bound}(\mathbb{E}_{x,q}[L(x)] - R(q)) \right) + \alpha\|\theta\|^2$$

Upper bound
(discarding terms independent of θ)

GAME Algorithm

- Idea: Minimize our upper bound while controlling norm of θ:

$$\theta^* = \arg\min_\theta \max_{\mathbf{q}} \underbrace{(E_{\mathbf{x},\mathbf{q}}[L(\theta)] - R(\mathbf{q}))}_{\text{Upper bound}} + \alpha\|\theta\|^2$$

Upper bound
(discarding terms independent of θ)

- GAME is a two-step algorithm for finding θ^*.
- GAME assumes L and R are convex.
- For simplicity, assume L = log loss

GAME Algorithm

- Step 1. Compute "pessimistic" label distributions $\mathbf{q}^*$:

$$\mathbf{q}^* = \arg\max_{\mathbf{q}} \min_{\theta} \mathrm{E}_{\mathbf{x},\mathbf{q}}[\mathrm{L}(\theta, \mathbf{f})] + \mathrm{R}(\mathbf{q})$$

(Swap min and max; OK since L and R are convex.) (Take dual of inner minimization)

- For L = log loss, we can find $\mathbf{q}^*$ by an exponentiated gradient method.

GAME Algorithm

- <u>Step 2.</u> Find best parameter θ^* for pessimistic label distributions $\mathbf{q}^*$:

$$\theta^* = \arg\min_\theta \mathrm{E}_{\mathbf{x},\mathbf{q}^*}[\mathrm{L}(\theta)] + ||\theta||^2$$

GAME Algorithm

- <u>Step 2.</u> Find best parameter θ^* for pessimistic label distributions $\mathbf{q}^*$:

$$\theta^* = \arg\min_\theta \mathrm{E}_{\mathbf{x},\mathbf{q}^*}[\mathrm{L}(\theta)] + \|\theta\|^2$$

- For L = log loss, this is MLE with respect to $\mathbf{x}$, $\mathbf{q}^*$.

GAME Algorithm

- Step 2. Find best parameter θ^* for pessimistic label distributions $\mathbf{q}^*$:

$$\theta^* = \arg\min_\theta \mathrm{E}_{\mathbf{x},\mathbf{q}^*}[\mathrm{L}(\theta)] + \|\theta\|^2$$

- For L = log loss, this is MLE with respect to $\mathbf{x}$, $\mathbf{q}^*$.

- Note: GAME algorithm efficiently finds global optimum of objective.

Experiments

- Paid labelers (e.g. Amazon MTurk) can be erratic, malign.
  - Sheng et al 2008, Welinder et al 2010.

- We test SSL algorithms on data labeled by simulated malign labelers.

Experiments

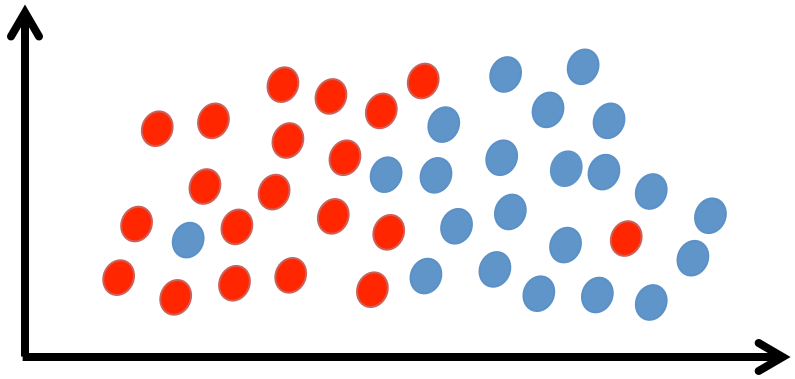
- Binary classification task:
  - GAME algorithm
  - Laplacian SVM [Belkin et al 04]
  - Transductive SVM [Joachims 99]

- Multiclass classification task:
  - GAME algorithm
  - Discriminitive EM [Jin & Gharamani 03]
  - Naïve maximum likelihood [Jin & Gharamani 03]

- GAME uses:
  - Label regularizer $\mathrm{R} = \mathrm{R}_{\mathrm{partial}} + \mathrm{R}_{\mathrm{laplacian}}$ (convex).
  - Loss function L = Log loss (but we report accuracy).

Binary Classification Task: Unhelpful Labeler

- Labels outliers first.

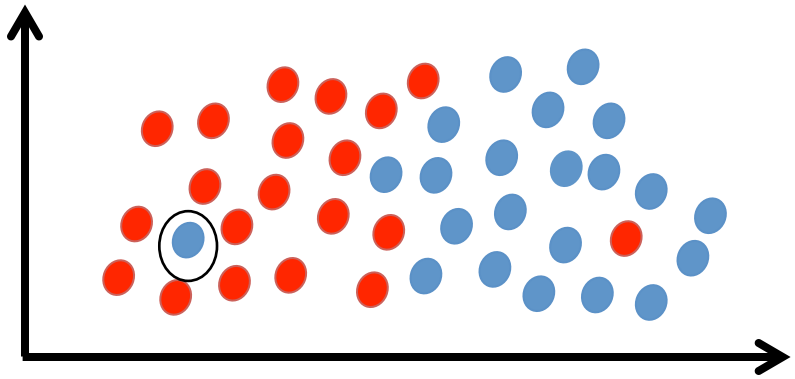


Labels Requested
0

- Circled = Labeled

Binary Classification Task: Unhelpful Labeler

- Labels outliers first.

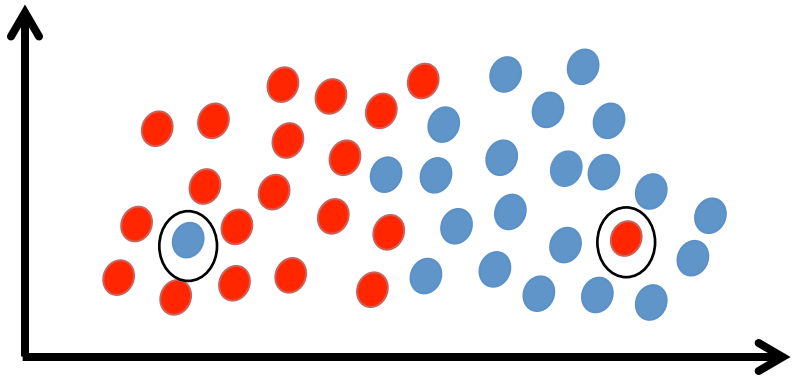


Labels Requested
1

- Circled = Labeled

Binary Classification Task: Unhelpful Labeler

- Labels outliers first.

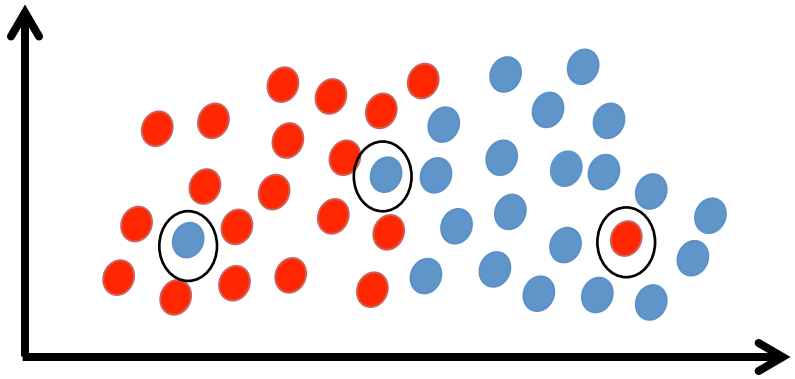


Labels Requested
2

- Circled = Labeled

Binary Classification Task: Unhelpful Labeler

- Labels outliers first.



Labels Requested

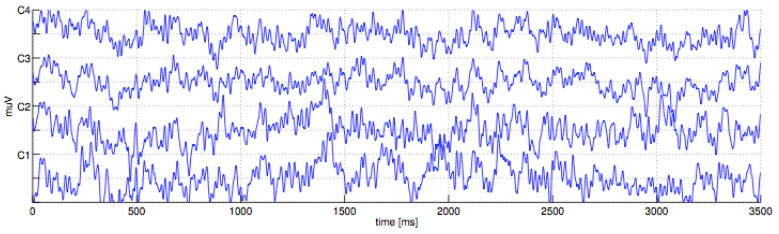
3

- Circled = Labeled

Binary Classification Task: Datasets

- Columbia Object Image Library [Nene et al 96]
  - Images of 24 types of household objects.
  - Objects divided into two classes.
  - Features are pixel values.

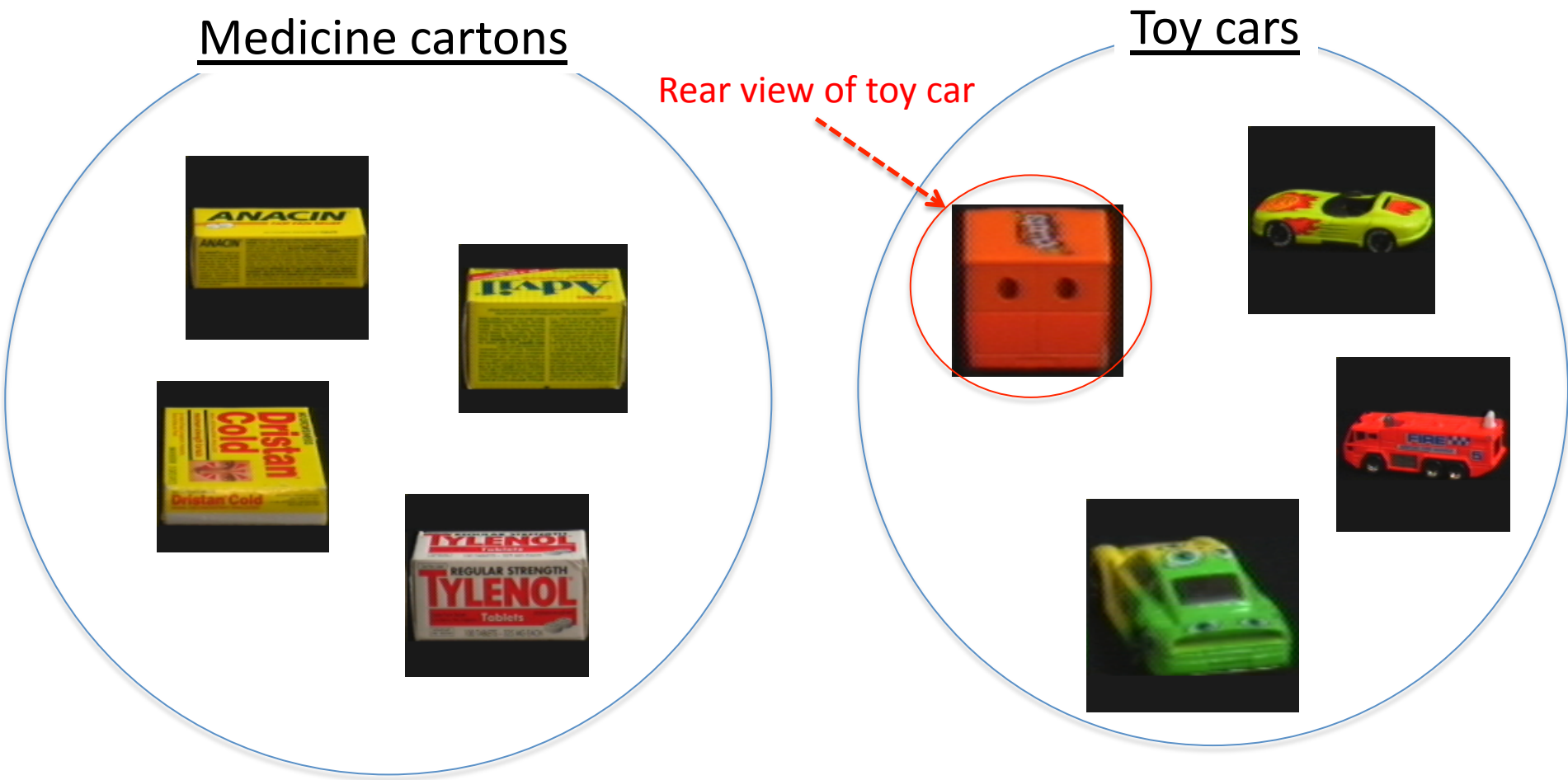
- Brain-computer interface [Lal et al 04]
  - EEG brain scans of single human subject.
  - Classes are: thinking “left” or “right”
  - Features are smoothed electrode values.



- Both datasets part of standard SSL benchmark [Chappelle et al 06].

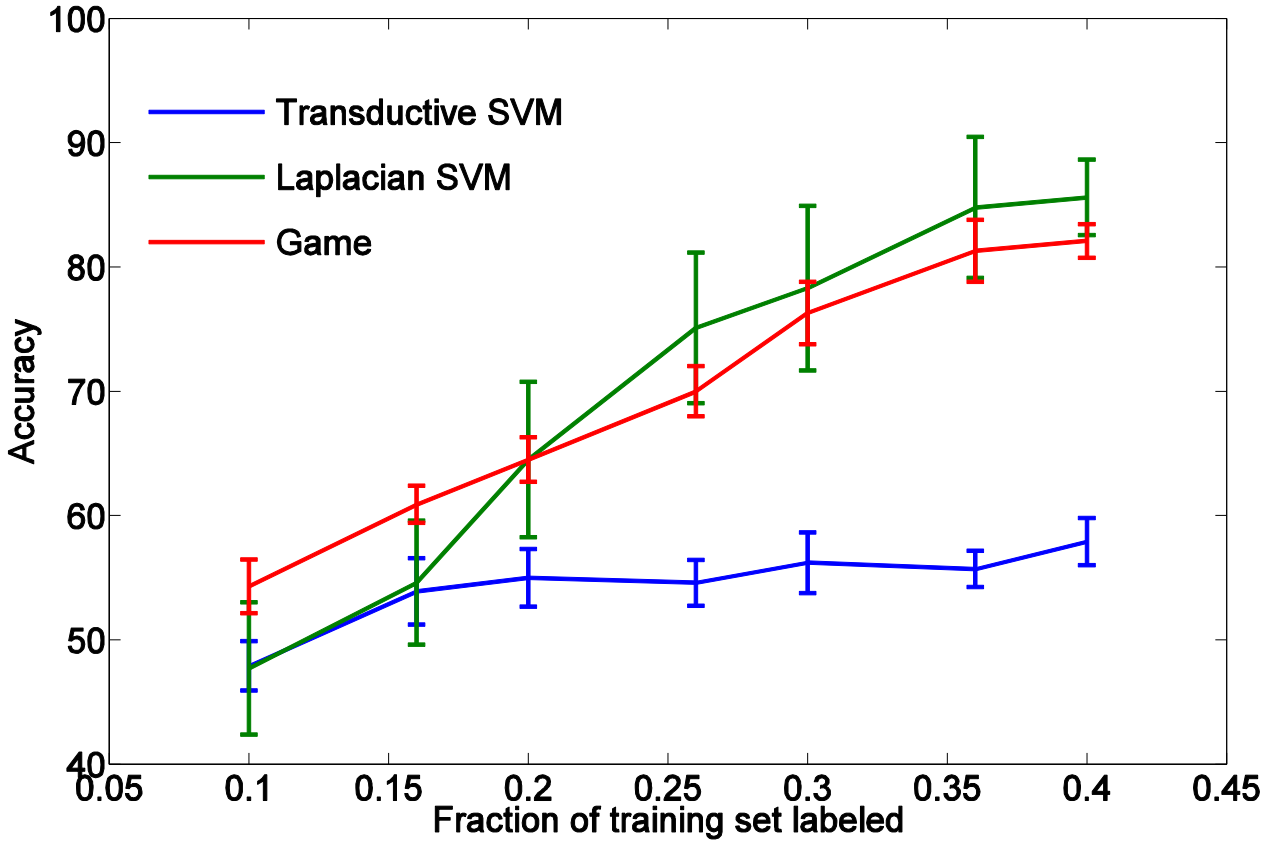
Binary Classification Task: Datasets

- First outlier labeled in image library:



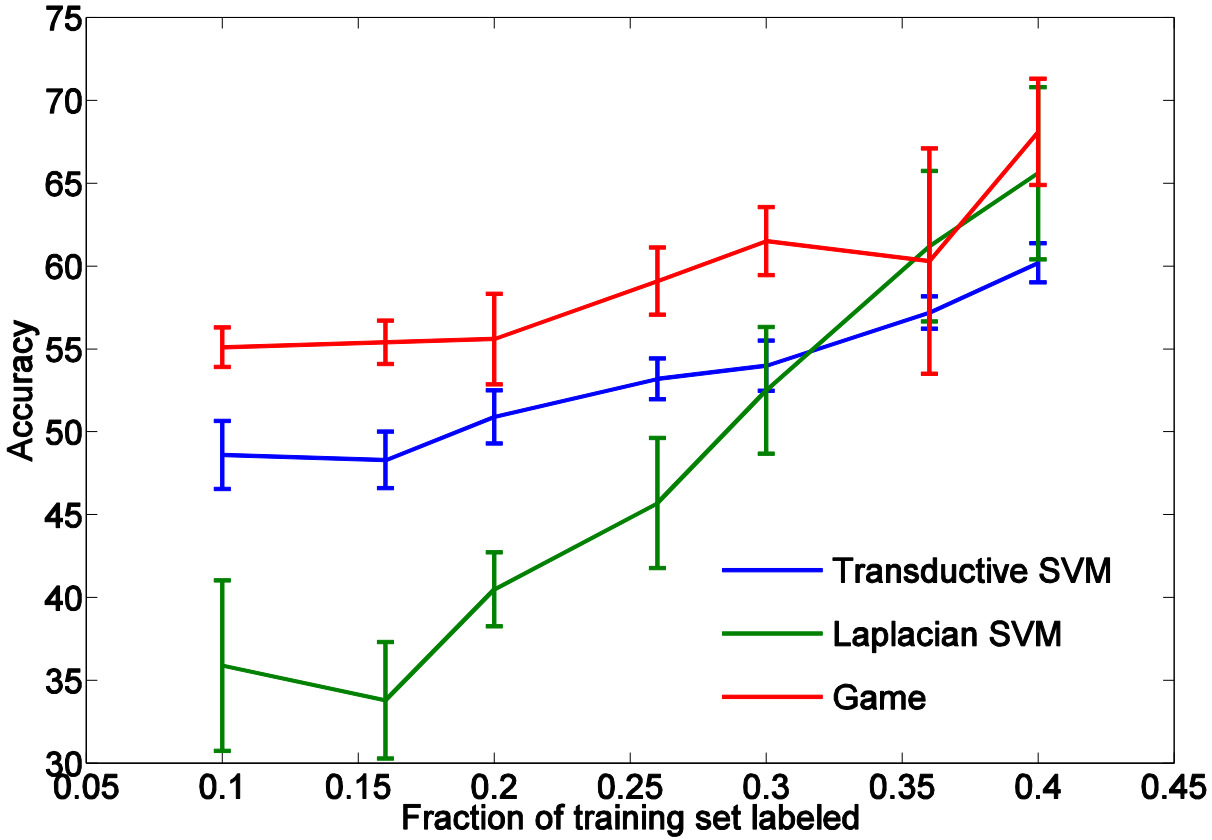
Binary Classification Task: Results

- Columbia Object Image Library dataset:



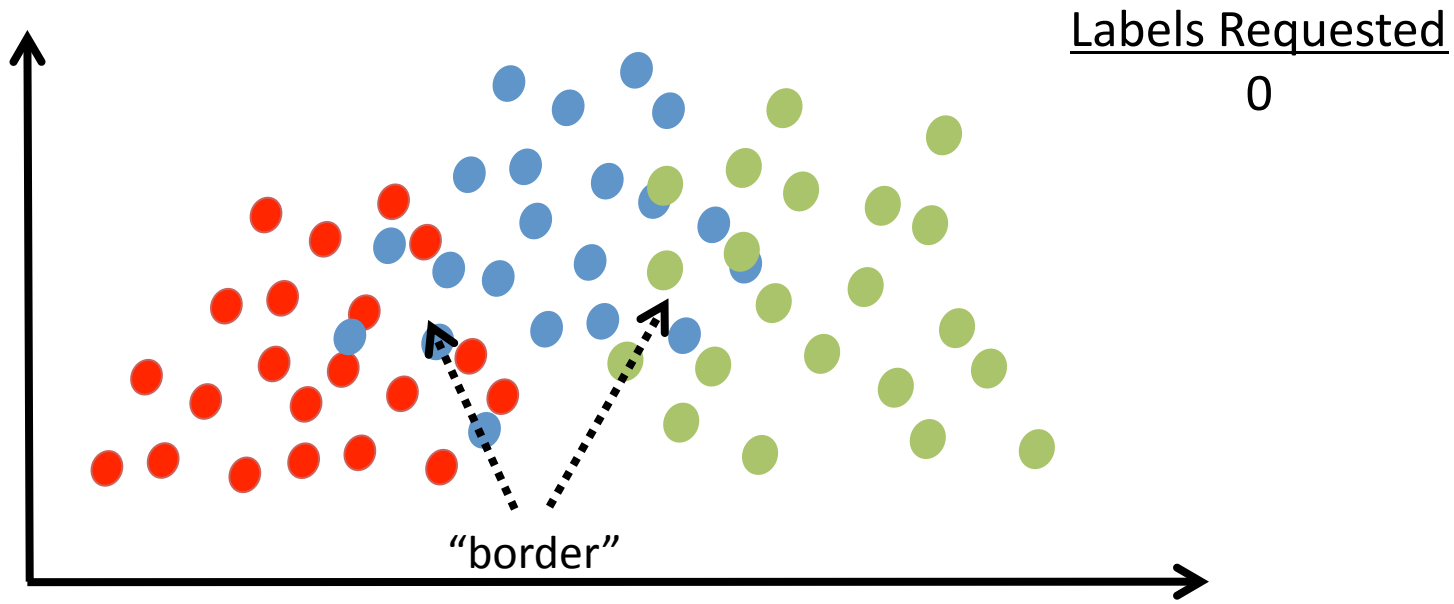
Binary Classification Task: Results

- Brain-computer interface data set:



Multiclass Classification Task: Lazy Labeler

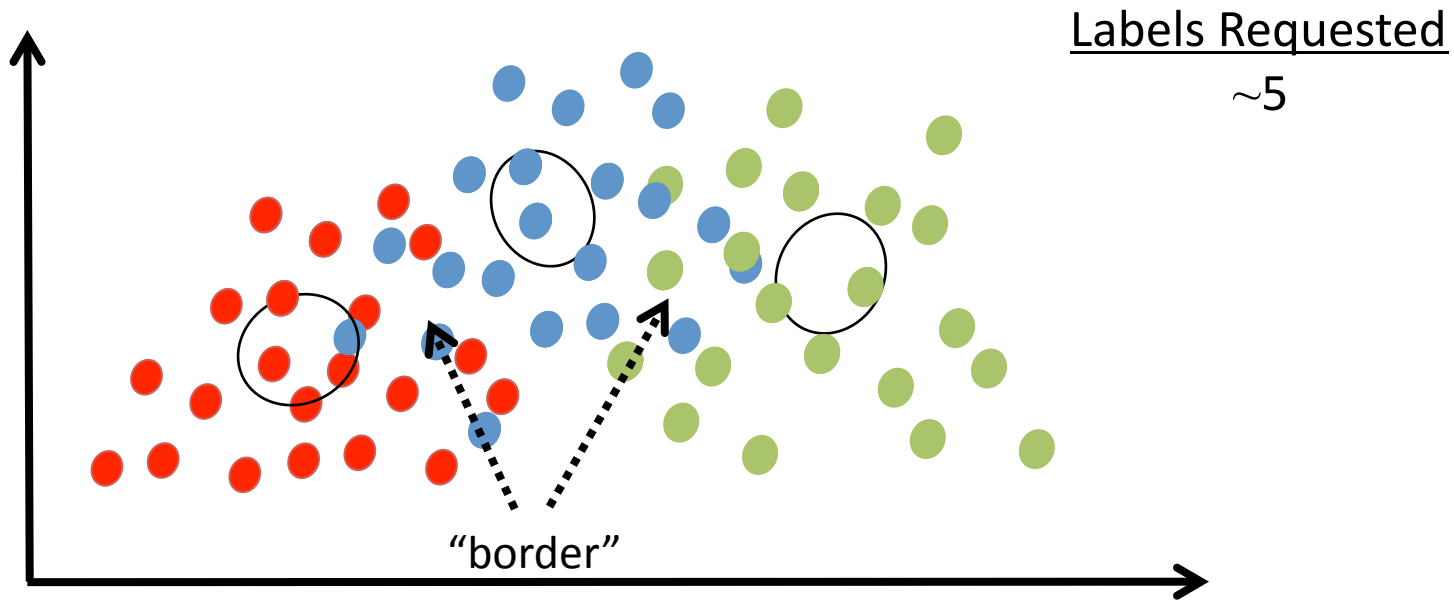
- Labels “border” examples last.



- Circled = Labeled

Multiclass Classification Task: Lazy Labeler

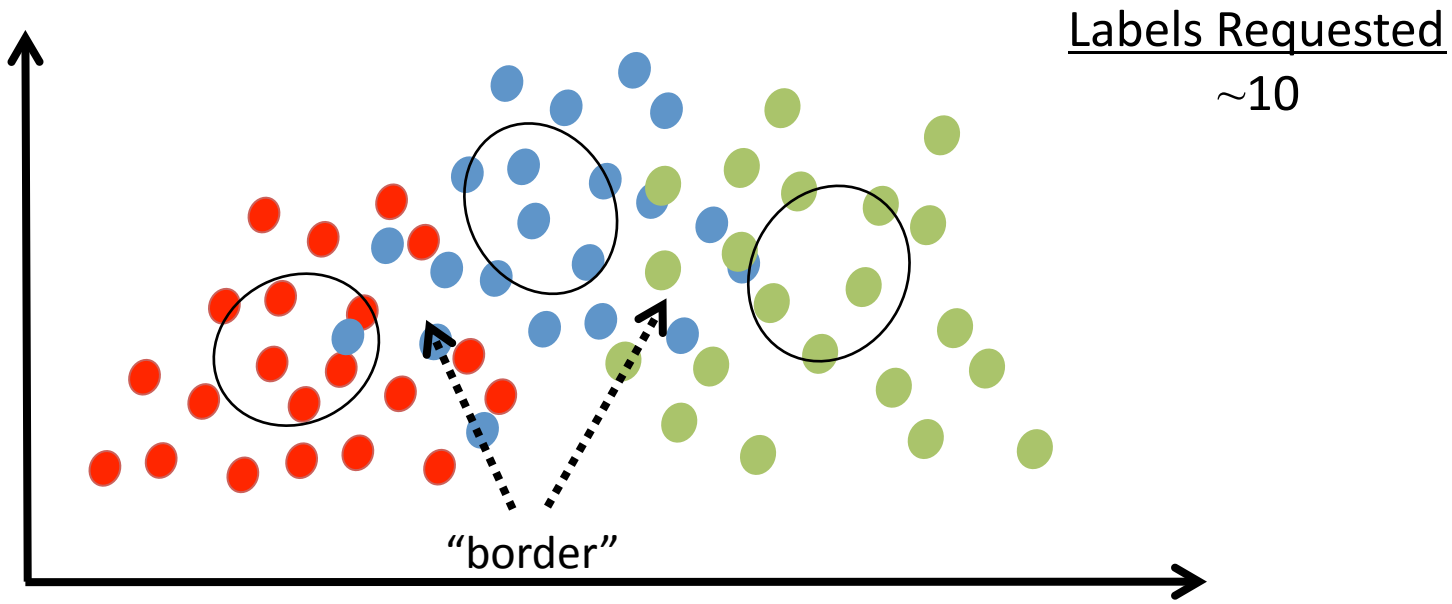
- Labels “border” examples last.



- Circled = Labeled

Multiclass Classification Task: Lazy Labeler

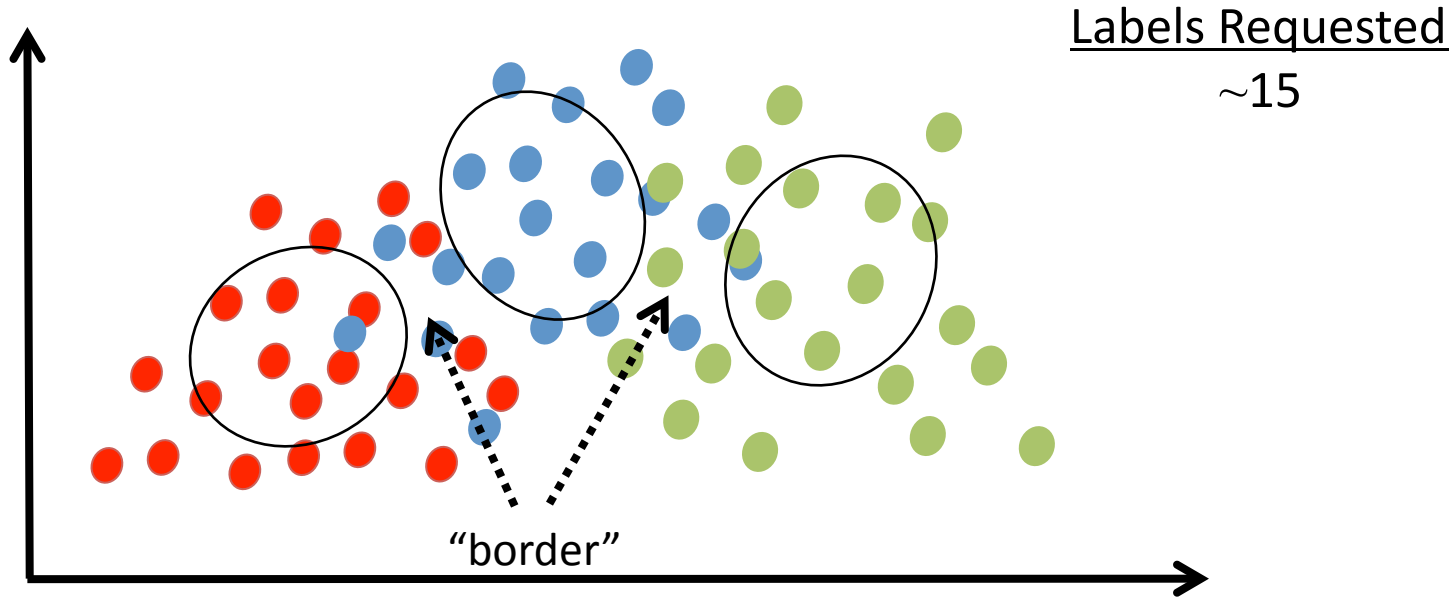
- Labels “border” examples last.



- Circled = Labeled

Multiclass Classification Task: Lazy Labeler

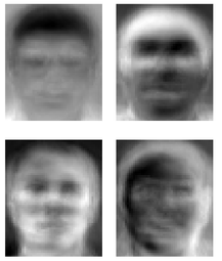
- Labels “border” examples last.

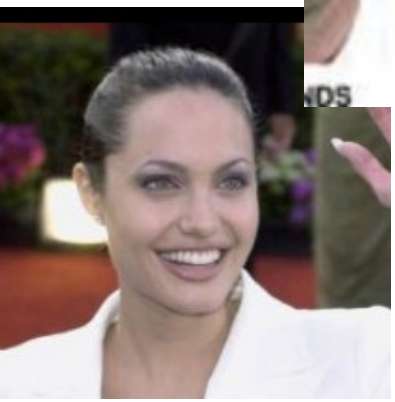


- Circled = Labeled

Multiclass Classification Task: Dataset

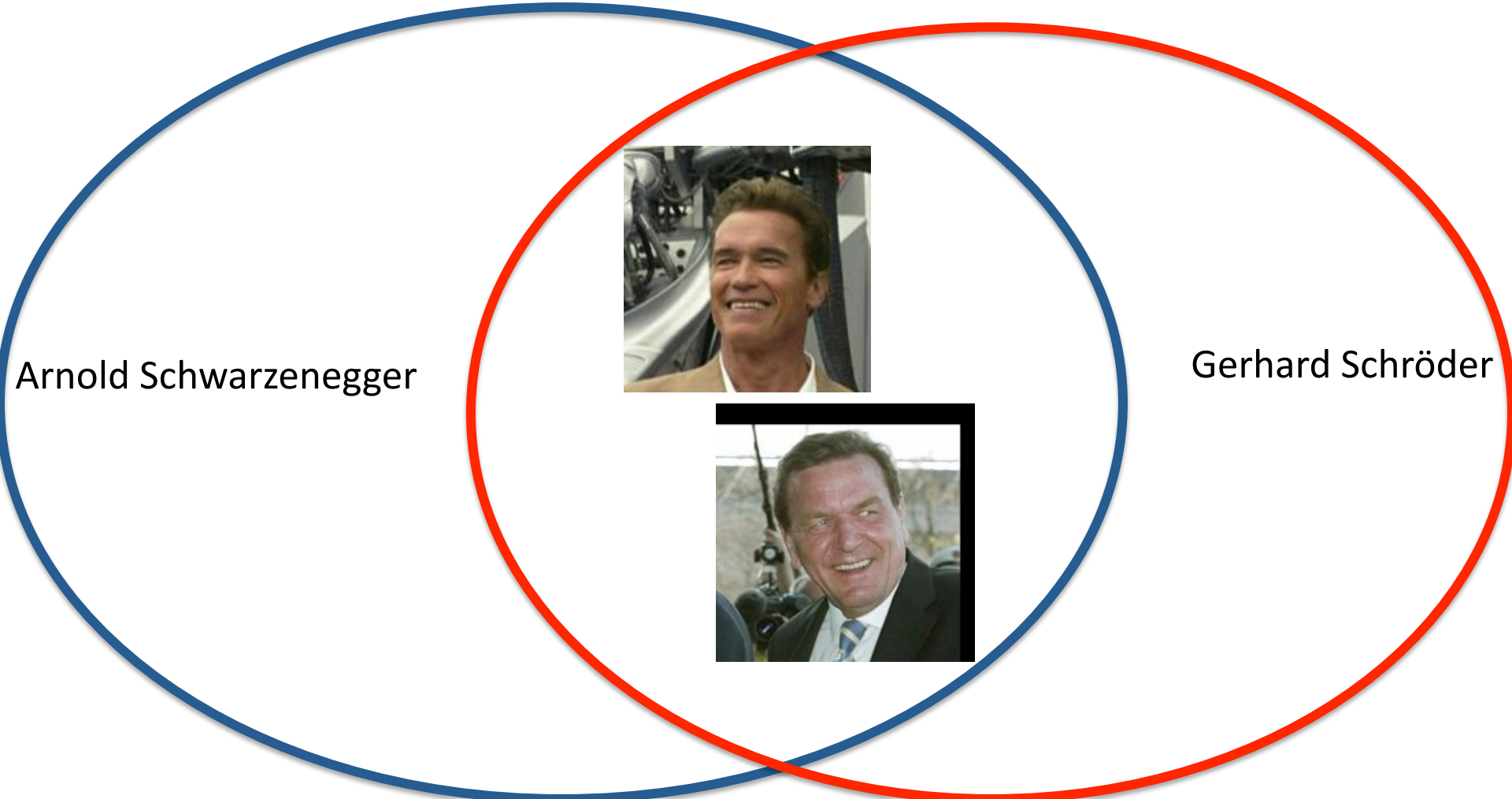
- Labeled Faces-In-The-Wild [Huang et al 07]
  - Face photographs of public figures.
  - We used subset of 446 phots of 10 most common people.
  - Features: Top 50 principal components (eigenfaces)





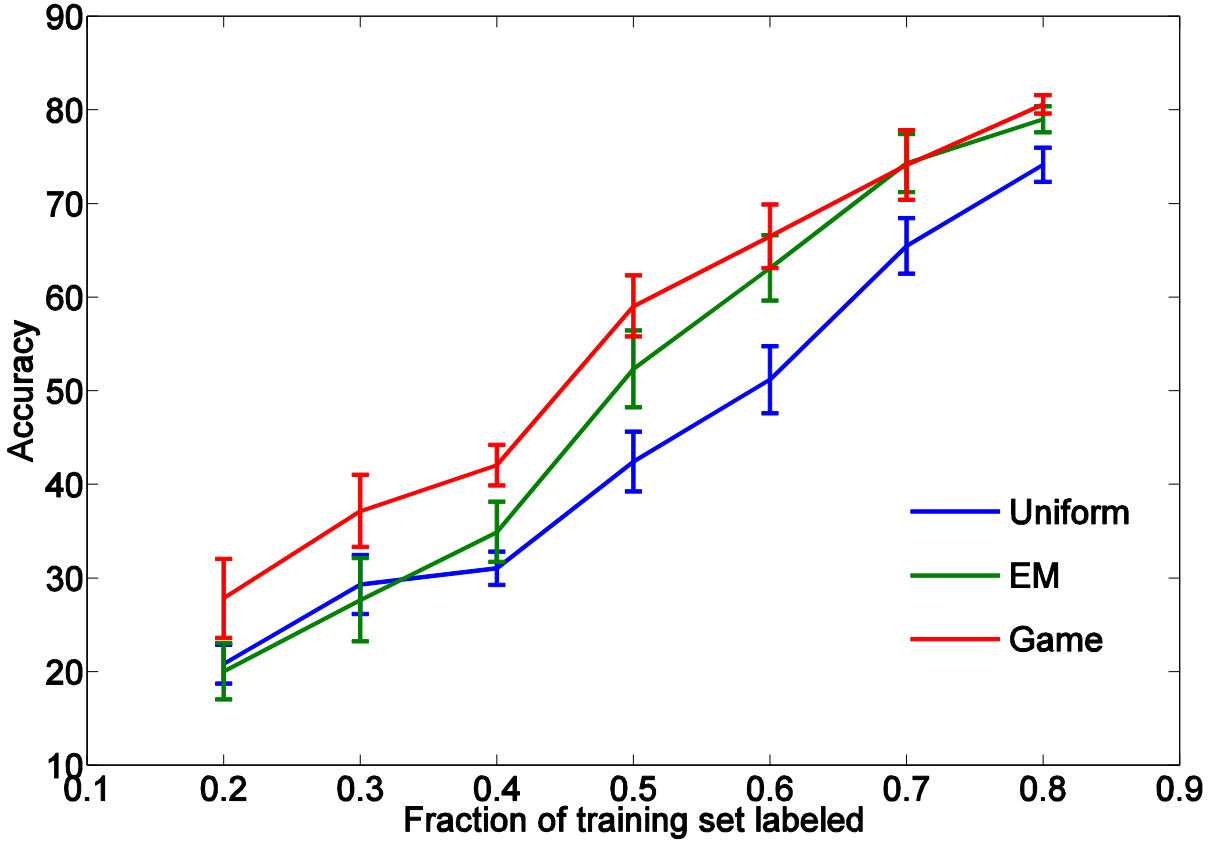
Multiclass Classification Task: Dataset

- Classes with largest overlap in Faces dataset:



Multiclass Classification Task: Results

- Labeled Faces-In-The-Wild:



Conclusion and Future Work

- Adversarial semi-supervised learning framework.
- Algorithm that is theoretically and experimentally robust to adversarially missing label information.

- Is this a good model of actual labelers?
- Extension to structured prediction?
- Active learning?

Thanks!