Semi-Supervised Learning with Adversarially Missing Label Information

Umar Syed and Ben Taskar

NIPS 2010
• Labeler examines training data ...
• Labeler examines training data ... reveals some labels to learning algorithm.
• Labeler examines training data ... reveals some labels to learning algorithm.

• Typical assumption: Labeled examples chosen randomly.
• “Naturally occurring” labeling is **not** random.
• “Naturally occurring” labeling is not random.
• “Naturally occurring” labeling is **not** random.
Our Framework

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

• Allows for local and global label information.
• **Idea**: Labeler examines training data \((x, y)\) ...
• **Idea:** Labeler examines training data \((x, y)\) ... reveals *label regularizer* function \(R\).
• **Idea:** Labeler examines training data \((x, y)\) ... reveals *label regularizer* function \(R\).

- \(R\) chosen arbitrarily from a known function family.
- \(R\) encodes all label information about training data.
• $R(y) \approx \text{large} \Rightarrow y \underline{\text{likely not}} \text{ true labeling of } x.$

• $R(y) \approx \text{small} \Rightarrow y \underline{\text{may be}} \text{ true labeling of } x.$
Overview of our Framework

Labeler

(x, y)

Distribution

(x, R)

Learner
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 $x_i$:

$$R_{\text{partial}}(y) = \begin{cases} 0 & \text{if } y_i \in Y_i \text{ for all } i, \\ \infty & \text{otherwise.} \end{cases}$$
**Label Regularizer: Graph Laplacian**

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

\[
R_{\text{laplacian}}(q) = \sum_y \sum_{i,j} w_{ij} (q_i(y) - q_j(y))^2
\]

($q =$ Set of label distributions, one per training example.)
Label Regularizer: Posterior Expectations

- Labeler reveals expected values $b$ of features $f$ under true posterior distribution.

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

($q$ = 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:

\[ R_{\text{partial}}(q) + R_{\text{laplacian}}(q) \]

• We use this type of label regularizer in our experiments.
Upper Bound

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

- With probability \(1 - \delta\):

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

- \((x, y) = \) Training set (size \(m\))
- \(D = \) Distribution
- \(L(\theta) = \) Loss
- \(R = \) Label regularizer
Upper Bound

- With probability $1 - \delta$:

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

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

- With probability $1 - \delta$:

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

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

• With probability $1 - \delta$:

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

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

• $(x, y) = \text{Training set (size } m\text{)}$
• $D = \text{Distribution}$
• $L(\theta) = \text{Loss}$
• $R = \text{Label regularizer}$
Upper Bound

• With probability $1 - \delta$:

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

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

• $(x, y) = \text{Training set (size m)}$
• $D = \text{Distribution}$
• $L(\theta) = \text{Loss}$
• $R = \text{Label regularizer}$
Upper Bound

- With probability $1 - \delta$:

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

#3: Uniform convergence

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

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

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

\[ R(y) - \min_y R(y') \]

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 $\theta$:

$$\theta^* = \arg\min_{\theta} \max_q \left( \mathbb{E}_{q|\theta} [r(B)] + \alpha \|\theta\|^2 \right)$$

( discarding terms independent of $\theta$)
GAME Algorithm

• **Idea**: Minimize our upper bound while controlling norm of $\theta$:

\[
\theta^* = \arg \min_\theta \max_q (E_{x,q}[L(\theta)] - R(q)) + \alpha \|\theta\|^2
\]

(upper bound (discarding terms independent of $\theta$))

• GAME is a two-step algorithm for finding $\theta^*$.
• GAME assumes $L$ and $R$ are convex.
• For simplicity, assume $L = \text{log loss}$
GAME Algorithm

• **Step 1.** Compute “pessimistic” label distributions $q^*$:

$$q^* = \arg \max_{p,q} \min_{\theta} \max_{p,q} \left[ L(\theta) \left\| \mathbb{E}_x [f] - \mathbb{E}_{x,q} [f] \right\|^2 - R(q) \right]$$

(Swap min and max; OK since $L$ and inner minimization)

• For $L = \text{log loss}$, we can find $q^*$ by an **exponentiated gradient method**.
GAME Algorithm

- **Step 2.** Find best parameter $\theta^*$ for pessimistic label distributions $q^*$:

$$\theta^* = \arg \min_{\theta} E_{x, q^*}[L(\theta)] + ||\theta||^2$$
GAME Algorithm

• **Step 2.** Find best parameter $\theta^*$ for pessimistic label distributions $q^*$:

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

• For $L = \text{log loss}$, this is MLE with respect to $x$, $q^*$. 
GAME Algorithm

- **Step 2.** Find best parameter $\theta^*$ for pessimistic label distributions $q^*$:

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

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

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

• Paid labelers (e.g. Amazon MTurk) can be erratic, malign.

• 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
  – Discriminative EM [Jin & Gharamani 03]
  – Naïve maximum likelihood [Jin & Gharamani 03]

• GAME uses:
  – Label regularizer \( R = R_{\text{partial}} + R_{\text{laplacian}} \) (convex).
  – Loss function \( L = \log \text{loss} \) (but we report accuracy).
Binary Classification Task: Unhelpful Labeler

• Labels outliers first.

• Circled = Labeled

Labels Requested
0
Binary Classification Task: Unhelpful Labeler

• Labels outliers first.

• Circled = Labeled

Labels Requested
1
Binary Classification Task: Unhelpful Labeler

- Labels outliers first.

• Circled = Labeled
Binary Classification Task: Unhelpful Labeler

• Labels outliers first.

• 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 [Chapelle et al 06].
Binary Classification Task: Datasets

- First outlier labeled in image library:

  **Medicine cartons**
  - [Image of medicine cartons]

  **Toy cars**
  - [Image of toy cars]
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

Labels Requested
~15
Multiclass Classification Task: Dataset

- Labeled Faces-In-The-Wild [Huang et al 07]
  - Face photographs of public figures.
  - We used subset of 446 photos 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!