Efficient Semi-supervised and Active Learning of Disjunctions

Maria-Florina Balcan, Christopher Berlind, Steven Ehrlich, and Yingyu Liang

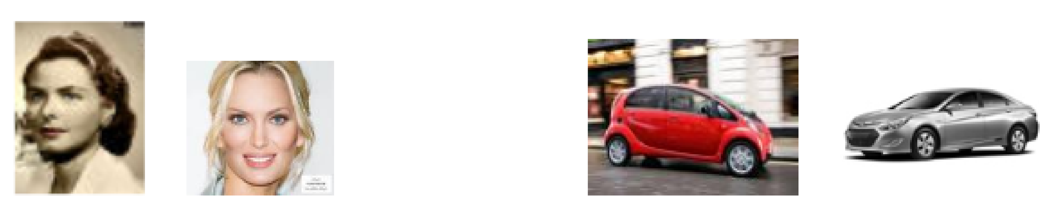
Georgia Tech College of Computing

Taking Advantage of Unlabeled Data

Classic paradigm: passive supervised learning

- Given labeled examples

faces cars



- Find function that correctly labels examples

Classic paradigm insufficient nowadays

- Massive amounts of unlabeled data
- Only small fraction can be labeled

protein sequences astronomical data social networks

Semi-supervised learning: Directly given both labeled examples and unlabeled examples.

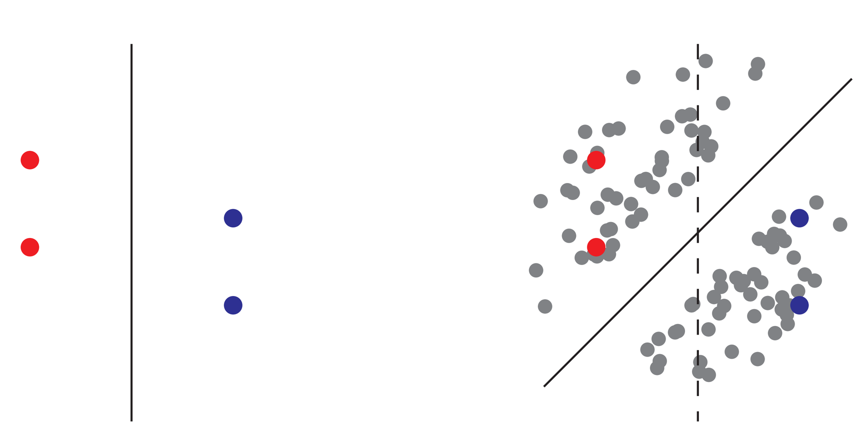
Active learning: Given unlabeled examples and ability to query the label of any unlabeled example.

Semi-supervised PAC Model [Balcan & Blum, 2010]

- Compatibility function χ relates hypotheses to unlabeled data

All hypotheses need $O(\frac{1}{\epsilon}\log|C|)$ labels

Highly compatible hypotheses need $O(\frac{1}{\epsilon}\log|C_{D,\chi}(\epsilon)|)$ labels

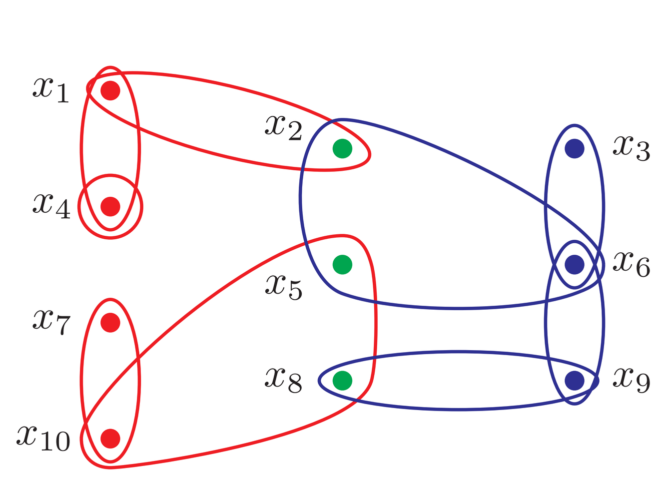


Labeled data only Labeled and unlabeled

- Fewer labels in principle than passive supervised
- Lack efficient algorithms to realize this potential

Two-sided Disjunctions [Blum & Balcan, 2007]

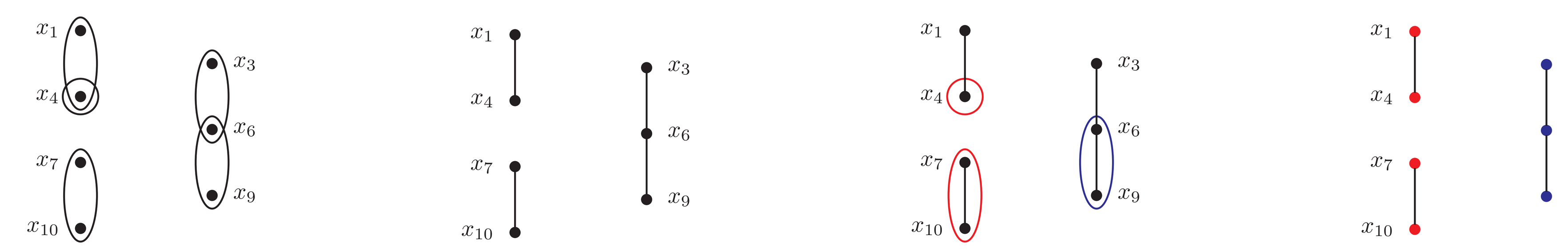
- Examples described by n boolean features
  - positive, negative, and non-indicators
- Examples labeled by contained indicators
- Compatibility:
  - Every example has an indicator
  - No example has conflicting indicators



A Simple Case: No Non-indicators

Algorithm:

1. Build the commonality graph by connecting variables that appear in examples together
2. Query one example from each connected component



Active Learning Algorithm

Key idea: Find and remove all k non-indicators and reduce to the previous case.

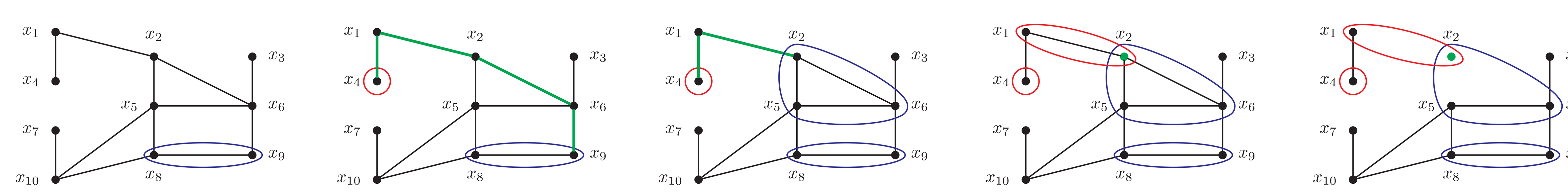
Algorithm:

1. Build the commonality graph
2. Query an example in each component
3. Test the hypothesis on a small sample
4. If not consistent, find non-indicator via binary search

Analysis:

- One query for each of $\log|C_{D,\chi}(\epsilon)|$ components
- One test/search for each of k non-indicators
  - $\frac{1}{\epsilon}\log\frac{k}{\delta}$ queries per test
  - $\log n$ queries per binary search

Queries: $O\left(\log|C_{D,\chi}(\epsilon)| + k\left(\log n + \frac{1}{\epsilon}\log\frac{k}{\delta}\right)\right)$



Semi-supervised Learning Algorithms

Key idea: With enough labeled data, every non-indicator appears in some labeled example.

Parameter: ϵ_0 = minimum non-indicator probability

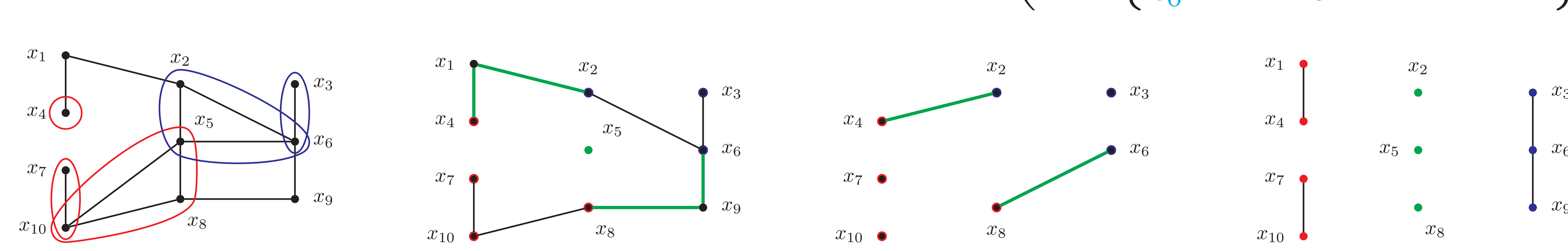
Algorithm 1:

1. Build the commonality graph
2. Assign variables to potential indicator sets
3. Build indicator graph from paths between opposite potential indicators
4. Find vertex cover (non-indicators) corresponding to a consistent and compatible hypothesis

Analysis:

- Need $\frac{1}{\epsilon_0}\log k$ labels to satisfy key idea above
- Target non-indicators form VC in indicator graph
- Need $\frac{1}{\epsilon}\log|C_{D,\chi}(\epsilon)|$ labels for generalization
- Computationally efficient when $k = O(\log n)$
- Finds consistent and compatible disjunction

Labels: $\tilde{O}\left(\max\left\{\frac{1}{\epsilon_0}\log k, \frac{1}{\epsilon}\log|C_{D,\chi}(\epsilon)|\right\}\right)$



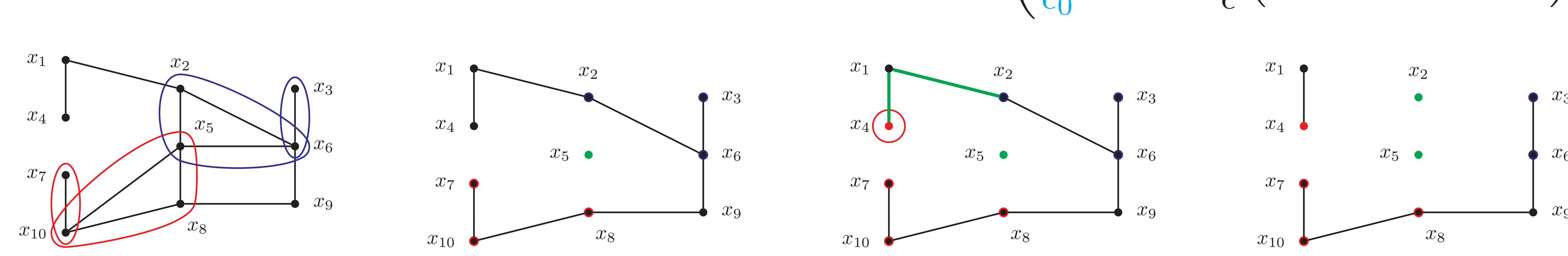
Algorithm 2:

1. Build the commonality graph
2. Assign variables to potential indicator sets
3. Test the nearest neighbor hypothesis
4. On each mistake, remove a non-indicator (or label a connected component)

Analysis:

- Need $\frac{1}{\epsilon_0}\log k$ labels to satisfy key idea above
- Mistakes reveal paths ending at a non-indicator
- At most $k + \log|C_{D,\chi}(\epsilon)|$ mistakes
- Computationally efficient
- Improper learner

Labels: $\tilde{O}\left(\frac{1}{\epsilon_0}\log k + \frac{1}{\epsilon}\left(k + \log|C_{D,\chi}(\epsilon)|\right)\right)$



Discussion

The power of active learning

- SSL poses computational challenges
- AL algo is efficient, proper, and less restrictive

Learning halfspaces with margins

- Problem has $L_\infty L_1$ margin $\frac{|w^*\cdot x|}{\|w^*\|_\infty\|x\|_1} \geq \frac{1}{k+1}$
- Margin differs from Perceptron and Winnow
- Main open question: are there efficient algorithms for the general problem?