Convex Adversarial Collective Classification

Ali Torkamani and Daniel Lowd

University of Oregon

Adversarial Collective Classification

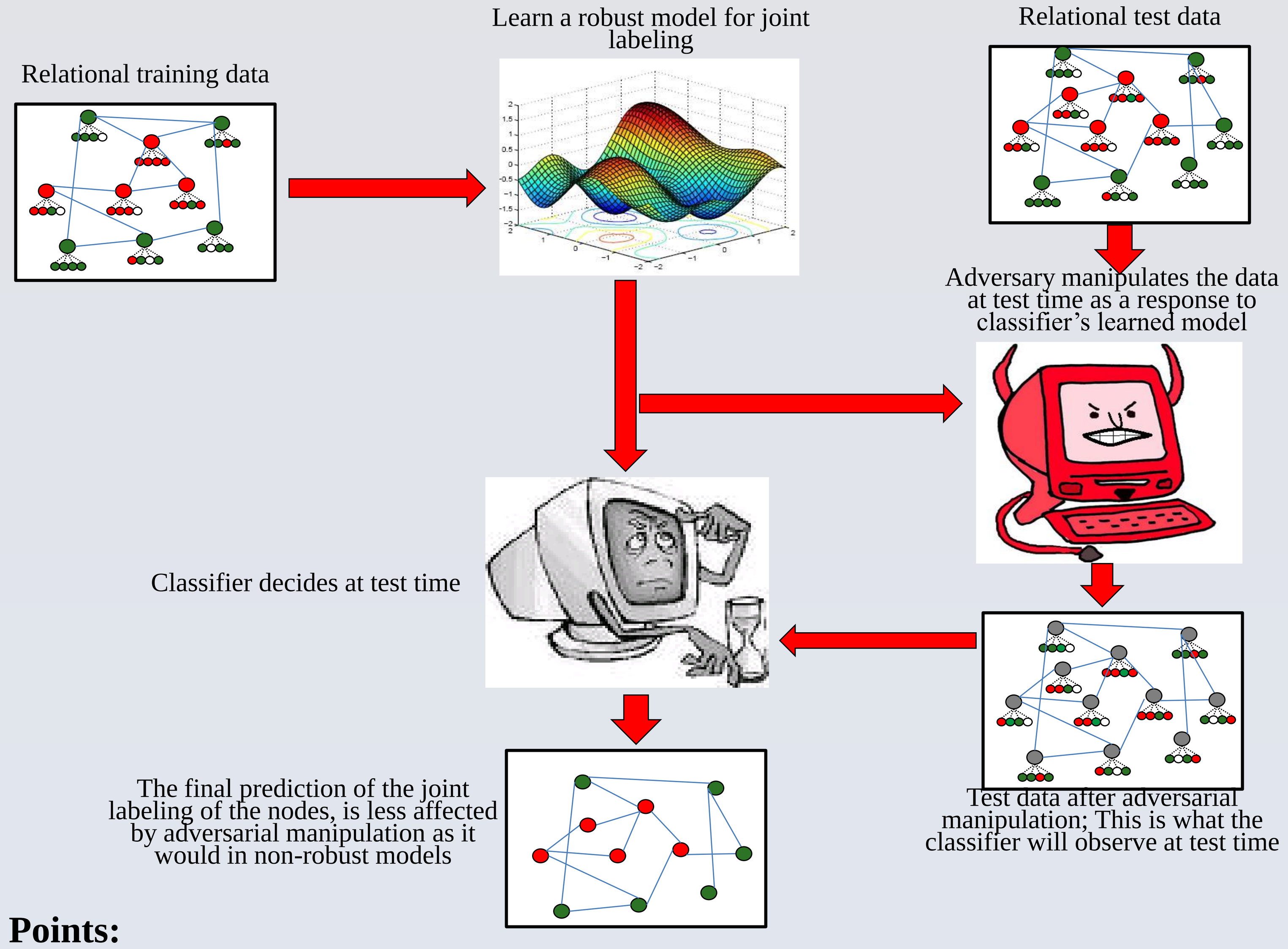
1

Goal: Learn to robustly label a set of related objects in the presence of adversarial manipulation.

Applications:

- **Adversarial Manipulation:** Collective classification problems in which the test data is manipulated by an active adversary to maximize the misclassification error. Examples: web-spam, counter-terrorism, auction fraud, etc.
- **Concept Drift:** Collective classification problems in which distribution of test data has diverged from the distribution of data at train time. For example, when classifying blogs, tweets, or news articles, the topics being discussed will vary over time.

Scenario:



Key Points:

- ***Relational Structure***: *Exploit both attributes and links.*
- ***Adversary Awareness:*** *Train a robust model against worst case adversarial manipulation of data at test time.*

Motivation and Overview

- Associative Markov networks [Taskar et al., 2004] allow polynomial-time learning and inference, but are not robust to malicious adversaries. Current work on adversarial machine learning are robust to rational or worst-case adversaries, but are limited to the case where labels of different objects were independent (e.g.,[Teo et al., 2008]).
- In this work, we develop **Convex Adversarial Collective Classification** (CACC), We have developed an efficient weight learning method for collective classification that is robust to malicious adversaries. Our method works by maximizing the margin between the true labeling and any alternate labeling, assuming a worst case manipulation of the features (up to some fixed budget). By taking the dual of the inner maximization, we can represent this as a single convex, quadratic program, which finds the optimal weights in polynomial time.

2

Notation

In the next sections of the poster we will use the following notation:

$\hat{x}$-True attribute values x-Adversarially modified attribute values D-Adversary's budget for maximum number of changes it can make on x to generate $\hat{x}$	$\hat{y}$-True object label related values y-Predicted label related values y (and $\hat{y}$) include both indicator variables for object labels y_i^k and dummy variables y_{ij}^k that are introduced to represent $\min(y_i^k, y_j^k)$	C - The margin regularization weight z_{ij}^k- The dummy variable that represents $\min(y_i^k, x_{ij}^k)$	W -The model weights; Similar to AMNs [Taskar et al., 2004], in our work, the score function should be linear in x, y, z_{ij}^k, and y_{ij}^k as well as in weights. We use w_j^k to refer to weights of the unary potential part of the score function, and w_e^k for its clique related weights.

3

Associative Markov Networks

- An associative Markov network (AMN) is a Markov network where linked nodes are more likely to have the same label. [Taskar et al., 2004]

Learner's Goal: Select w to maximize the margin between true labeling and alternate labeling:

$$\min_{w,\xi} \quad \frac{1}{2}\|w\|_F^2 + C\xi, \quad \text{s.t.} \quad \xi \geq \max_y [score(w,x,y) - score(w,x,\hat{y}) + \Delta(y,\hat{y})] \quad (1)$$

Where $\Delta(y,\hat{y})$ is the number of misclassified nodes.

- **Good News:** For score function being bilinear in w and y (i.e. $score(w,x,y) = w^T xy$), we can convert the non-convex bilevel mathematical program in (1) , to a convex standard QP, by substituting the inner maximization linear program with it's dual.
- **Efficient Inference**: The label prediction problem is formulated by a linear program. $\max_y \ q^T y \quad \text{s.t.} \quad y \geq 0; Ay \leq b;$
- Integral solution is guaranteed for binary valued labels.
- AMN's performance reduces in presence of an active adversary that alters the features at test time!

4

Convex Adversarial Collective Classification

- **Goal**: Learn to jointly predict the labels of the nodes in an AMN, while being aware of possible existence of active adversary at test time.

What can an adversary do?

- **Adversary's Weakness**:
- It has a budget D for the maximum number of features that it can change. For $\Delta(x,\hat{x})$ being the difference measure between the true features $\hat{x}$ and the features after adversarial manipulation x, we always have: $\Delta(x,\hat{x}) \leq D$

- **Adversary's Inference**:
- Given the parameters w, the adversary can choose x such that the alternate labeling receives a high score, making it hard for the classifier to predict the correct joint labels, plus getting a reward when the alternate labeling is more different from the true labeling. The adversary can achieve this by solving the following non-convex program:

$$max_{x,y} \quad score(x,y,w) - score(x,\hat{y},w) + \Delta(y,\hat{y}) \quad \text{s.t.} \quad \Delta(x,\hat{x}) \leq D \quad (2)$$

What can the learner do?

- The learner should be robust against rational adversaries; this can be achieved by introducing an adversarially constrained large margin SVM :

$$\min_{w,\xi} \quad \frac{1}{2}\|w\|_F^2 + C\xi, \quad \text{s.t.} \quad (3)$$

$$\xi \geq max_{x,y} \quad [score(x,y,w) - score(x,\hat{y},w) + \Delta(y,\hat{y})] \quad \text{s.t.} \quad \Delta(x,\hat{x}) \leq D$$

Can we solve them?

- With the score function being linear in each of the variables (i.e. $score(w,x,y) = w^T xy$), both of the programs (2) and (3) are non-convex.
- Since the program (2) is the same as the inner maximization in program (3), we can use the same trick for solving both of the problems. The procedure is as follows:

1. Convert the trilinear form in the score function that has both x and y to bilinear form, by introducing a dummy matrix variable z=xy.

   How and why it works:
   x_{ij} being the jth feature of the ith node and y_i^k being the indicator variable which is equal to 1 when the label of the ith node is k, otherwise zero, we introduce dummy variable z_{ij}^k to replace $x_{ij} * y_i^k$
   For binary valued x_{ij} , we will have $z_{ij}^k = min(x_{ij}, y_i^k)$

2. Add necessary linear constraints on z: We can encode the $z_{ij}^k = min(x_{ij}, y_i^k)$ constraint by adding two linear inequalities to the program.
3. Given w, the resulting formula will be linear in z, x and y; therefore adversary's problem is just a linear program over x,y and z.
4. By substituting the dual of the resulted linear program with the inner maximization in equation (2), the bilinearity will be removed and final program will become a convex standard quadratic program that can be solved efficiently.

- Theorem: Equation (2), has an integral solution for binary valued x and y.

5

Classifier and Adversary's inference

- Both the classifier and adversary's inference problems are linear programs.
- Equation (2) is the adversary's linear program.
- Classifier's inference for predicting the joint labeling of nodes, is the same as in an AMN.

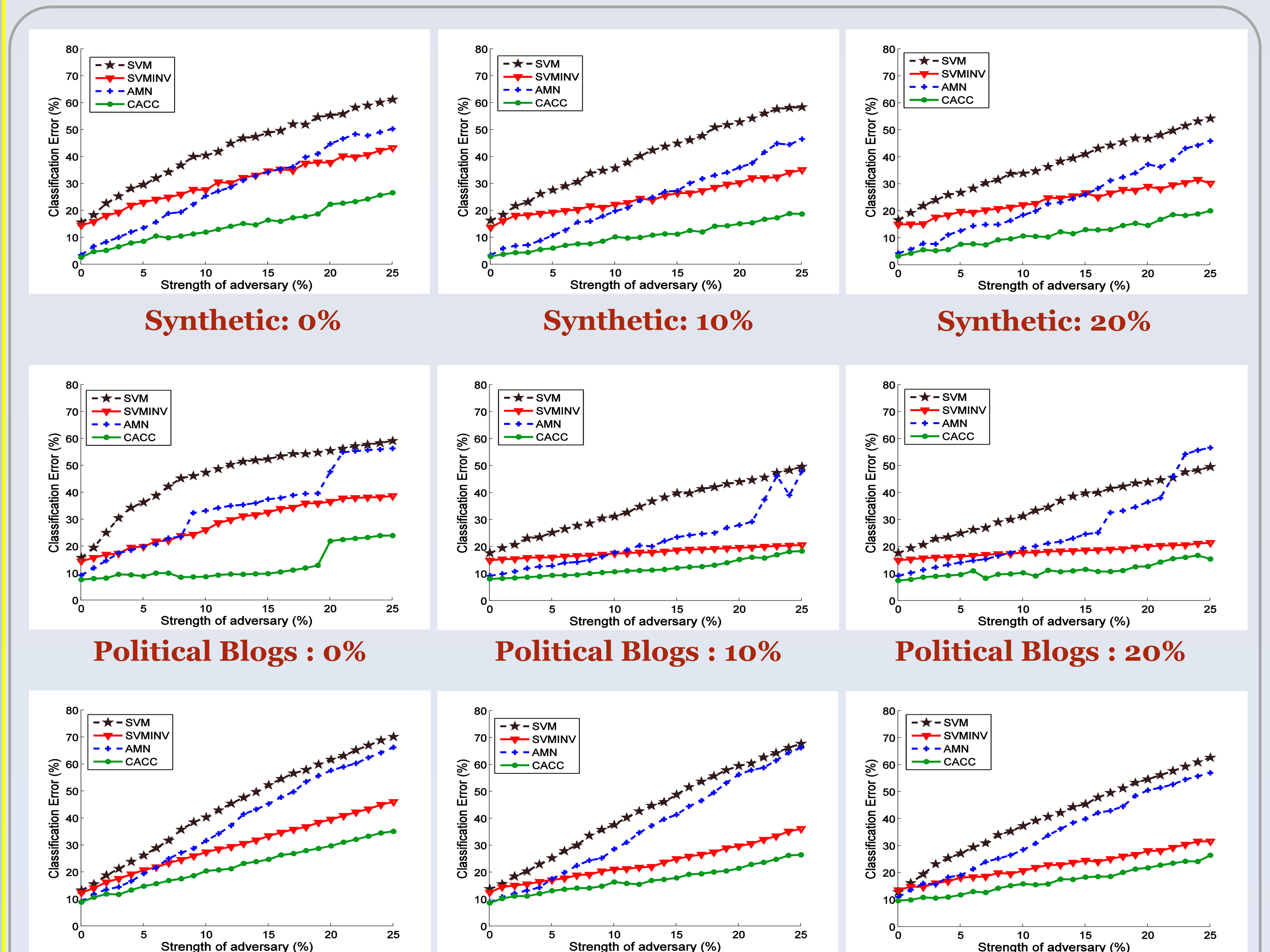
6

Experiments

7

- **Experimental Setup**
  - Naïve baseline methods: AMN [Taskar et al., 2004] and SVM
  - Robust methods: SVMInvar [Teo et al., 2008] (Baseline) and CACC (Our method)
  - Parameter C for all methods and adversary's train budget D, are tune with 0%, 10% , and 20% of adversarial manipulation strength at the tuning data.
- **Datasets**
  - **Synthetic.** 10 random graphs, each with 100 nodes (half positive and half negative labels) and 30 Boolean features. Nodes are more likely to link to the ones that have the same label, and half of the nodes were only recognizable by their links
  - **Political Blogs.** collected by [Adamic et al 2005]. We extended this dataset by crawling the blogs at different times and cleaning dead pages manually. In this dataset, we observe some concept drift at different times
  - **Reuters.** ModApte split of the Reuters-21578 corpus. Four classes: crude, grain, trade, and money-fx are selected.

Comparison with baselines



Synthetic: 0% — Synthetic: 10% — Synthetic: 20%

Political Blogs : 0% — Political Blogs : 10% — Political Blogs : 20%

Reuters : 0% — Reuters : 10% — Reuters : 20%

Conclusion and future work

- Robustness combined with the ability to reason about interrelated objects
- Representation of the adversarial learning task as a bilevel quadratic Stackelberg game

Future work: Extend our method to learn adversarially regularized variants of non-associative relational models, also scale to large size problems where many of which are semi-supervised.