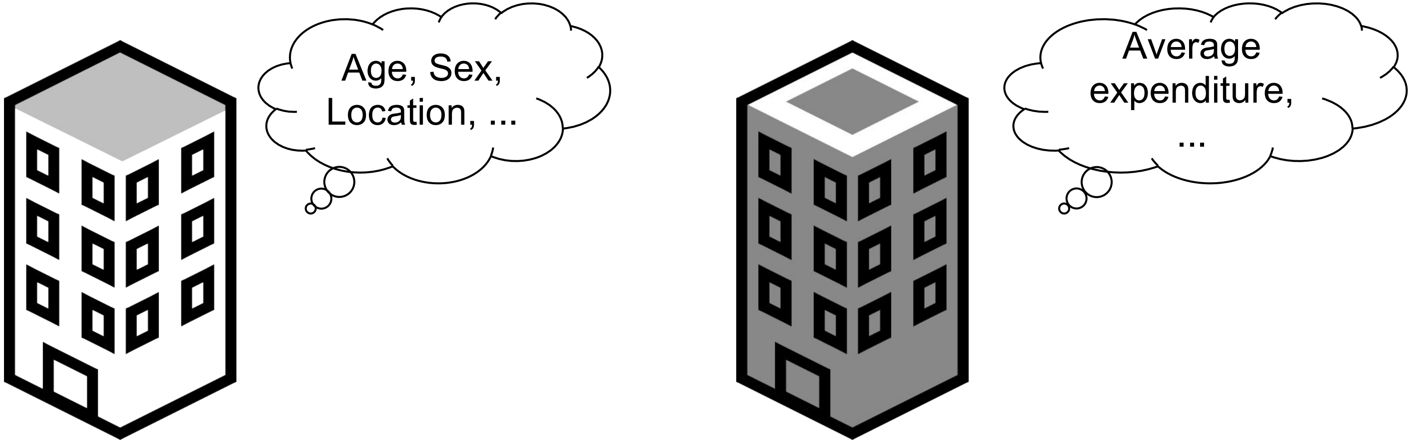
Adversarial Feature Selection

Karan K. Budhraja, Tim Oates

Contents

Motivation

- ➔ Real-world contains *vertically partitioned* data
  - E.g. company *A* has demographic data
  - E.g. company *B* has shopping data
  - Companies know certain features about a person



Motivation

➔ Example classification task: modeling personal behaviour
- ◆ Required feature data spread across companies
  - Quality of data varies per feature
  - Data access is priced
- ◆ Features accessed should be *worth paying for*
  - This is a feature selection problem

Motivation

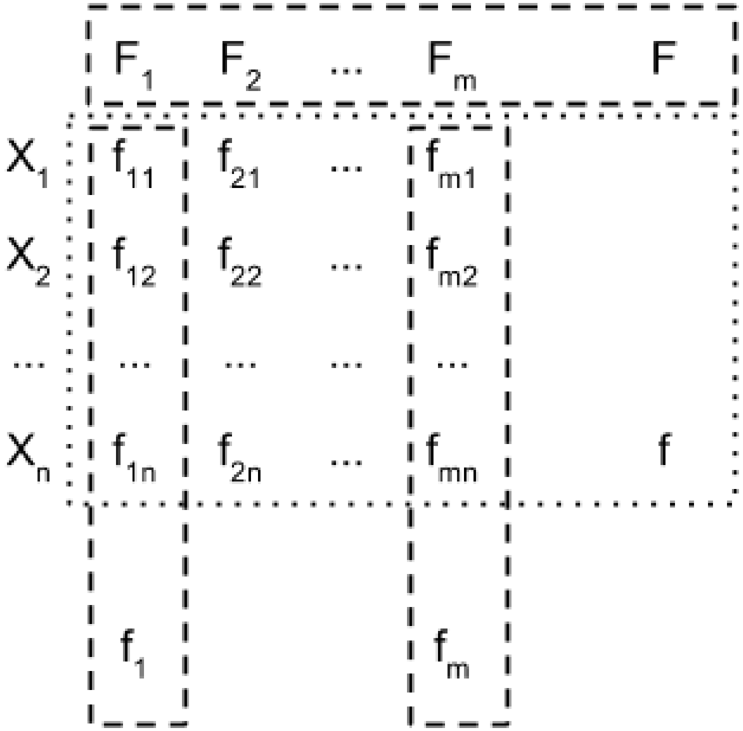
- Estimating data price
  - Price $\propto$ Data importance
  - Always? No!
    - Price $\propto$ *Perceived* data importance
- Unimportant data can be masked
  - Perceived to have increased importance
  - Identified as *adversarial* behaviour
  - New definition of *dishonesty*
  - We discuss feature selection in this environment

Related Work

- ➔ Adversarial Noise
  - ◆ Random manipulation of features in data
  - ◆ Good model of hardware failures
- ➔ Feature Blankout
  - ◆ Annul certain features in data
  - ◆ Good to learn deeper relations between features
- ➔ Such work does not consider active data manipulation

Related Work

- ➔ Adversarial Classification
- ➔ Data actively manipulated per row (*X*)
- ➔ Adversary causes misclassification
- ➔ Used as basis to formulate our problem
- ➔ Manipulate data per column (*F*)



Related Work

- ➔ Existing buyer-seller model describe dishonesty
- ➔ e.g. eBay, craigslist etc.
- ➔ Current dishonesty definitions: bad data
  - ◆ Seller initially had important features
  - ◆ Seller later only left with unimportant features
  - ◆ Dishonesty: reduction in importance of data supplied
- ➔ Our (*new*) definition: *Active modification of data by seller*

Problem Definition

- Defined as a game
  - Players: Feature Selector (*FS*), Adversary (*A*)
- Cost definition
  - *FS*: cost of accessing features (*v*)
  - *A*: cost of modifying feature (e.g. *f* to *f'*) (*Q*)
- Utility (*U*) definition
  - *FS*: proportional to classification accuracy
  - *A*: *profit earned* on modified feature
- Analogy
  - *FS*:buyer, *A*: seller

Problem Definition

- ➔ True feature costs (v) obtained as normalized weights
  - Weight assigned per feature by *Logo*
  - *Logo*: embedded feature selection technique
  - Used to provide ground truth for problem
- ➔ Q: scaled value of $|f - f'|$
- ➔ U_{FS}: scaled feature cost (scaled by K_{FS})
- ➔ U_A: scaled value of $|f - f'|$

Problem Definition

- Use of *trusted third-party*
  - Analogous to a central authority in ecosystem
  - Evaluates features on classification accuracy
  - Used to calculate U_{FS}
  - Can model situation where trusted third-party has *hidden features*
    - Private to trusted third-party
    - Hidden features: features *excluded* from feature selection game
- A may deceive *FS* to take unimportant features
  - *FS* identifies deception using trusted third-party
  - *FS* avoids that feature in the future

Proposed Method

- ➔ Adversary strategy
  - ◆ Observe *FS* behaviour
    - Observe features selected and not selected
    - Probabilistically identify unimportant (bad) features
    - Probabilistically identify important (good) features
  - ◆ Camouflage bad feature (f_{bad}) as good feature (f_{good})
    - Make f_{bad} look like f_{good} (more correlated)
    - $f'_{bad} = f_{bad} + \beta(f_{good} - f_{bad})$

Proposed Method

- Feature selector strategy
  - Probabilistically consider features provided by *A*
    - Consider features not currently in *FS* subset
    - Evaluate U_{FS} to determine adding feature to subset
  - Probabilistically remove a feature from *FS* subset
    - Removal not mandatory per round
    - Serves to purge *FS* subset
    - Not applicable if only 1 feature in *FS* subset
    - Force selection of at least 1 feature
- Turn based game
  - A’s turn, FS’s turn, A’s turn, ...

Evaluation

- ➔ Time complexity
  - N: number of data items / rows (X)
  - J: number of features / columns (F)
  - Time complexity for A is $O(N)$
  - Time complexity for FS is $O(N^2J) + O(N) = O(N^2J)$
    - $O(N^2J)$ caused by Logo: *third-party bottleneck*
  - Can be improved by using different weighting algorithm
    - Towards real time execution

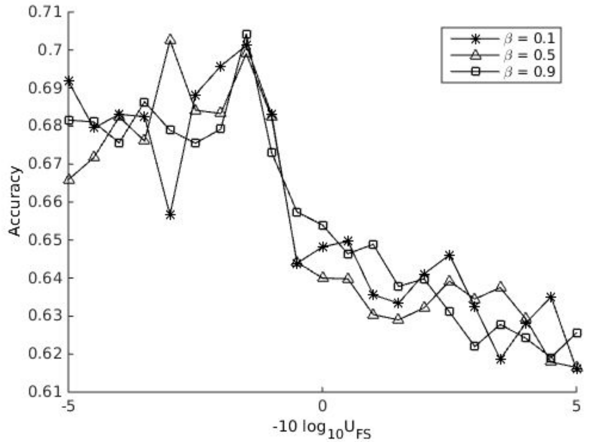
Evaluation

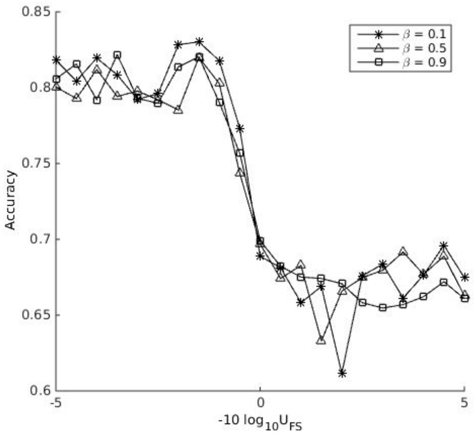
- ➔ Experimental setup
  - ◆ Preliminary evaluation: proof of concept
  - ◆ UCI datasets
    - Use: 2/3rd training data, 1/3rd test data
    - Arbitrary values of β used (*0.1, 0.5, 0.9*)
    - For reference: $f'_{bad} = f_{bad} + \beta(f_{good} - f_{bad})$
    - Testing minimal, moderate and extensive feature modifications
  - ◆ K_{FS} varied to vary U_{FS}
    - For reference: U_{FS}: scaled feature cost (scaled by K_{FS})

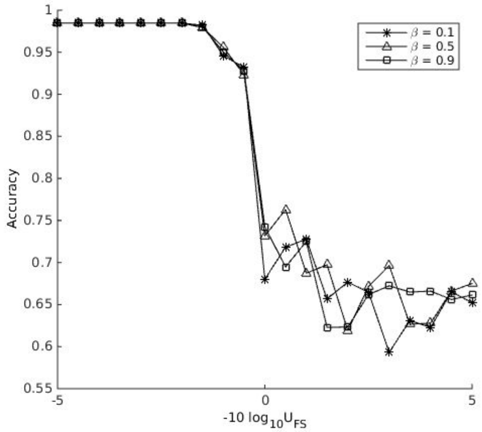
Evaluation

➔ Experimental results

- As U_{FS} decreases, features are not worth the price
- Results coherent with theory (plateaus represent *max, min* accuracy)
- E.g.: (Left to right) Diabetes, Heart, Banknote authentication datasets







Applications

- ➔ Such dishonest behaviour may not have long term benefits
  - ◆ In a scenario with *multiple buyers and sellers*, short term is extended
- ➔ Model dishonest information sharing
  - ◆ Maximize adversary profit
  - ◆ But everyone will lose trust?
    - *Cycle trust gain and trust loss*
    - Among different buyers
    - Dishonest seller sustenance

Applications

- Unexplored problem domain
  - Seller perspective
    - Maximize profit by dishonesty
  - Buyer perspective
    - Learn to avoid such dishonesty

Current and Future Work

- ➔ Current Work
  - ◆ Improve efficiency of third-party evaluation
    - Using preprocessing / memoization
  - ◆ Model multiple buyers (*FS*) and sellers (*A*)
- ➔ Future work
  - ◆ *FS* currently has unlimited purchasing power
  - ◆ More interesting if budget is limited

Selected References

Dalvi, Nilesh, et al. "Adversarial classification." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.

Sun, Yijun, Sinisa Todorovic, and Steve Goodison. "Local-learning-based feature selection for high-dimensional data analysis." Pattern Analysis and Machine Intelligence, IEEE Transactions on 32.9 (2010): 1610-1626.

Peng, Xinjun, and Dong Xu. "A local information-based feature-selection algorithm for data regression." Pattern Recognition 46.9 (2013): 2519-2530.

Dekel, Ofer, Ohad Shamir, and Lin Xiao. "Learning to classify with missing and corrupted features." Machine learning 81.2 (2010): 149-178.

Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1 (1997): 273-324.

Questions?