On the Complexity of Aggregating Information for Authentication and Profiling

Christian A. Duncan Vir V. Phoha

Louisiana Tech University

Data Privacy Management 2011

Outline

1. Motivation
   - Sharing Information
   - Relevant Work
2. Theory
   - Model Overview
   - NP-Complete
   - Pseudo-polynomial Time Solution
3. Experimental Results
   - Keystroke Authentication
   - Feature Selection

Outline

The Drug

- Social Networking: Communicate with
  - Relatives
  - Friends
  - Acquaintances
  - Strangers
- Convenient (and quite useful)
- ... but sometimes too convenient.

The Drug

- Social Networking: Communicate with
  - Relatives
  - Friends
  - Acquaintances
  - Strangers
- Convenient (and quite useful)
- ... but sometimes too convenient.

The Drug

- Social Networking: Communicate with
  - Relatives
  - Friends
  - Acquaintances
  - Strangers
- Convenient (and quite useful)
- ... but sometimes too convenient.

The Abuser

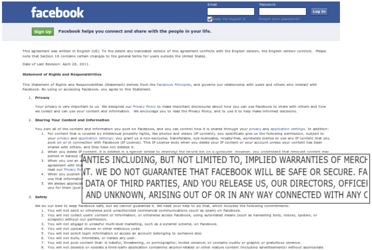
- People often reveal too much information...
- across numerous sites.
- Intentional: User doesn't care or think of consequences
- Unintentional: Didn't read the fine-print
- No control: Stolen information... or even friends.

The Abuser

- People often reveal too much information...
- across numerous sites.
- Intentional: User doesn't care or think of consequences
- Unintentional: Didn't read the fine-print
- No control: Stolen information... or even friends.

The Abuser

- People often reveal too much information...
- across numerous sites.
- Intentional: User doesn't care or think of consequences
- Unintentional: Didn't read the fine-print
- No control: Stolen information... or even friends.



The Abuser

- People often reveal too much information...
- across numerous sites.
- Intentional: User doesn't care or think of consequences
- Unintentional: Didn't read the fine-print
- No control: Stolen information... or even friends.

Happy Birthday

Alice: **posted on 2011/09/15**
`Happy 40th Birthday, Bob!`

Bob: **posted on 2011/09/15**
`Thanks! Why not just go ahead and tell everyone my Bank Account Number too.`

Alice: **posted on 2011/09/15**
`Um, ok.`

The Collector

- Aggregates that information
- Generates profile of user(s)
- Examples:
  - Police (criminal inv.)
  - Business (ad. revenue)
  - Employer (security)

The Collector's Intent

The collector's intent could be

- Malicious (to the individual):
  - No concern for individual's privacy.
  - Concern for best profile information.
- Ambivalent:
  - No malicious intent. Simply wants a good profile.
  - Still often disregards individual's privacy, or treats as secondary.
- Benevolent:
  - Individual privacy a top priority.
  - Wishes to maximize profile information while respecting privacy.

The Collector's Intent

The collector's intent could be

- Malicious (to the individual):
  - No concern for individual's privacy.
  - Concern for best profile information.
- Ambivalent:
  - No malicious intent. Simply wants a good profile.
  - Still often disregards individual's privacy, or treats as secondary.
- Benevolent:
  - Individual privacy a top priority.
  - Wishes to maximize profile information while respecting privacy.

The Collector's Intent

The collector's intent could be

- Malicious (to the individual):
  - No concern for individual's privacy.
  - Concern for best profile information.
- Ambivalent:
  - No malicious intent. Simply wants a good profile.
  - Still often disregards individual's privacy, or treats as secondary.
- Benevolent:
  - Individual privacy a top priority.
  - Wishes to maximize profile information while respecting privacy.

The Collector's Intent

The collector's intent could be

- Malicious (to the individual):
  - No concern for individual's privacy.
  - Concern for best profile information.
- Ambivalent:
  - No malicious intent. Simply wants a good profile.
  - Still often disregards individual's privacy, or treats as secondary.
- Benevolent:
  - Individual privacy a top priority.
  - Wishes to maximize profile information while respecting privacy.

The Collector's Intent

The collector's intent could be

- Malicious (to the individual):
  - No concern for individual's privacy.
  - Concern for best profile information.
- Ambivalent:
  - No malicious intent. Simply wants a good profile.
  - Still often disregards individual's privacy, or treats as secondary.
- Benevolent:
  - Individual privacy a top priority.
  - Wishes to maximize profile information while respecting privacy.

The Collector's Intent

The collector's intent could be

- Malicious (to the individual):
  - No concern for individual's privacy.
  - Concern for best profile information.
- Ambivalent:
  - No malicious intent. Simply wants a good profile.
  - Still often disregards individual's privacy, or treats as secondary.
- Benevolent:
  - Individual privacy a top priority.
  - Wishes to maximize profile information while respecting privacy.

Examples

Malicious

Stealing Reality by Altschuler et al. [1]

- Malware threat that steals personal and behavioral info.
- Not just email addresses, passwords, phone numbers, etc.
- Gets static info: birthdate, mother's maiden name.
- Challenge: Very hard to change once acquired.

[1] Y. Altshuler, N. Aharony, Y. Elovici, A. Pentland, and M. Cebrian. Stealing reality. Tech. rep., arXiv, October 2010. arXiv:1010.1028v1

Examples

Benevolent

PerGym by Pareschi et al. [2]

- Provides context-aware personalized services... while maintaining strong system security.
- Gym service: monitors workout experience, e.g.
  - Body temperature, Location, Mood
- User wishes to use service but does not trust enough to provide all info.

[2] L. Pareschi, D. Riboni, A. Agostini, and C. Bettini. Composition and generalization of context data for privacy preservation. *Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008)*., pp. 429 –433, March 2008, `http://dx.doi.org/10.1109/PERCOM.2008.47`

Examples

Ambivalent

User authentication

- Old school: Password
- Biometrics: fingerprint, voice, face, typing pattern
- Multiple: Password, voice, *and* fingerprint scan
- System needs to collect biometric information.
- User might not want system to store all such information.

Outline

Relevant Work

- Carminati et al. [3] provide model to give user strong control over access to private info.
- Gambs et al. [4] discuss how geolocated applications (Google Latitude) enable a user to reveal too much personal info by sharing positional and mobility info.

[3] B. Carminati, E. Ferrari, and A. Perego. Enforcing access control in web-based social networks. *ACM Trans. Inf. Syst. Secur.* 13:6:1–6:38, November 2009, `http://doi.acm.org/10.1145/1609956.1609962`

[4] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Show me how you move and I will tell you who you are. *Transactions on Data Privacy* 4(2):103–126, 2011

Relevant Work

- Liu and Terzi [5] estimate user's privacy score from info they provide online, notifying user if it exceeds selected threshold. (Like credit score/credit watch)
- Domingo-Ferrer [6] discuss trade-offs between privacy and functionality: cooperation while preventing "free rides"

[5] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. *ACM Trans. Knowl. Discov. Data* 5:6:1–6:30, December 2010, `http://doi.acm.org/10.1145/1870096.1870102`

[6] J. Domingo-Ferrer. Rational privacy disclosure in social networks. *Modeling Decisions for Artificial Intelligence*, vol. 6408, pp. 255–265. Springer Berlin / Heidelberg, Lecture Notes in Computer Science, 2010, `http://dx.doi.org/10.1007/978-3-642-16292-3_25`

Outline

Model Assumptions

User has

- collection of private info (facts) $S = \{f_1, f_2, \ldots, f_n\}$,
- weights - importance of each fact, and
- a notion of acceptable privacy based on combination of these weights.

Model Assumptions

Aggregator has

- algorithm to generate profile from given subset of S
- including a (confidence/quality) score,
- minimum score threshold (valid/acceptable profile), and
- costs associated with collection of each fact.
  - Home address and phone number purchased by phonebook database.
  - Birth dates might require thorough searching of public birth records or social engineering.
  - Fingerprint relatively inexpensive.
  - DNA sample might be a bit more costly (and intrusive).

Model Assumptions

Benevolent aggregator

Success: if can find a subset of facts generating acceptable profile while not exceeding user's privacy threshold or possible collection cost limits.

Malicious aggregator

Same but simply ignores privacy threshold, and would still be bound by cost limitations.

Model Assumptions

- Given set S of facts
- Find subset $S' \subseteq S$
- Given profile function $F^p(S')$ and threshold T^p:
  Measure score of profile using S'
- Given privacy function $F^u(S')$ and threshold T^u:
  Measure user's privacy score of having revealed S'
- Given cost function $F^c(S')$ and threshold W:
  Cost of acquiring S'
- A subset S' yields *valid* profile if $F^p(S') \geq T^p$ and $F^u(S') \leq T^u$ (for benevolent aggregators).

Goal and Problems

Goal

Analyze complexity of determining what information of a user is most valuable to collect given acquisition costs to create an acceptable (valid) profile.

Problems

- More information does not nec. mean better profile
- Valuable but costly info
- Incorrect or contradictory info
- Value of item might depend on other info as well

Outline

1. Motivation
   - Sharing Information
   - Relevant Work
2. Theory
   - Model Overview
   - NP-Complete
   - Pseudo-polynomial Time Solution
3. Experimental Results
   - Keystroke Authentication
   - Feature Selection

Profile Aggregator Problem

Theorem 1

Given

- *a set S of facts,*
- *a cost function F^c, a cost goal W,*
- *profiling function F^p, and confidence threshold T^p,*

NP-C to determine if exists valid $S' \subseteq S$ s.t. $F^c(S') \leq W$.

That is, (most likely) no polynomial-time algorithm exists that can select sufficient info (valid profile) while minimizing cost.

Since this holds when ignoring privacy function, it also holds with privacy function.

Proof

Due to a reduction from the classic 0-1 Knapsack problem.

Outline

Pseudo-polynomial Time Solution: 0-1 Knapsack

- Given n items, with value v_i and weight w_i,
- find a subset of items such that
  - total weight is below some limit W and
  - total value is as large as possible.
- Though NP-complete, pseudo-poly solution exists using dynamic programming.
- Time is $O(nW)$ - thus polynomial in W.
- Result works because adding an item i, increases the total value by v_i and the total weight by w_i.
- That is, the value and weight functions are monotonic.
- In our setting, the weight function is the cost function F^c and the value function is the profile function F^p.
- Thus...

Pseudo-polynomial Time Solution: Profile Aggregator

Theorem 2

Given

- *a set S of facts,*
- *a* monotonic *cost function F^c, a cost goal W,*
- *a* monotonic *profiling function F^p, and confidence threshold T^p.*

One can determine in time $O(nW)$ if there exists valid $S' \subseteq S$ such that $F^c(S') \leq W$.
(Note this only applies to the case when privacy is ignored.)

Pseudo-polynomial Time Solution: Profile Aggregator

Theorem 2

Given

- *a set S of facts,*
- a monotonic *cost function F^c, a cost goal W,*
- a monotonic *profiling function F^p, and confidence threshold T^p.*

One can determine in time $O(nW)$ if there exists valid $S' \subseteq S$ such that $F^c(S') \leq W$.
(Note this only applies to the case when privacy is ignored.)

LIE LIE LIE LIE LIE

Monotonic versus Consistently Monotonic

Monotonic

A function is ***monotonic*** if for two subsets A and B, $F(A) \leq F(A \cup B)$. That is, adding elements to a subset will never decrease the score.

Consistently Monotonic

A function is ***consistently monotonic*** if for three subsets A, B, and C, $F(A) \leq F(B) \rightarrow F(A \cup C) \leq F(B \cup C)$.
That is, if the score for A is lower than for B then adding C to both sets will not change this order.

Monotonic versus Consistently Monotonic

Informal Example

- Assume one is going backpacking across Europe
- and has to choose among several food staples

  (just a subset here.)

  A. Potato Chips
  B. Canned food
  C. Can opener
- If choosing just one item, we have a clear winner - $F(A)$ is going to be better than the other two.
- Adding any item does not decrease score - so *monotonic*.
- However, although $F(B) \leq F(A)$, clearly (for health reasons) $F(B \cup C) > F(A \cup C)$ - so *not consistently monotonic*.

Monotonic versus Consistently Monotonic

One more issue

- Dynamic programming solution requires that values for the cost function be nonnegative integers.
- Or else it cannot store all possible cost values.
- Can scale if within a known fractional range.
- For simplicity, assume purely a summation of costs.

Pseudo-polynomial Time Solution: Profile Aggregator

Theorem 2

Given

- *a set S of facts,*
- *a set of integer costs c_s, one per fact s, a cost goal W,*
- *a* consistently monotonic *profiling function F^p and T^p.*

Can see in time $O(nW)$ if there exists valid $S' \subseteq S$ such that $\Sigma_{s \in S'} c_s \leq W$.
(Note this still only applies to the case when privacy is ignored.)

Theorem 3 (Monotonic case):

When F^p is merely monotonic, NP-complete even if $W \in \Theta(n^k)$.

Reduction from the Vertex-Cover Problem.

Outline

1. Motivation
   - Sharing Information
   - Relevant Work
2. Theory
   - Model Overview
   - NP-Complete
   - Pseudo-polynomial Time Solution
3. Experimental Results
   - Keystroke Authentication
   - Feature Selection

Justification

- Increasing the number of facts collected (and used) does not necessarily improve profile generated.
- In fact, it may hurt it... significantly.
- Do an experiment to see this.

Justification

- Increasing the number of facts collected (and used) does not necessarily improve profile generated.
- In fact, it may hurt it... significantly.
- Do an experiment to see this.

Justification

- Increasing the number of facts collected (and used) does not necessarily improve profile generated.
- In fact, it may hurt it... significantly.
- Do an experiment to see this.

Keystroke Authentication

- Traditional Authentication: User enters a password and system checks if password matches.
- Here: Authentication system collects (and verifies) password but also collects keystroke information, namely:
    - Key hold latencies: press to release of same key
    - Key interval latencies: release to press of new key
    - Key press latencies: press of one key to the next
- User authenticates if enters correct password *and* keystroke pattern best matches claimed user's.

Keystroke Authentication

- Traditional Authentication: User enters a password and system checks if password matches.
- Here: Authentication system collects (and verifies) password but also collects keystroke information, namely:
  - Key hold latencies: press to release of same key
  - Key interval latencies: release to press of new key
  - Key press latencies: press of one key to the next
- User authenticates if enters correct password *and* keystroke pattern best matches claimed user's.

Keystroke Authentication

- Traditional Authentication: User enters a password and system checks if password matches.
- Here: Authentication system collects (and verifies) password but also collects keystroke information, namely:
    - Key hold latencies: press to release of same key
    - Key interval latencies: release to press of new key
    - Key press latencies: press of one key to the next
- User authenticates if enters correct password *and* keystroke pattern best matches claimed user's.

Keystroke Authentication

- Traditional Authentication: User enters a password and system checks if password matches.
- Here: Authentication system collects (and verifies) password but also collects keystroke information, namely:
    - Key hold latencies: press to release of same key
    - Key interval latencies: release to press of new key
    - Key press latencies: press of one key to the next
- User authenticates if enters correct password *and* keystroke pattern best matches claimed user's.

Keystroke Authentication

- Our data consists of 43 users entering a 37-character phrases (repeatedly - 9 times).
- 37 characters means we had $37 \cdot 3 - 2 = 109$ features.
- Each feature represents one dimension in 109-d space.
- Contains $43 \cdot 9 = 387$ points in this space.

Classification

Process works as follows:

- Train on a sample of the data set - creating a classification system.
- For a test point, query the system to identify to which user class this point most likely belongs.
- If it matches the known user for this query, considered a correct match; otherwise, considered an error.
- Used LOOCV (leave-one-out cross validation) scheme, training data is all but one item (the test query).

Classification

Process works as follows:

- For given training set and a subset of 109 features,
- build classifiers on feature subset for this training set.
- A successful profile is one where the user matches.
- The confidence in our profile function is the accuracy it is estimated to predict correctly.
- $F(S')$ is the accuracy of classifier, as measured by percentage of correct classifications.
- Wish to identify the subset that maximizes this function. Thus, classifier remains fixed but features to train vary.

Classification

Process works as follows:

- Trying all possible 2^{109} subsets of features is infeasible.
- Heuristics would likely do well but our goal is to "justify that more is not always better" and to stress the importance of selecting a good subset.
- Not to discover the best way to find a subset.
- We also chose to use the weighted k-nearest neighbors classifier
  - for its simplicity and
  - decent classification abilities.
  - By no means is this an optimal classifier.

Outline

Experiment

- LOOCV
- k-NN classifier
- Best subset of 109 features
- Profiling function is too complicated to analyze directly and in fact depends on the training data.
- Two approaches to choosing features:
  - Dynamic programming:
    even though do not know if function is cons. monotonic.
  - Sequential approach (in order until "full"):
    For comparison and to help see property of the function.
- Ran two versions of experiment:
  - with equal (unit) weights per feature.
    Cost for using k features is k.
  - with weight growing linearly based on character position.
    Reflects user exhaustion - longer sequences, higher cost.

Experiment

- LOOCV
- k-NN classifier
- Best subset of 109 features
- Profiling function is too complicated to analyze directly and in fact depends on the training data.
- Two approaches to choosing features:
  - Dynamic programming:
    even though do not know if function is cons. monotonic.
  - Sequential approach (in order until "full"):
    For comparison and to help see property of the function.
- Ran two versions of experiment:
  - with equal (unit) weights per feature.
    Cost for using k features is k.
  - with weight growing linearly based on character position.
    Reflects user exhaustion - longer sequences, higher cost.

Experiment

- LOOCV
- k-NN classifier
- Best subset of 109 features
- Profiling function is too complicated to analyze directly and in fact depends on the training data.
- Two approaches to choosing features:
  - Dynamic programming:

    even though do not know if function is cons. monotonic.
  - Sequential approach (in order until “full”):

    For comparison and to help see property of the function.
- Ran two versions of experiment:
  - with equal (unit) weights per feature.
    Cost for using k features is k.
  - with weight growing linearly based on character position.
    Reflects user exhaustion - longer sequences, higher cost.

Experiment

- LOOCV
- k-NN classifier
- Best subset of 109 features
- Profiling function is too complicated to analyze directly and in fact depends on the training data.
- Two approaches to choosing features:
  - Dynamic programming:
    even though do not know if function is cons. monotonic.
  - Sequential approach (in order until “full”):
    For comparison and to help see property of the function.
- Ran two versions of experiment:
  - with equal (unit) weights per feature.
    Cost for using k features is k.
  - with weight growing linearly based on character position.
    Reflects user exhaustion - longer sequences, higher cost.

Experiment

- LOOCV
- k-NN classifier
- Best subset of 109 features
- Profiling function is too complicated to analyze directly and in fact depends on the training data.
- Two approaches to choosing features:
  - Dynamic programming:
    even though do not know if function is cons. monotonic.
  - Sequential approach (in order until “full”):
    For comparison and to help see property of the function.
- Ran two versions of experiment:
  - with equal (unit) weights per feature.
    Cost for using k features is k.
  - with weight growing linearly based on character position.
    Reflects user exhaustion - longer sequences, higher cost.

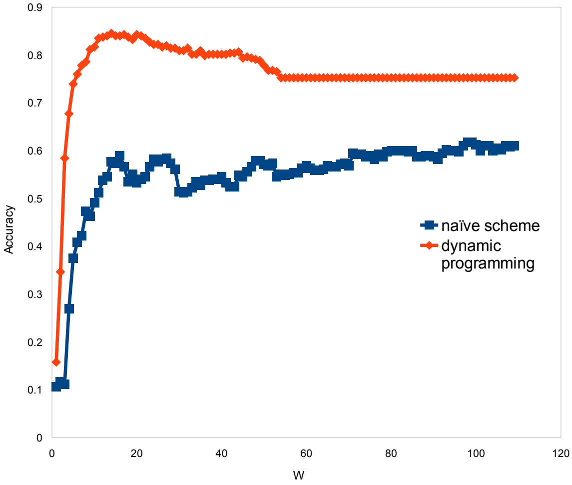
Experiment

- LOOCV
- k-NN classifier
- Best subset of 109 features
- Profiling function is too complicated to analyze directly and in fact depends on the training data.
- Two approaches to choosing features:
  - Dynamic programming:

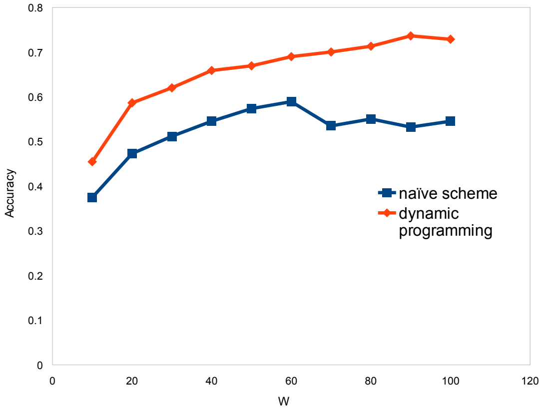
    even though do not know if function is cons. monotonic.
  - Sequential approach (in order until “full”):

    For comparison and to help see property of the function.
- Ran two versions of experiment:
  - with equal (unit) weights per feature.
    Cost for using k features is k.
  - with weight growing linearly based on character position.
    Reflects user exhaustion - longer sequences, higher cost.

Experiment (Equal Weights)



Experiment (Increasing Weights)



Summary

- Information aggregation - good and bad uses
- Minimizing cost/maximizing profit - difficult in theory

  Not surprising
- The properties of profit function affect difficulty

  Not surprising
- Being monotonic isn't particularly helpful but being consistently monotonic is.

  Surprising?
- Picking correct subset of information is important

  More is definitely not always better
- Future Outlook
  - Study other (real) classifiers: even better improvements?
  - Study heuristical means of selecting features: comparison to DP version

Summary

- Information aggregation - good and bad uses
- Minimizing cost/maximizing profit - difficult in theory

  Not surprising
- The properties of profit function affect difficulty

  Not surprising
- Being monotonic isn't particularly helpful but being consistently monotonic is.

  Surprising?
- Picking correct subset of information is important

  More is definitely not always better
- Future Outlook
  - Study other (real) classifiers: even better improvements?
  - Study heuristical means of selecting features: comparison to DP version

Summary

- Information aggregation - good and bad uses
- Minimizing cost/maximizing profit - difficult in theory

  Not surprising
- The properties of profit function affect difficulty

  Not surprising
- Being monotonic isn't particularly helpful but being consistently monotonic is.

  Surprising?
- Picking correct subset of information is important

  More is definitely not always better
- Future Outlook
  - Study other (real) classifiers: even better improvements?
  - Study heuristical means of selecting features: comparison to DP version

Summary

- Information aggregation - good and bad uses
- Minimizing cost/maximizing profit - difficult in theory

  Not surprising
- The properties of profit function affect difficulty

  Not surprising
- Being monotonic isn't particularly helpful but being consistently monotonic is.

  Surprising?
- Picking correct subset of information is important

  More is definitely not always better
- Future Outlook
  - Study other (real) classifiers: even better improvements?
  - Study heuristical means of selecting features: comparison to DP version

Summary

- Information aggregation - good and bad uses
- Minimizing cost/maximizing profit - difficult in theory

  Not surprising
- The properties of profit function affect difficulty

  Not surprising
- Being monotonic isn't particularly helpful but being consistently monotonic is.

  Surprising?
- Picking correct subset of information is important

  More is definitely not always better
- Future Outlook
  - Study other (real) classifiers: even better improvements?
  - Study heuristical means of selecting features: comparison to DP version

Summary

- Information aggregation - good and bad uses
- Minimizing cost/maximizing profit - difficult in theory

  Not surprising
- The properties of profit function affect difficulty

  Not surprising
- Being monotonic isn't particularly helpful but being consistently monotonic is.

  Surprising?
- Picking correct subset of information is important

  More is definitely not always better
- Future Outlook
  - Study other (real) classifiers: even better improvements?
  - Study heuristical means of selecting features: comparison to DP version

Summary

- Information aggregation - good and bad uses
- Minimizing cost/maximizing profit - difficult in theory

  Not surprising
- The properties of profit function affect difficulty

  Not surprising
- Being monotonic isn't particularly helpful but being consistently monotonic is.

  Surprising?
- Picking correct subset of information is important

  More is definitely not always better
- Future Outlook
  - Study other (real) classifiers: even better improvements?
  - Study heuristical means of selecting features: comparison to DP version

Summary

- Information aggregation - good and bad uses
- Minimizing cost/maximizing profit - difficult in theory

  Not surprising
- The properties of profit function affect difficulty

  Not surprising
- Being monotonic isn't particularly helpful but being consistently monotonic is.

  Surprising?
- Picking correct subset of information is important

  More is definitely not always better
- Future Outlook
  - Study other (real) classifiers: even better improvements?
  - Study heuristical means of selecting features: comparison to DP version

Summary

Any Questions?