For this talk, Adam=Atom

Classification as an Adam

John Langford

Outline

1. Adamicity

2. Example Reductions

Adamicity: Why?

Consider:

1. Transistor (For all those computers we know and love.)

2. Triangle (Quake)

3. Simplex (PDE solvers)

Small, Simple, easy to understand $\Rightarrow$ ability to build circuits, scenes, predict in physics.

Adamicity

1. Density estimation outputs a “real” number.

2. Regression outputs a “real” number.

3. Classification outputs 1 bit.

Classification is more adamic.

Adamicity Implication: Durability

We want a tool which always works.

Always works $\Rightarrow$ minimal assumptions.

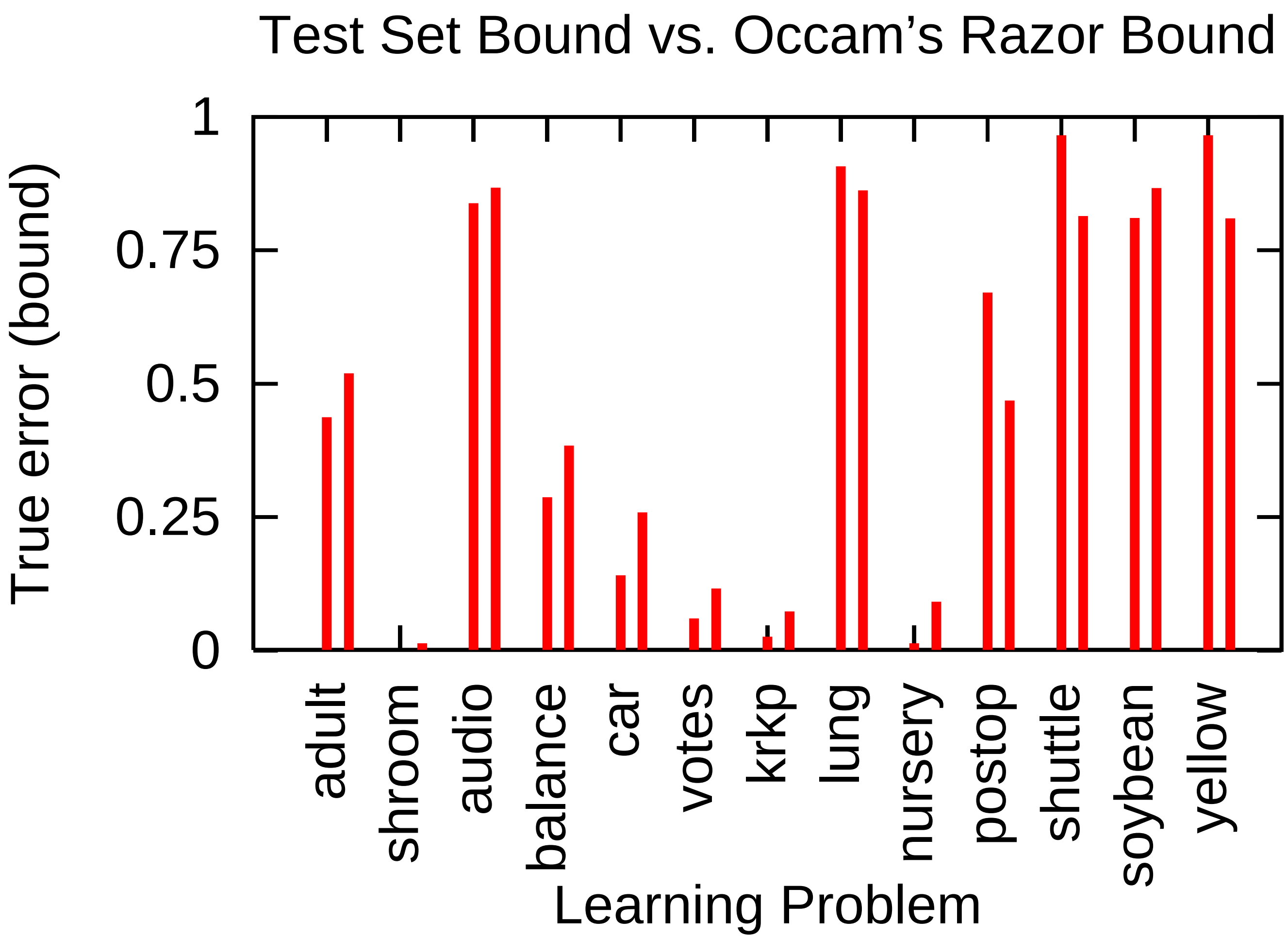
Common assumption = independence of samples.

Durability: confidence in results

1. No confidence intervals for density estimation without extra assumptions.

2. No confidence intervals for regression without extra assumptions.

3. Confidence intervals do exist for classification, assuming only independence.


Test Set Bound vs. Occam's Razor Bound
True error (bound)
0
0.25
0.5
0.75
1
adult
shroom
audio
balance
car
votes
krkp
lung
nursery
postop
shuttle
soybean
yellow
Learning Problem

Durability: Principled methods

Density Estimation method motivations very weak (never have ∞ datasets).

Regression method motivations weak in practice (never have sufficiently large datasets).

Classification method motivations often weak but sometimes not in practice.

Outline

1. Adamicity

2. Example Reductions

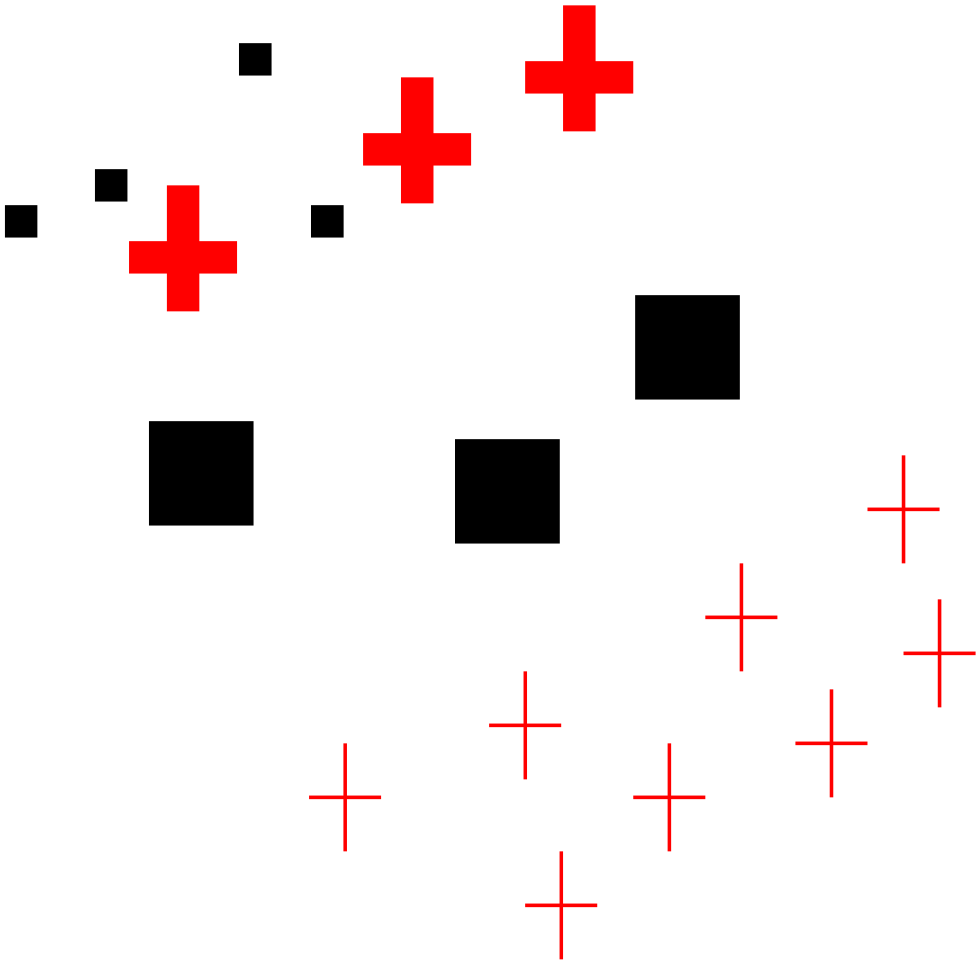
Example I: Cost Sensitive Classification

Many business decision problems have 'costs' and 'revenues' associated with them.

Basic observation: Some decisions more important than others.

Importance = the change in profit for correct vs incorrect decision.

Heavy hitters and Lightweights in Classification



Algorithm (Bianca Zadrozny): Costing (converts any 0-1 loss minimizing classifier into a cost sensitive classifier)

Analysis (Naoki Abe): In the PAC model of learning, costing is a good idea.

Results: Comparable to winner on KDDcup 98 dataset.

Note: No large suite of cost sensitive learning problems for comprehensive comparison.

Example II: Outlier Detection

All training examples come from the “common” class.

Future examples come from “common” and “outlier” classes.

Method (with Naoki Abe):

1. create bogus second class so as to get a 'background class'.

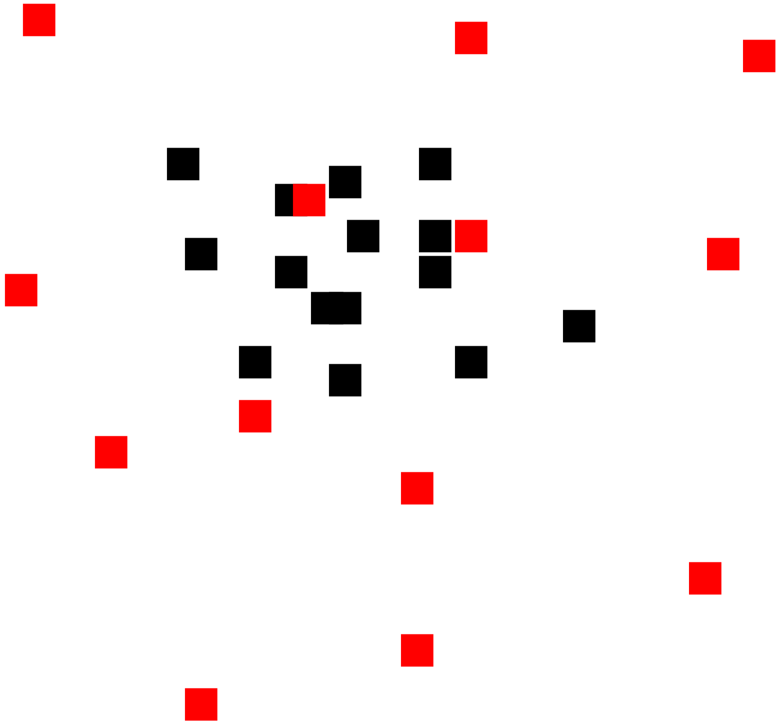
2. Classifier will tend to find the smallest region(s) enclosing the "common" class.

3. Everything else is an outlier.

Tuning ratio of background to common points = tuning sensitivity.

Cool side note: all those classification bounds apply here.

The Common Class and the background class



Example III: Reinforcement Learning given Model access

The MDP setup:

1. get state s

2. Choose action a with policy $\pi(s)$

3. Get Reward r, next state s'

4. repeat

Note that $\pi(s)$ is really a classifier of sorts.

Can we use a classifier learning algorithm to learn π?

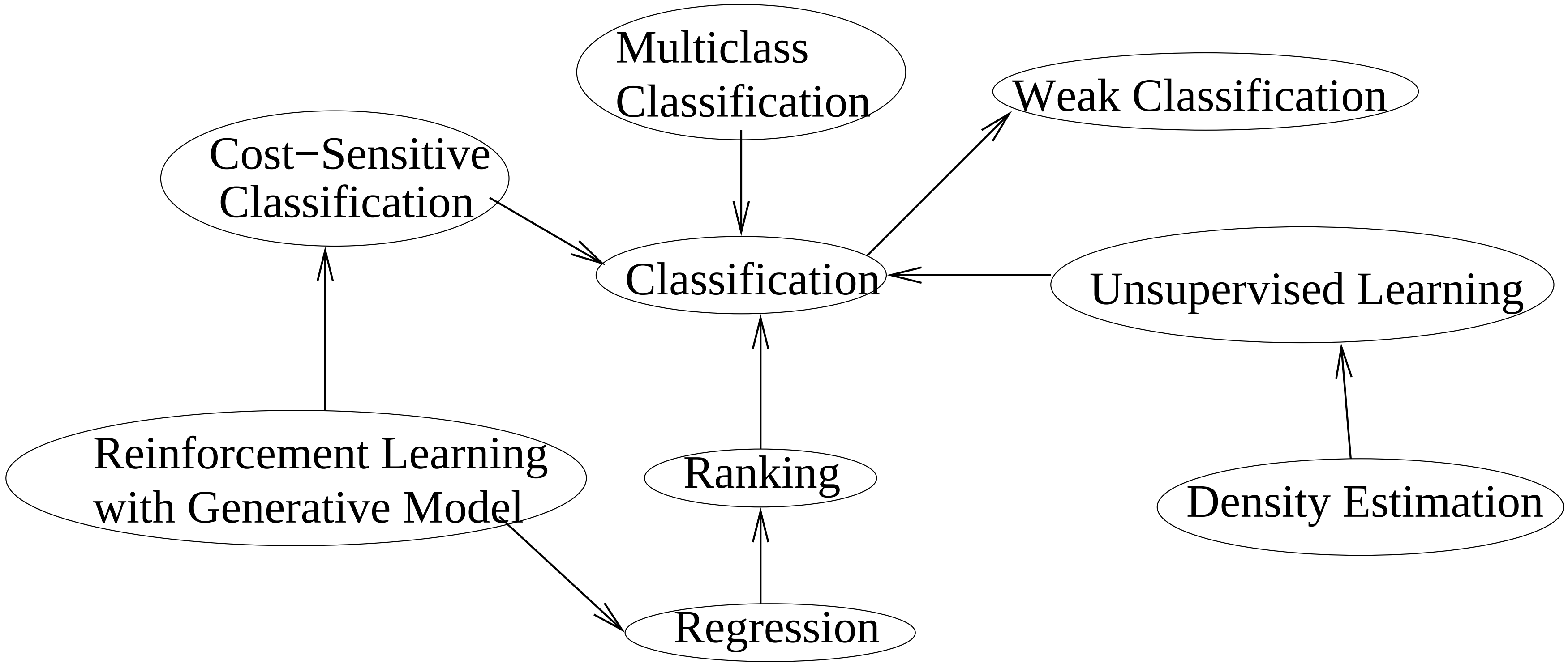
Theorem: Yes, WITH blackbox access to the MDP.

Results

(Andrew Ng and Jin Kim): Performance superior to control algorithms on simulated control systems. (Note: control algorithms optimized for worst case performance.)

(Michail Lougadakis and Ron Parr): SVM can solve the “bicycle problem”.

(Alan Fern and Robert Givan): Slightly different reduction, with positive results.


Multiclass Classification
Weak Classification
Cost−Sensitive Classification
Classification
Unsupervised Learning
Reinforcement Learning with Generative Model
Ranking
Density Estimation
Regression

Conclusion

Classification is atomic and thus durable.

As an engineering principle: Try to use or abuse classification first, and if that fails think about other tools.

Open Problems

1. What is a good reduction? Bad Reduction?

2. To what extent is “hand compilation” more effective than “automatic compilation”?

3. Are there natural problems not reducable to classification?