The Method of Reduction in Learning

John Langford

Reductions Coauthors: Naoki Abe, Drew Bagnell, Nina Balcan, Alina Beygelzimer, Varsha Dani, Hal Daume, Tom Hayes, Roberto Oliveira, Bianca Zadrozny

What is the goal?

Automated, effective, robust learning methods.

1. Mega-use.

2. Space travel.

3. (AI...)

How can we hope to achieve the goal?

We must hope the world is simple, somehow.

We hope that there exists some general purpose learning algorithm capable of solving the problems we care about.

- For some learning problems, problem specific prior information is required for success. We hope these aren't our problems.

Characteristics of the Master Prediction Engine

1. Cope with any available loss information. Techniques for optimizing loss l' when you care about loss l can not work in general.

2. Use little to lots of samples.

3. Use little to lots of features.

4. Computationally efficient.

Featuritis is disallowed.

Coping with Loss

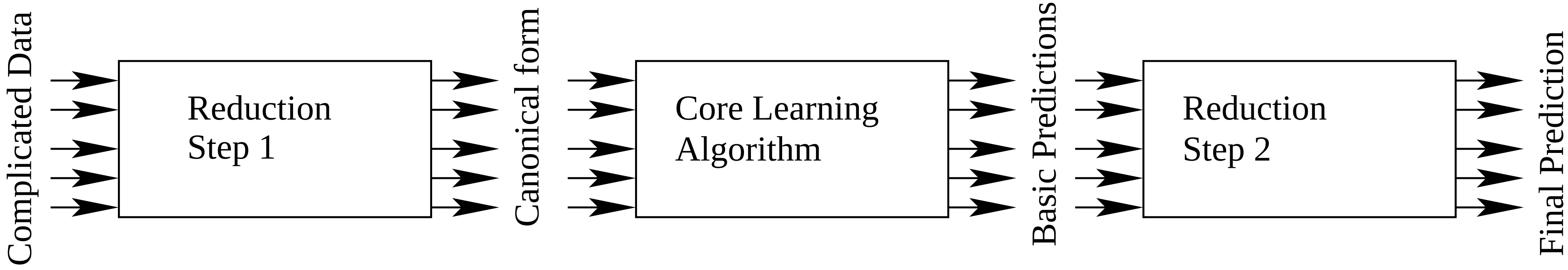
Approaches:

1. Ignore the loss function imposed by the world and optimize your favorite (log loss, hinge loss, 0/1 loss, l_2, etc...) at learning time. This can not work in general (but does work sometimes).

2. New loss function, new learning algorithm. Effective, but unsatisfying. There are *many* loss functions, so this requires much work.

3. Learning Reductions. Maybe we can transform the ability to solve one loss into the ability to solve another.

What is a learning reduction?



Some properties that might be useful for you

1. Learning reduction = making 10^{+} or so learning algorithms. (Just swap in any of several learning cores.)

2. Major reuse $\Rightarrow$ fast robust code.

3. Theory weenie curiousity: What are the vital ingredients of learning? Are all learning problems the same? What does "similar learning problem" mean?

Outline

1. What is a learning reduction?

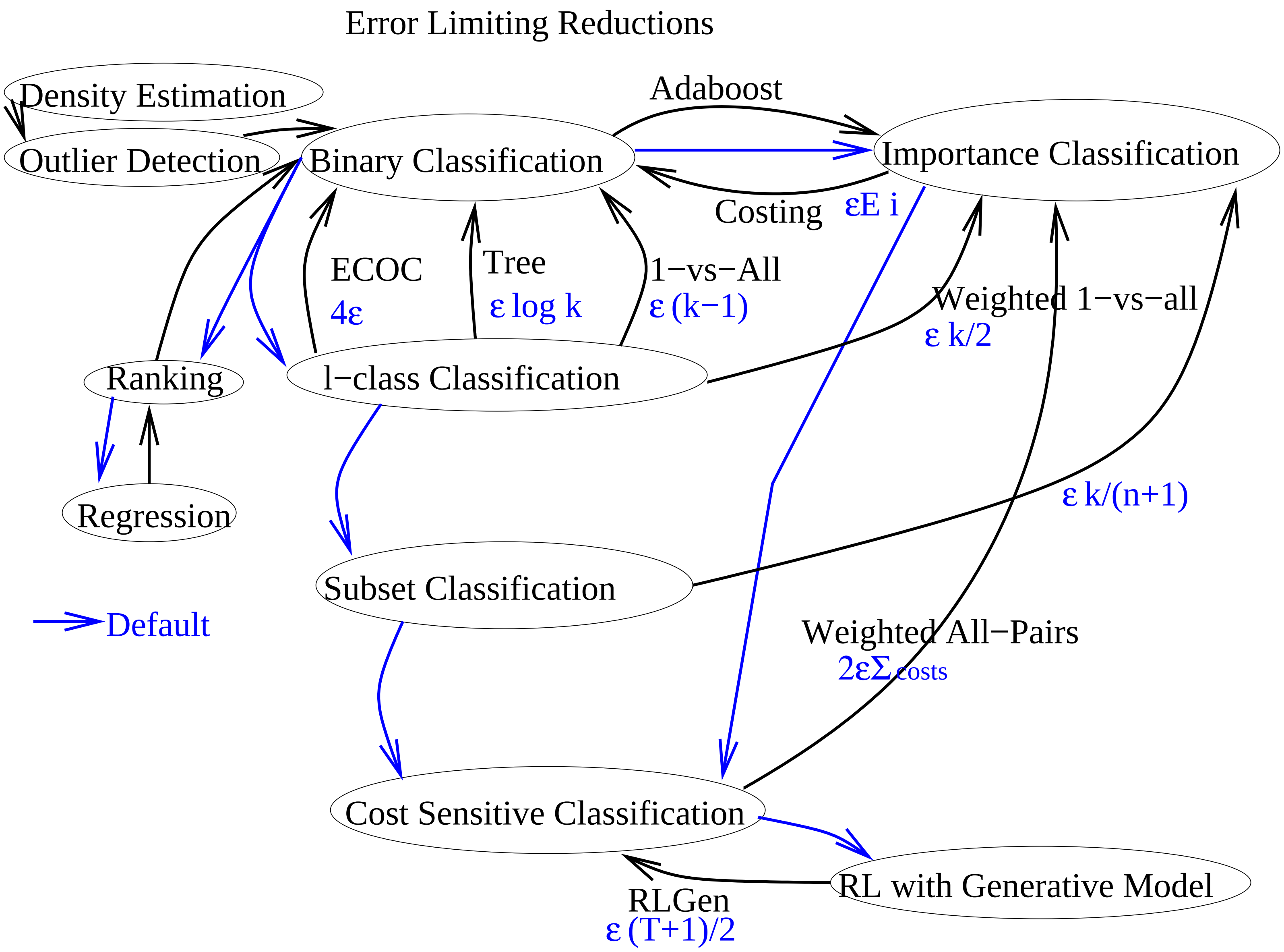
2. The Evidence.

3. The Weighted All Pairs algorithm.

What is your evidence threshold?

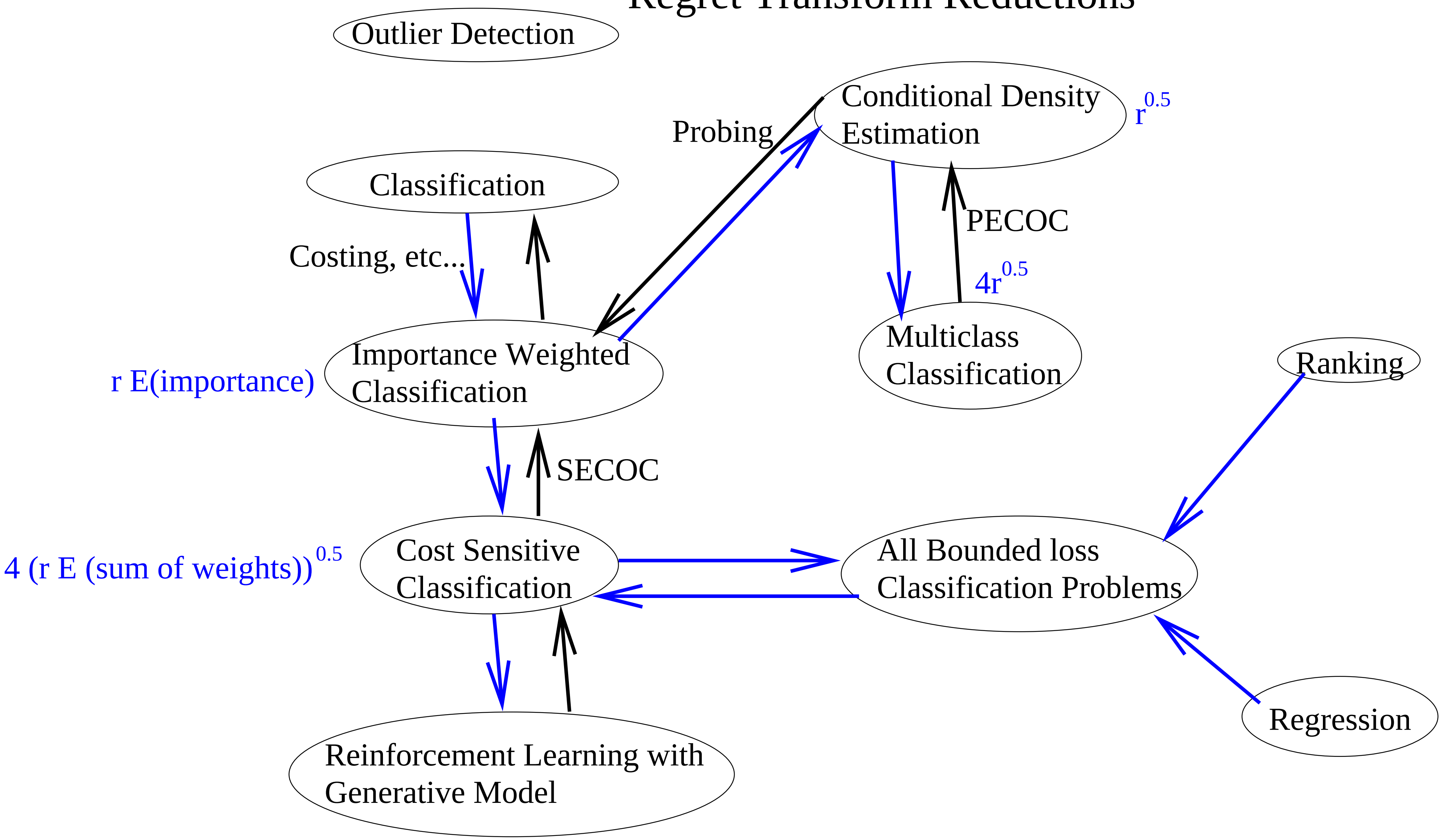
1. Basic theory? (How basic?)

2. Experiments? (How many?)



Error Limiting Reductions
Density Estimation
Outlier Detection
Binary Classification
Adaboost
Importance Classification
Costing
εE i
ECOC
4ε
Tree
ε log k
1−vs−All
ε (k−1)
Weighted 1−vs−all
ε k/2
Ranking
l−class Classification
Regression
ε k/(n+1)
Subset Classification
Default
Weighted All−Pairs
2εΣ costs
Cost Sensitive Classification
RL with Generative Model
RLGen
ε (T+1)/2

Regret Transform Reductions



Quiz: Which of the following prediction problem types have been empirically shown solvable by binary classifiers?

1. Importance Weighted classification ($E_{x,y,i\sim D}[iI(h(x) \neq y)]$)?

2. Multiclass classification ($E_{x,y\sim D}[I(h(x) \neq y)]$)?

3. Probabilistic Classification/Regression ($E_{x,y\sim D}(h(x) - y)^2$)?

4. Cost sensitive Classification ($E_{x,\vec{c}}\left[c_{h(x)}\right]$)?

5. Reinforcement learning ($E_{(s,a,r)^T\sim D,h}\left[\sum_{t=1}^{T} r_t\right]$)?

Trick Question: Which of the following prediction problem types have been empirically shown solvable by binary classifiers?

1. Importance Weighted: "Costing" ZLA ICDM 2003 (others).

2. Multiclass: "ECOC" DB JAIR 1995, "loss" ASS JMLR 2000, "PECOC" LB COLT 2005, "WOA" BLZ AAAI 2005.

3. Probabilistic Classification/Regression: "Probing" LZ AIStat 2005.

4. Cost-Sensitive Classification: "WAP" BDHLZ ICML 2005

5. Reinforcement Learning, only partially: "PSDP" BKNS 2003, "Searn" DLM 2006, BA NIPS 2005.

What do these statements have in common?

1. This laptop is a Turing machine.

2. Planets move according to Newtonian mechanics.

3. Any classification problem can be solved.

What do these statements have in common?

1. This laptop is a Turing machine.

2. Planets move according to Newtonian mechanics.

3. Any classification problem can be solved.

They are all false, but useful abstractions.

Outline

1. What is a learning reduction?

2. The Evidence.

3. The Weighted All Pairs algorithm.

Importance Weighted Classification

- Problem: A measure D on $X \times \{0,1\} \times [0,\infty)$ where X is an arbitrary space.

- Classifier: $c : X \to \{0,1\}$ = predictor

- Given $S = (X \times \{0,1\}, [0,\infty))^*$ find classifier c with small importance weighted loss

$$e(D,c) = E_{(x,y,i)\sim D}[iI(c(x) \neq y)]$$

Cost Sensitive Classification

- Problem: A measure D on $X \times [0, \infty)^k$ where X is an arbitrary space.

- Multiclass Classifier: $h : X \to \{1, ..., k\}$ = predictor

- Given $S = (X \times [0, \infty)^k)^*$ find classifier with small expected loss

$$l(D, h) = E_{(x,\vec{c}) \sim D}[c_{h(x)}]$$

Algorithm: Weighted All Pairs

WAP-Train(Cost sensitive S, IW Binary learner B)

1. Let $v_i(x, \vec{c}) = \int_0^{c_i} \frac{1}{|\{j | c_j \leq t\}|} dt$

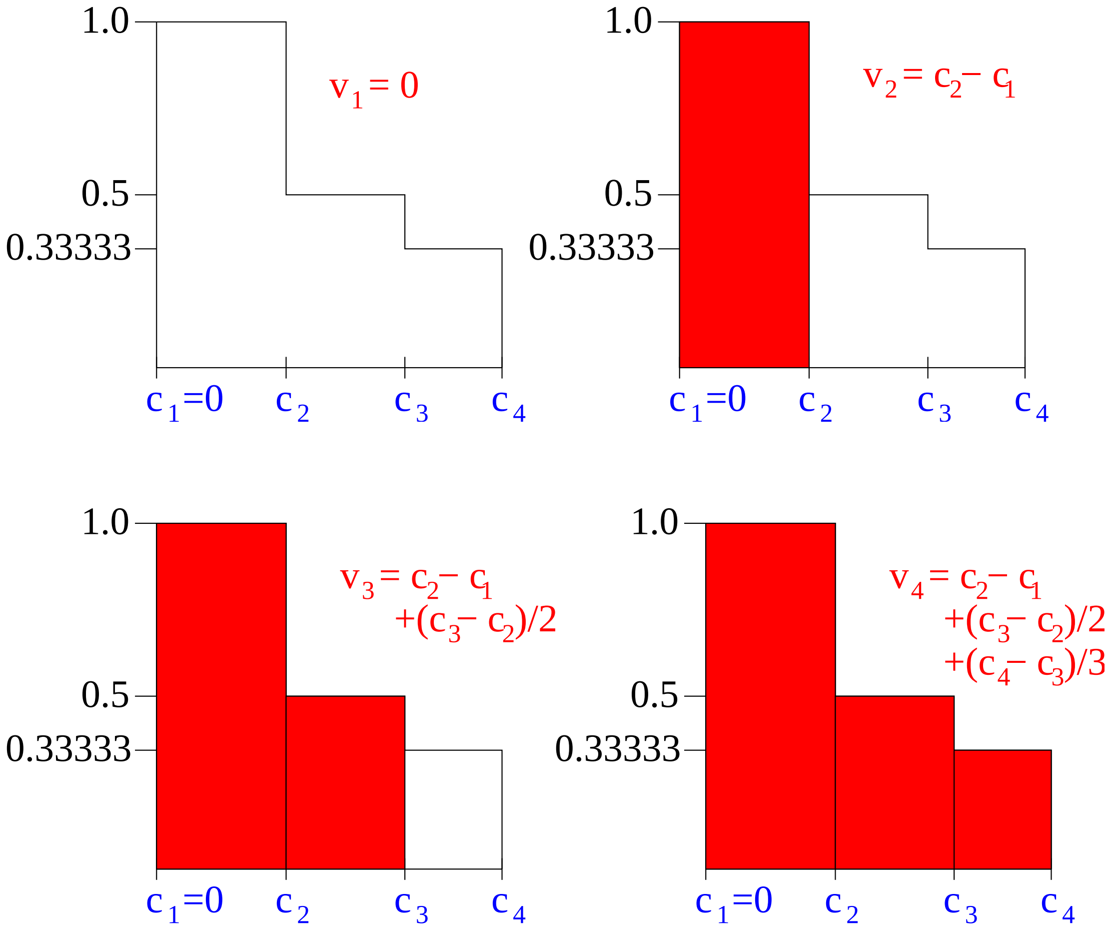
2. $S' = \{(\langle x, i, j \rangle, I(c_i \leq c_j), |v_j - v_i|) : (x, \vec{c}) \in S$ and $i, j \in \{1, k\}\}$

3. return $c = B(S')$

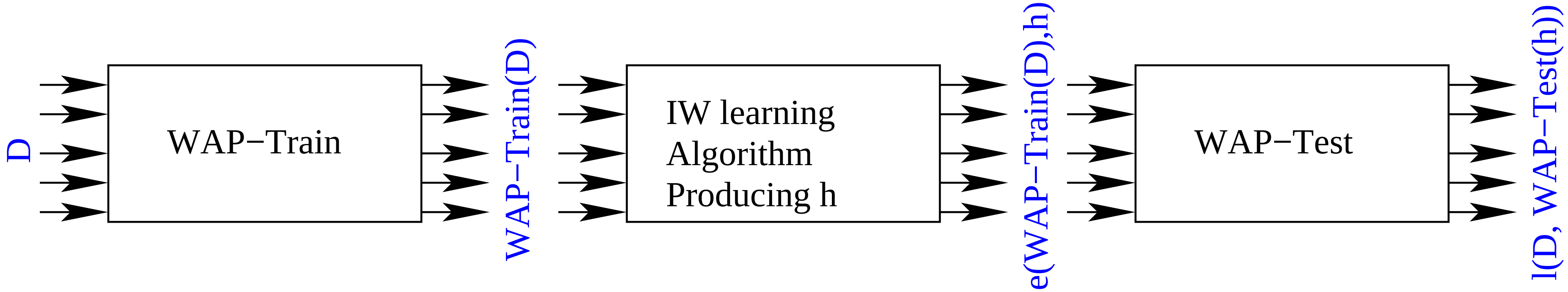
WAP-Test(Classifier c, example x)

1. return $\arg\max_{i \in \{1,k\}} |\{j : c(x, i, j) = 1\}|$

What is $v_i(x, \vec{c}) = \int_0^{c_i} \frac{1}{|\{j|c_j \le t\}|} dt$?



Main Theorem



Theorem: For all D, for all learned h,

$$l(D, \text{WAP-test}(h)) \leq 2e(\text{WAP-train}(D), h)$$

Lower bound: And no choice of v can tighten the above by more than a factor of 4.

Searn

Spanish Named Entity Recognition (1517 test)

	CRF(300)	SVM(300)	SVMISO(300)	Searn(300)	Searn(8324)
Acc.	94.83	94.94	94.9	95.01	97.67

(Multidigit) Handwriting Recognition (5500 or 600 test)

Training set size	Perceptron	Logistic	SVM	M^3N	Searn
600 train	65.56	68.65	82.63	87.5	88.2
5500 train	70.05	72.10	82.52	?	90.91

Joint sequence labeling (8936 train, 2012 test)

	Inc. P.	LASO	Factored CRF	CRF	Searn
Joint Accuracy	93.12	95.12	96.48	?	96.81
Chunk F-score	92.44	93.49	93.87	94.77	94.36
Parts of speech	98.05	98.90	98.92	?	99.07

Some extra discussion at http://hunch.net

For papers, search for John Langford