JESSE’S RESEARCH

Jesse Davis
jesse.davis@cs.kuleuven.be
https://dtai.cs.kuleuven.be/sports/

Research Program: Jesse Davis

Technology Push: Significantly advance the state of the art in machine learning

Applications drive innovation in machine learning

Anticipate scientific advances needed to address applications

Application Pull: Use machine learning to address significant problems in health, sports, and their intersection

ML Group Research Goals: Desired Solution Characteristics

Is distance between players important?

Attacker

Location: (30,100)

Prefers right foot

Tired?

$$\mathrm{dist}(p1,p2) < 2\mathrm{m} \wedge \mathrm{pr}(\mathrm{p1}=\mathrm{tried}) > 0.8 \wedge \mathrm{prefRt}(\mathrm{p1}) \Rightarrow \mathrm{dribbleRt}(\mathrm{p1})$$

1. Represent discrete and continuous attributes
2. Model uncertainty
3. Capture important relationships
4. Incorporate domain knowledge
5. Produce interpretable output

Part I: Learning Probabilistic (Relational) Models

Outline

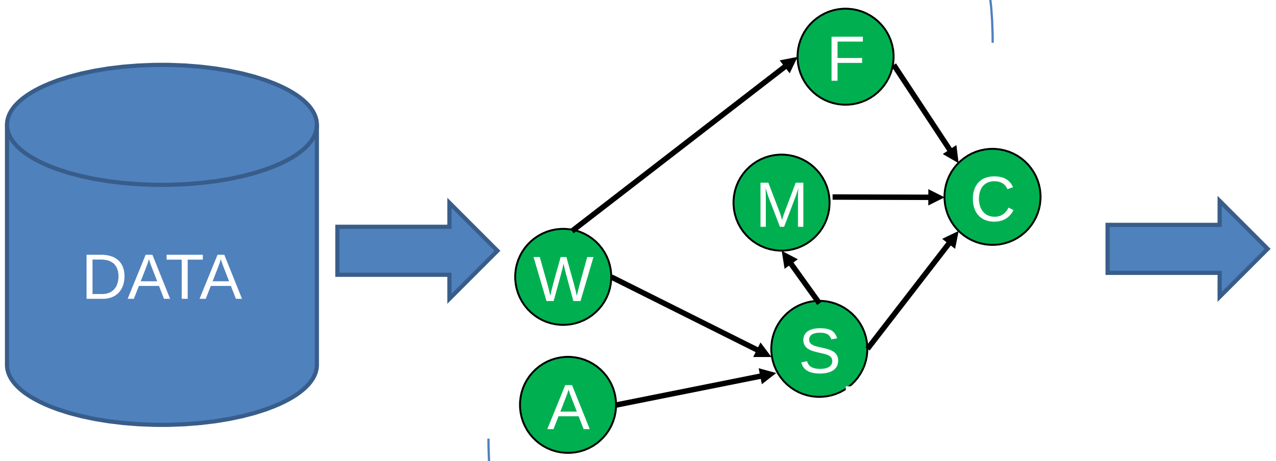
- Learning while accounting for model use
- Learning the structure of propositional probabilistic graphical models
- Learning the structure of probabilistic relational models
- Deep transfer: Transferring across entirely different domains

Outline

- Learning while accounting for model use
- Learning the structure of propositional probabilistic graphical models
- Learning the structure of probabilistic relational models
- Deep transfer: Transferring across entirely different domains

Motivation

Learning is hard and requires lots of approximations



P(C | F, M)
P(M | C, W)
…

Inference is hard and model has big effect on inference

Problem: Learning and inference treated separately, but really should consider model use at learning stage

Three Directions

- Prediction with learned models that considers energy constraints [Verachtert et al. IJCAI’16]
- Learning tractable for Markov logic networks [Van Haaren et al. MLJ’16]
- Expanding the set of queries that can be answered efficiently [Bekker et al. NIPS’15]

Motivation

- Learned models are increasingly deployed on portable devices with resource constraints
  - Battery
  - Memory
  - Etc.
- Goal: Prediction with learned models must account for these constraints
  - Focus NOT on training efficiency: Done off line

Prediction with Naïve Bayes

$$\text{Argmax}_c \log(C{=}c) + \sum_i \log P(A_i = a_{i,j} \mid C{=}c)$$

A_1	A_2	A_3	A_4	A_5	A_6	A_7	. . .	A_n	C

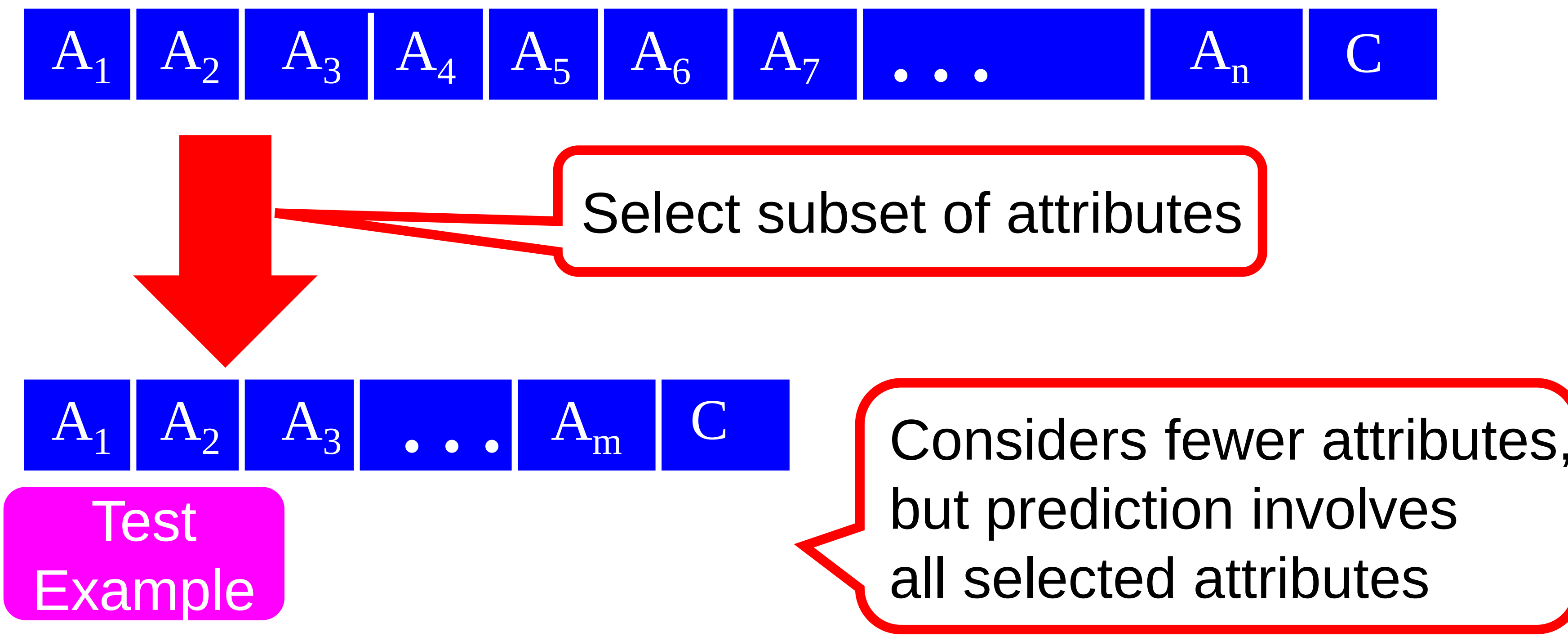
Test Example

Prediction based on all attributes

Question: Can we improve prediction efficiency?

Idea 1: Feature Selection

$$\text{Argmax}_c \log(C=c) + \sum_i \log P(A_i = a_{i,j} \mid C=c)$$



Question: Can we do better?

Our Idea:
Naïve Bayes with Stop Points

Stop Point: 0.85, 0.12

Stop Point: 0.88, 0.10

Stop Point: 0.91, 0.11

A_1	. . .	A_i	. . .	A_j	. . .	A_k	. . .	A_n	C

Test Example 1

Check prediction $P(C=1 \mid A_1, \ldots, A_i)$

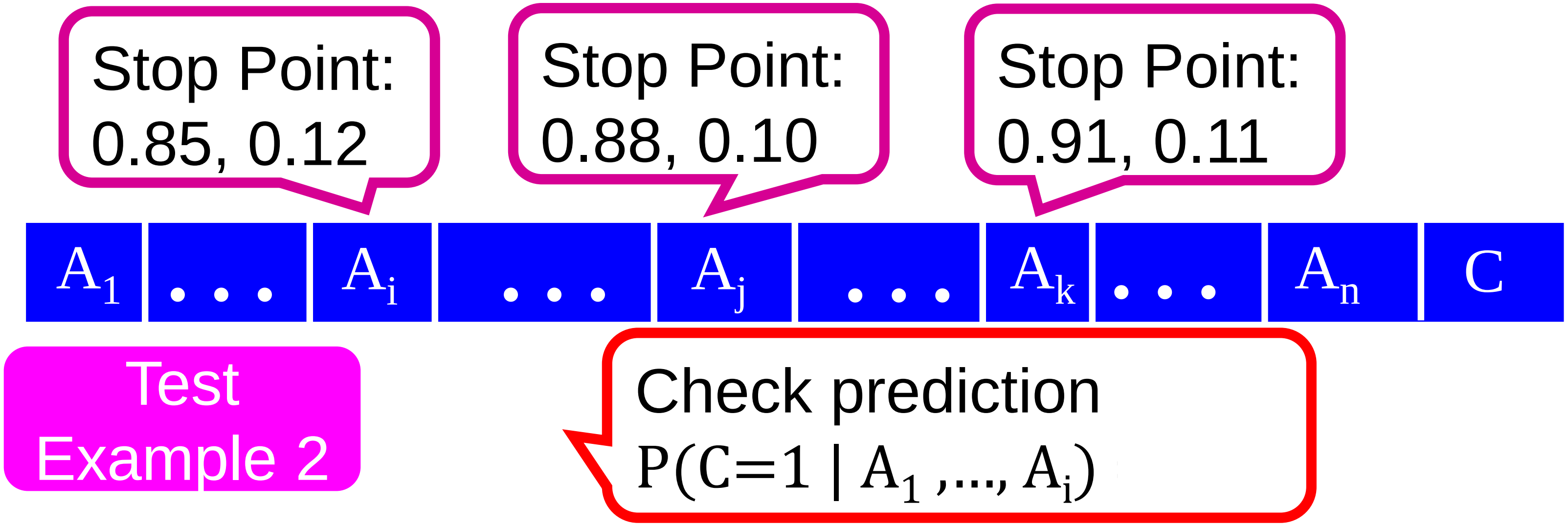
Stop inference

IF $P(C=1 \mid A_1,\ldots, A_i) > 0.85$ THEN predict $C=1$

ELSE IF $P(C=1 \mid A_1,\ldots, A_i) < 0.12$ THEN predict $C=0$

ELSE continue observing features until next stop point

Our Idea:
Naïve Bayes with Stop Points



IF $P(C=1 \mid A_1, \ldots, A_i) > 0.85$ THEN predict $C=1$

ELSE IF $P(C=1 \mid A_1, \ldots, A_i) < 0.12$ THEN predict $C=0$

ELSE continue observing features until next stop point

Continue inference

Our Idea:
Naïve Bayes with Stop Points

Stop Point: 0.85, 0.12

Stop Point: 0.88, 0.10

Stop Point: 0.91, 0.11

A_1	. . .	A_i	. . .	A_j	. . .	A_k	. . .	A_n	C

Test Example 2

Check prediction $P(C=1 \mid A_1, \dots, A_k)$

IF $P(C=1 \mid A_1, \dots, A_j) > 0.88$ THEN predict $C=1$

ELSE IF $P(C=1 \mid A_1, \dots, A_j) < 0.10$ THEN predict $C=0$

Stop inference

ELSE continue observing features until next stop point

Number of observed attributes selected per example

Intuition: Stop if prediction “confident enough”

Adding Stop Points

- Stop point (k, u, l) checks at attribute k if
  - $P(C=1 \mid A_1, \ldots, A_k) > u$: stop and predict $C=1$
  - $P(C=1 \mid A_1, \ldots, A_k) < l$: stop and predict $C=0$

- Order features from most to least informative
- Add a stop point at attribute k if u and l exist:
  - S% of examples are stopped
  - Accuracy on stopped examples higher than accuracy on
    - Stopped examples if all attributes observed
    - All examples if all attributes observed

Empirical Evaluation

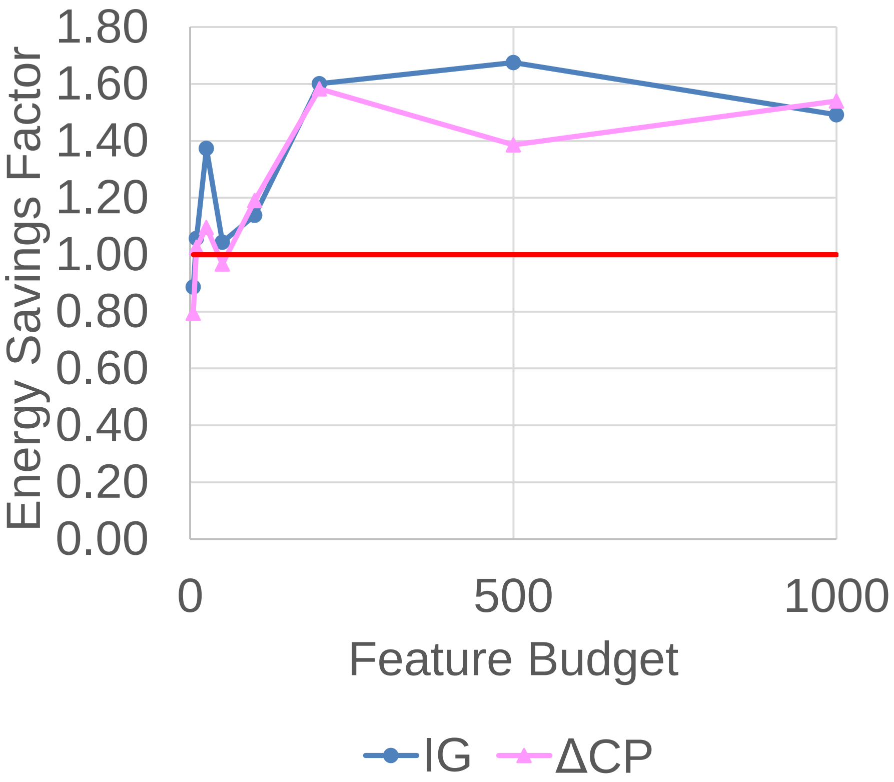
- Question: How does our approach compare to static orderings from standard feature selection?
  - IG: Information gain
  - ΔCP: Difference in conditional probabilities
  - Three others (omitted from graphs for readability)

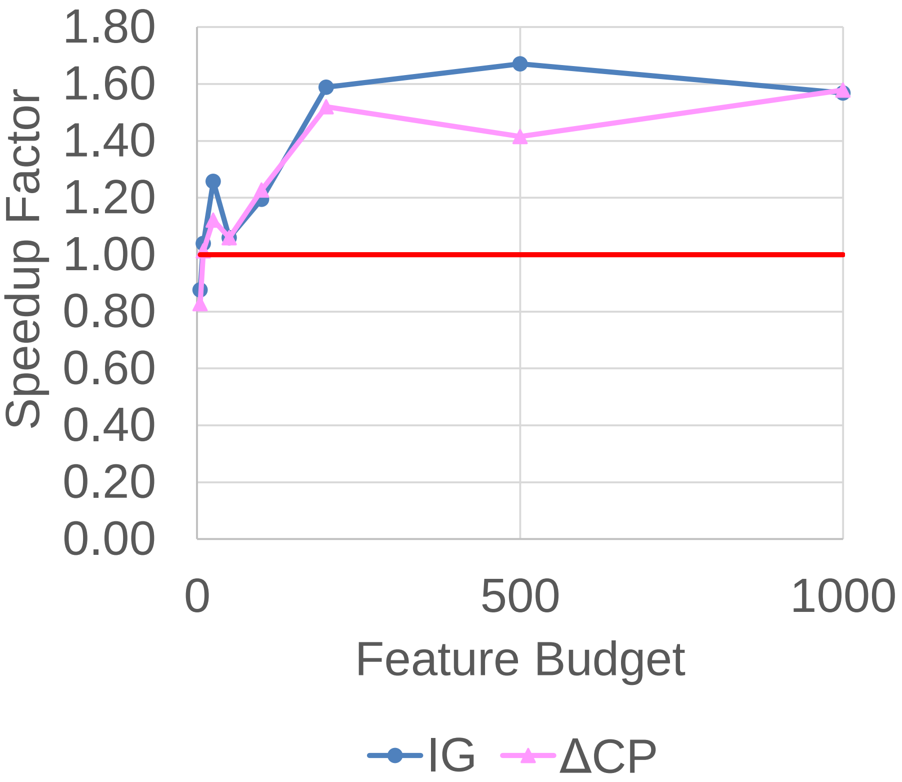
- Give each approach the same feature budget
  - Energy improvement factor
  - Speed up as proxy for energy usage
  - Weighted accuracy

Data and Methodology

- Evaluation on seven data sets
  - Attributes: 1,000 to 139,000
  - Examples: 2,500 to 800,000
- 10 random splits: 40% train, 20% tune, 40% test
- Energy measurements: Raspberry Pi
  - Gives a controlled environment
  - Use multimeter to measure energy consumption for prediction

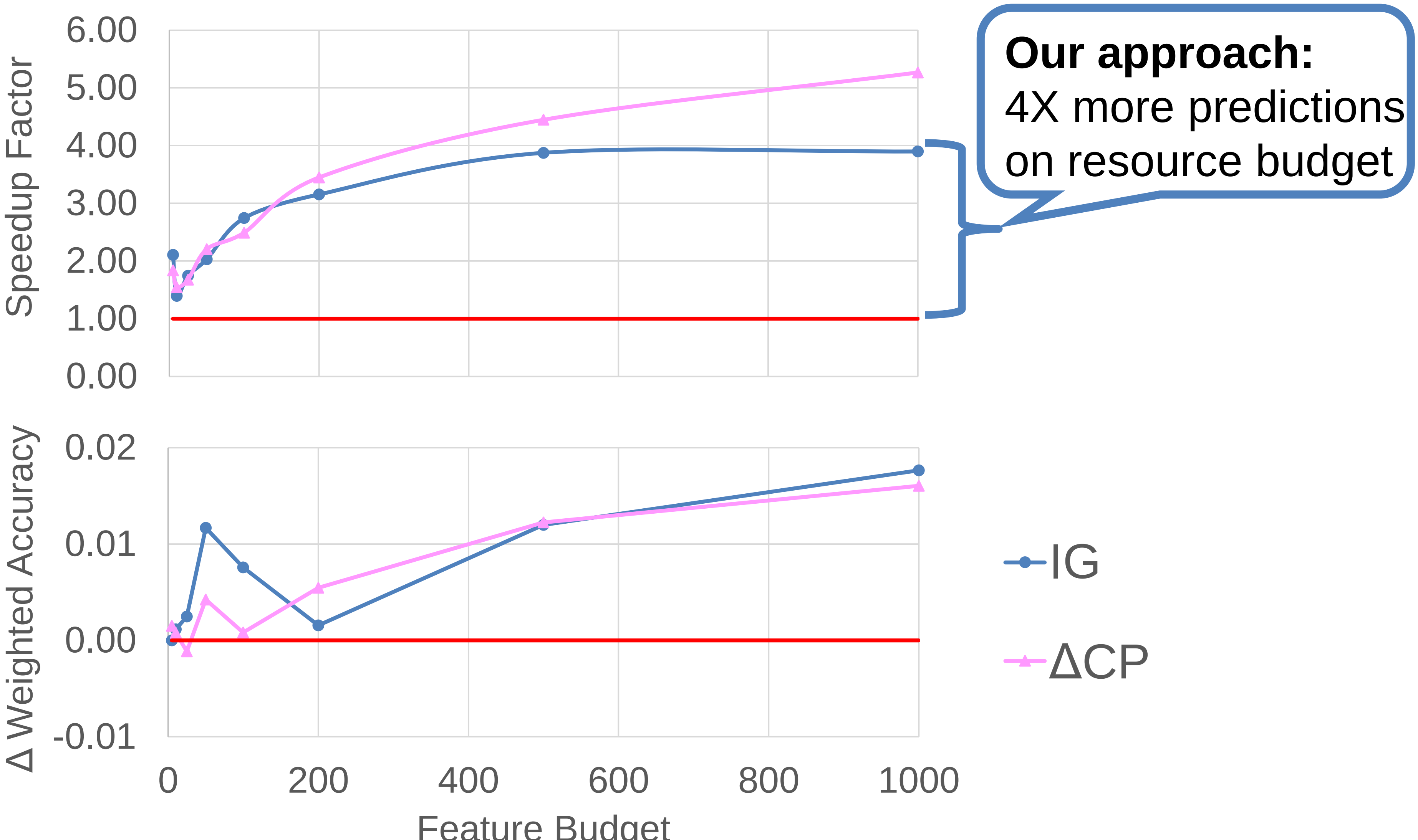
IMDB.drama: Energy Measurements



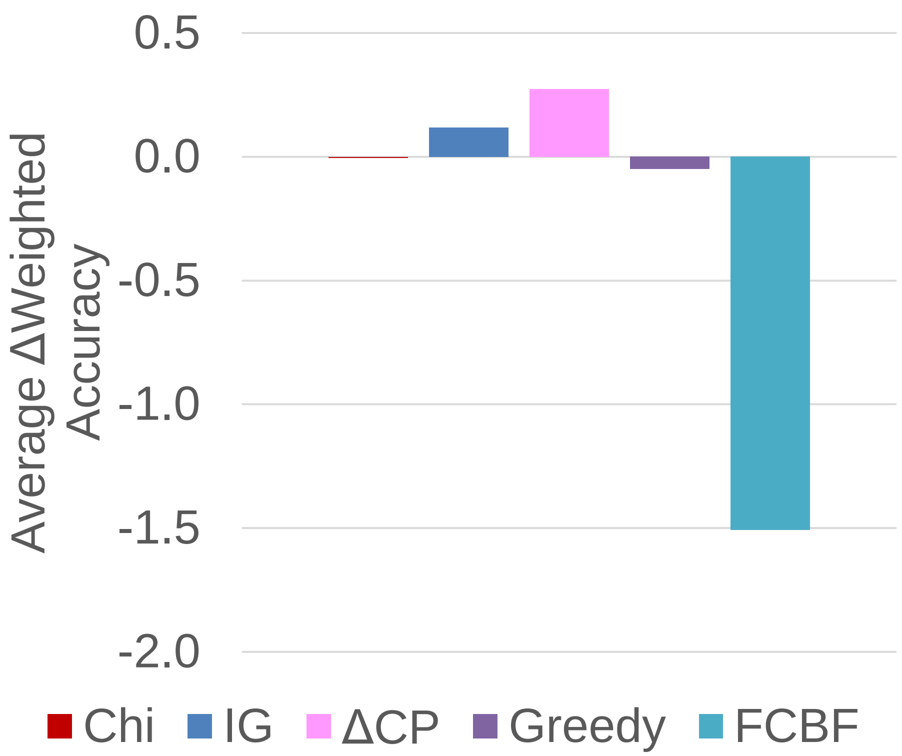


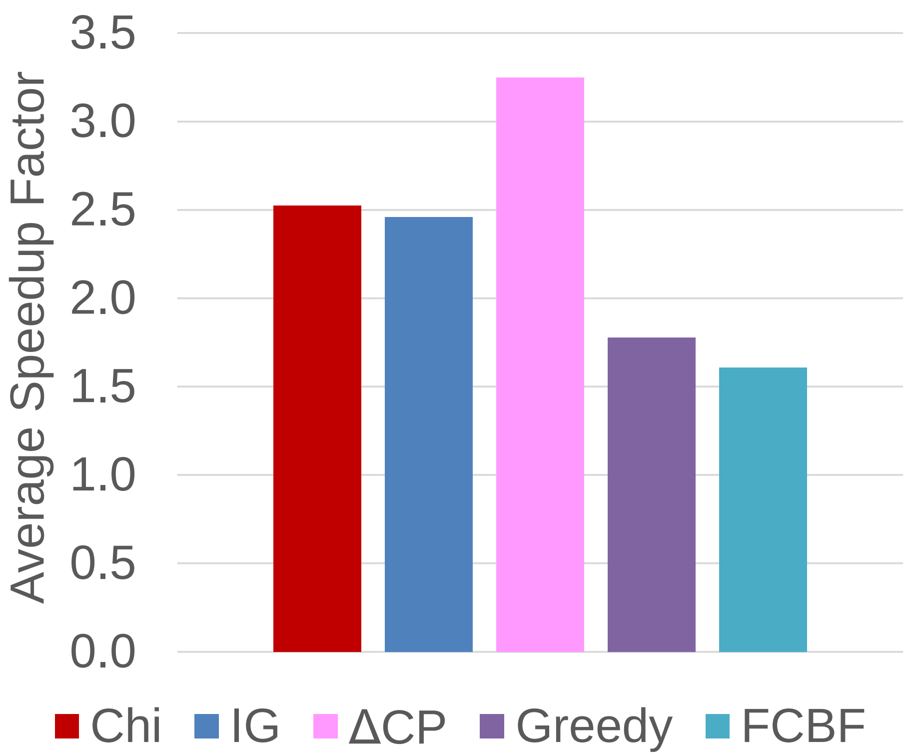
CPU time is a good proxy for energy usage

RCV: Speedup and Weighted Accuracy VS. Feature Budget



Summary of Results





Performance of best static model vs. dynamic model with the same feature budget

Outline

- Learning while accounting for model use
- Learning the structure of propositional probabilistic graphical models
- Learning the structure of probabilistic relational models
- Deep transfer: Transferring across entirely different domains

Problem Definition

Training Data

F	W	A	S	C
T	T	T	F	F
F	F	T	T	F
F	T	T	F	F
T	F	F	T	T
F	F	F	T	T

Goal:

Represent probability distribution over different configurations the variables can take on

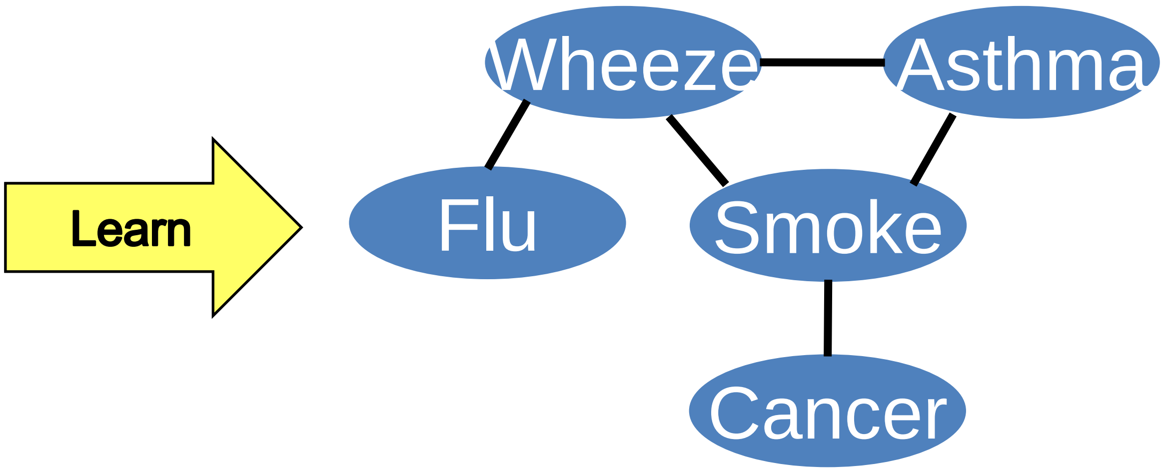
Applications: Diagnosis, prediction, recommendations, and much more!

Problem Definition

Training Data

F	W	A	S	C
T	T	T	F	F
F	F	T	T	F
F	T	T	F	F
T	F	F	T	T
F	F	F	T	T

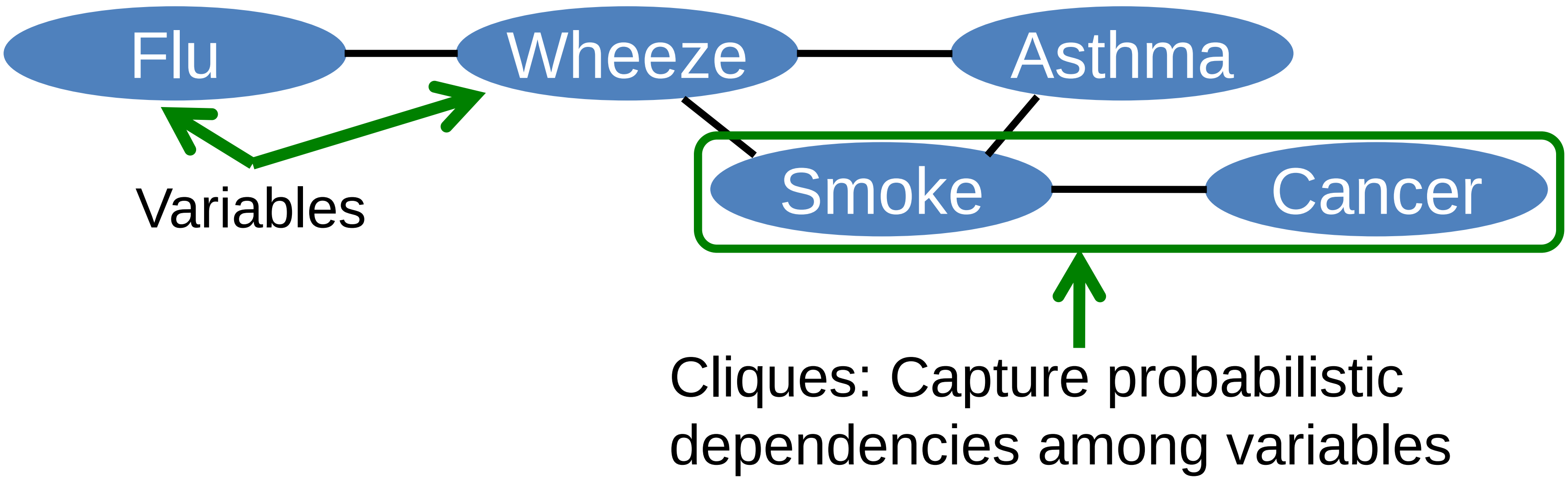
Markov Network Structure



$$P(F,W,A,S,C)$$

Applications: Diagnosis, prediction, recommendations, and much more!

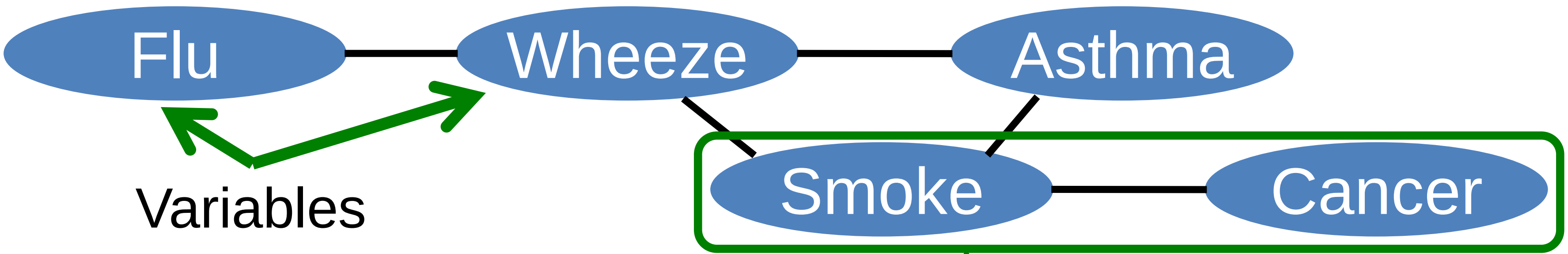
Markov Networks: Representation



Undirected, graphical model that represents a joint distribution over a set of variables

(aka Markov random fields, Gibbs distributions, log-linear models, exponential models, maximum entropy models)

Markov Networks: Representation



Represents the following distribution

$$P(x) = \frac{1}{Z} \prod_c \Phi_c(x_c)$$

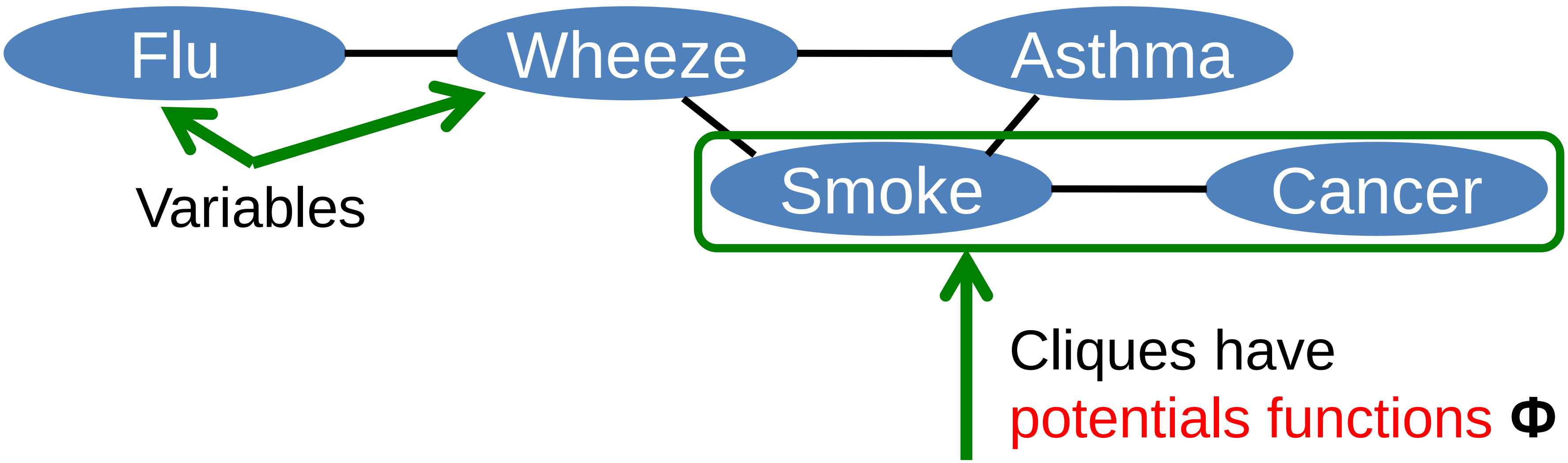
Z is the normalization constant

$$Z = \sum_x \prod_c \Phi_c(x_c)$$

Cliques have potentials functions **Φ**

Smoke	**Cancer**	**Φ(S,C)**
False	False	4.5
False	True	4.5
True	False	4.5
True	True	2.7

Markov Networks: Representation



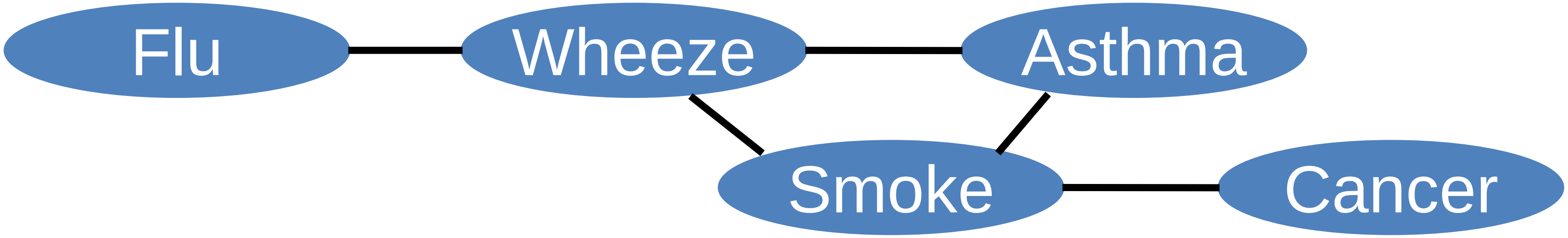
Convert potentials to features

Smoke	Cancer	Φ(S,C)
False	False	4.5
False	True	4.5
True	False	4.5
True	True	2.7

1.5 Smoke ∧ Cancer

Weight Feature

Markov Networks:
Log-Linear Representation



$$P(x) = \frac{1}{Z} \exp\left(\sum_i \overbrace{\boxed{w_i}}^{\text{Weight of Feature } i} \overbrace{\boxed{f_i(x)}}^{\text{Feature } i}\right)$$

Markov Networks: Learning

1.5 Smoke $\wedge$ Cancer

Weight of Feature *i* Feature *i*

$$P(x) = \frac{1}{Z} \exp\left(\sum_i \boxed{w_i} \boxed{f_i(x)} \right)$$

Two Learning Tasks

Weight Learning

- Given: Features, Data
- Learn: Weights

Structure Learning

- Given: Data
- Learn: Features, Weights

Markov Networks: Weight Learning

- Maximum likelihood weights

$$\frac{\partial}{\partial w_i} \log P_w(x) = \boxed{n_i(x)} - \boxed{E_w\left[n_i(x)\right]}$$

No. of times feature *i* is true in data

Expected no. times feature *i* is true according to model

Slow: Requires inference at each step

- Pseudo-likelihood

$$PL(x) \equiv \prod_i P(x_i \mid neighbors(x_i))$$

No inference: More tractable to compute

Why Is Inference Hard?

Exponentially Many States

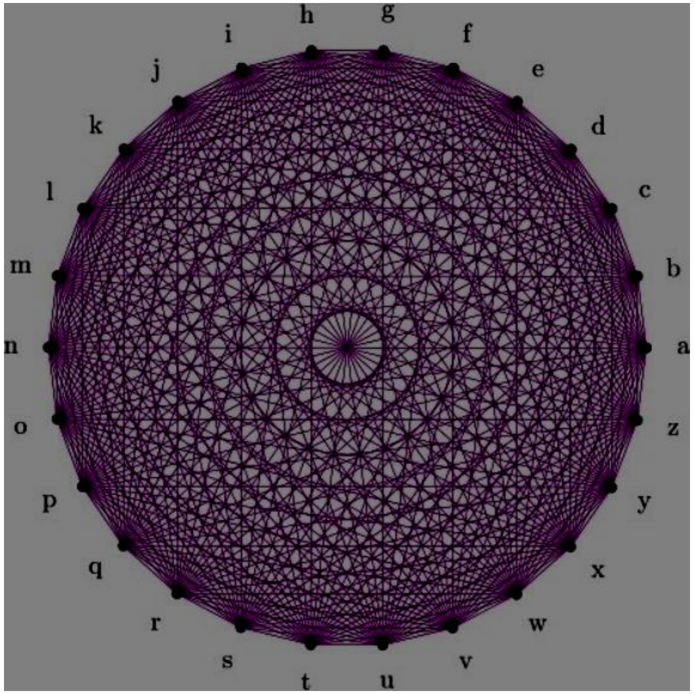
F	W	A	S	C	Weight
F	F	F	F	F	
F	F	F	F	T	
F	F	F	T	F	
F	F	F	T	T	
F	F	T	F	F	
…	…	…	…	…	…
T	T	T	T	T	

$$P(x) = \frac{1}{\boxed{Z}} \exp\left(\sum_i w_i \; f_i(x)\right)$$

Computing Z requires summing over all possible states!

Inference Problem Highlighted

- Example: Smokes(X) ∧ Friends(X,Y) ⇒ Asthma(Y)
  - People: 26 (a,…,z)
  - Variable: 728



- Real-world data
  - People: 1,000
  - Variables > 1,000,000

Markov Network Structure Learning

- Goal find the features

- Broadly speaking, two standard approaches:
  - Search through space of possible models (subproblem, search to generate features

  - Local models: Use classifiers in a clever way

Search-Based Structure Learning

[Della Pietra et al., 1997]

- Given: Set of variables = {**F, W, A, S, C**}
- At each step

Current model = {F, W, A, S, C, S $\wedge$ C}

Candidate features:
Conjoin variables to features in model

{**F $\wedge$ W, F $\wedge$ A, …, A $\wedge$ C, F $\wedge$ S $\wedge$ C, …, A $\wedge$ S $\wedge$ C**}

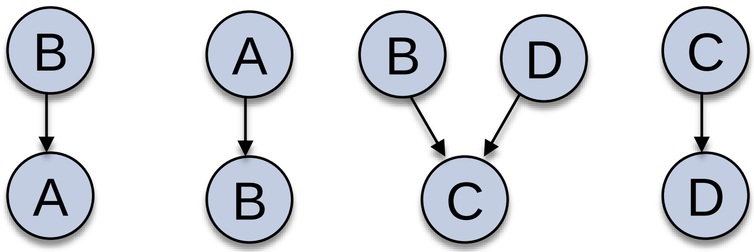
Select best candidate

New model = {F, W, A, S, C, S $\wedge$ C, F $\wedge$ W}

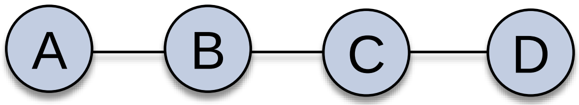
Iterate until no feature improves score

Local Model Approach Overview

- Step 1: Learn "local models" to predict each variable given the others



- Step 2: Combine local models into global model



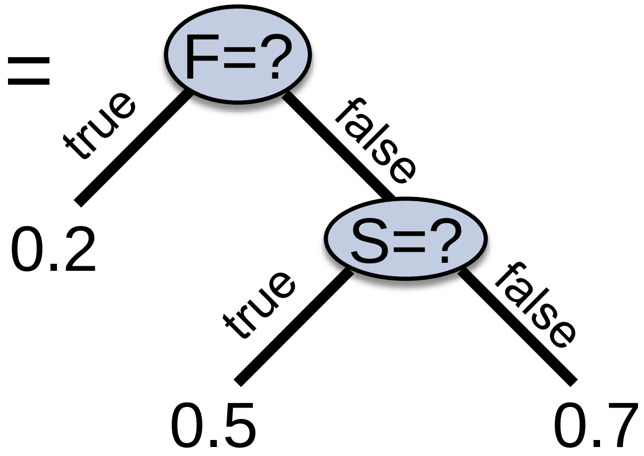
- Step 3: Learn weights

+Avoid running weight learning multiple times

DTSL: Decision Tree Structure Learning [Lowd and Davis, 2014]

- Given: Set of variables= {**F, W, A, S, C**}
- Do: Learn decision tree to predict each variable

P(C|F,S) =

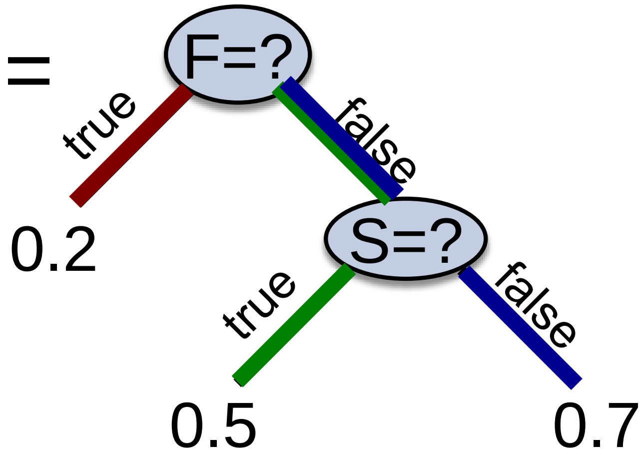


P(F|C,S) = …

DTSL: Feature Construction

- Construct one feature for each root-to-leave path in a tree

$P(C|F,S) =$



$P(F|C,S) = \cdots$

- Features include

$F \wedge C$
$F \wedge \neg C$

$\neg F \wedge S \wedge C$
$\neg F \wedge S \wedge \neg C$

$\neg F \wedge \neg S \wedge C$
$\neg F \wedge \neg S \wedge \neg C$

Motivation

- Search-based approaches
  - Slow because due to lots of weight learning
  - Generate long features in data-driven way

- Local-modal approaches
  - Fast because weights learned only once
  - Slow if many examples or variables

- Goal: Combine benefits of each approach

GSSL: Generate Select Structure Learn [Van Haaren & Davis, tbd]

- Two step process
  - Step 1: Generate features
  - Step 2: Select features

- Benefits include
  - Fast, directed approach to feature generation
  - Only run weight learning once

Step 1: Initialize by Converting Examples to Features

F	W	A	S	C
T	T	T	F	F
T	F	T	T	F
F	T	T	F	F
T	F	F	T	T
F	F	F	T	T

F1: $F \wedge W \wedge A$

F2: $F \wedge A \wedge S$

F3: $W \wedge A$

F4: $F \wedge S \wedge C$

F5: $S \wedge C$

Step 1: Feature Generation

<u>Base Features</u>

F1: F ∧ W ∧ A

F2: F ∧ A ∧ S

F3: W ∧ A

F4: F ∧ S ∧ C

F5: S ∧ C

<u>Generated Features</u>

F ∧ A	3
S ∧ C	4
W	5
F ∧ W	1
…	…
C	4

<u>Repeat:</u>

1) Randomly select feature
2) Drop arbitrary number of variables
3) Add generalized feature to feature set

Step 2: Feature Selection

Generated Features	
F ∧ A	5
S ∧ C	4
W	5
~~F ∧ W~~	~~1~~
…	…
C	4

Prune →

Final Model	
2.3	F ∧ A
~~0.0~~	~~S ∧ C~~
-1.2	W
…	…
-2.1	C

1) Prune features generated fewer times than a threshold
2) Weight learning with L1 prior to enforce sparsity

GSSL Control Structure

Given: Example Set, Integer m, Threshold t

- Let BS = Example Set
- For i = 1 to m
  - Randomly pick feature from BS
  - Drop arbitrary number of variables, add new feature to FS
- Prune each feature generated less than t times
- Run L1 weight learning on remaining features

Empirical Evaluation

- Compared the following algorithms
  - BLM [Davis and Domingos, 2010]
  - L1 [Ravikumar et al., 2009]
  - DTSL [Lowd and Davis, 2014]
  - GSSL [Van Haaren and Davis, 2012]
- Compared on 20 different real-world domains
  - 1,600 to 290,000 train examples
  - 200 to 38,000 tune examples
  - 300 to 58,000 test examples
  - 16 and 1,500 variables

Note: Implementations and most datasets available:

http://alchemy.cs.washington.edu/papers/davis10a

Experimental Details

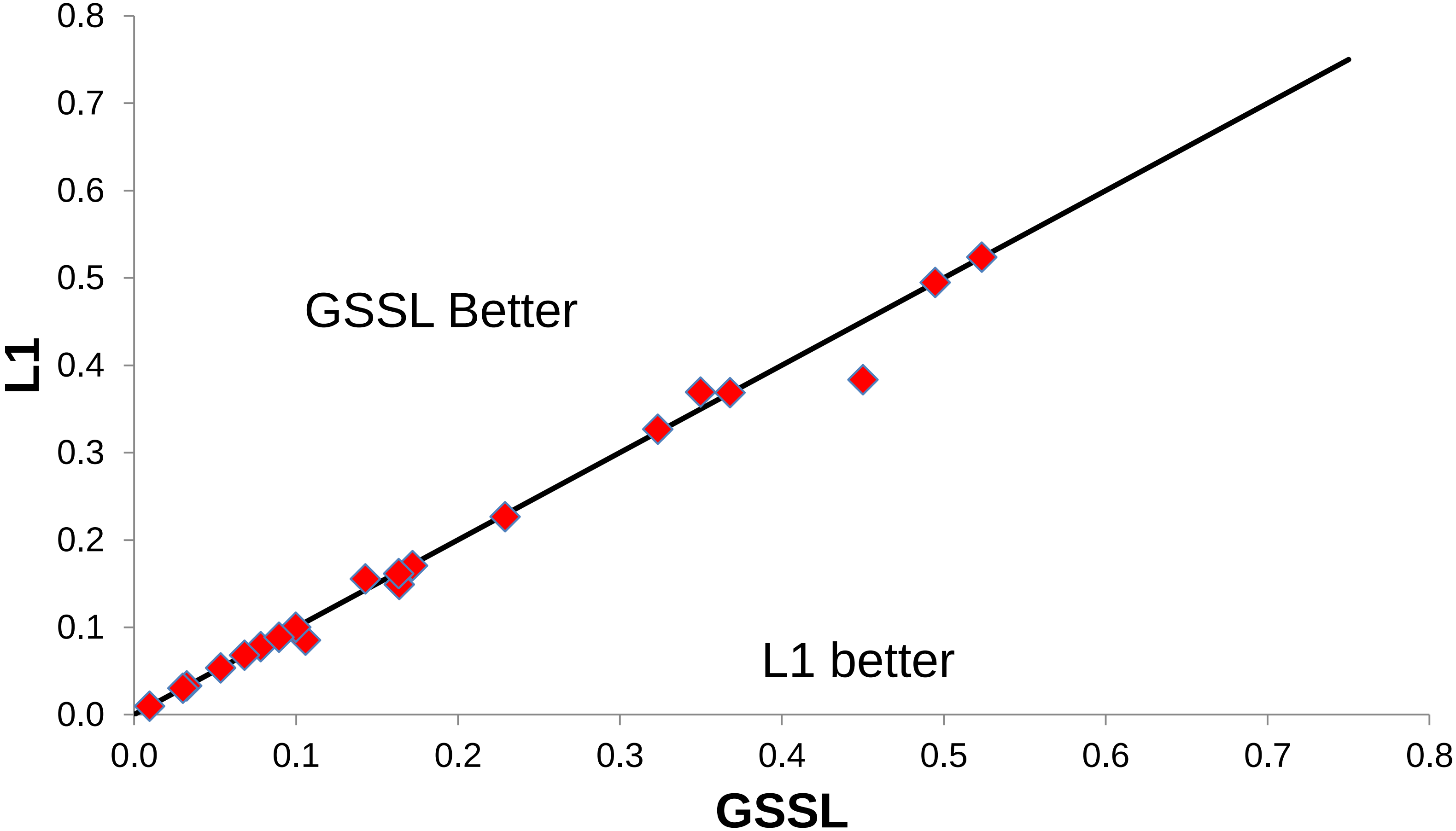
- Optimize pseudo-log-likelihood (PLL)
- Tried variety of parameters for each algorithm
- Use tune set PLL to pick best model
- Evaluation metrics
  - Accuracy: Conditional marginal likelihood

$$CMLL(x,e) = \sum_i \log P(X_i = x_i \mid E = e)$$

  - Speed: Run time

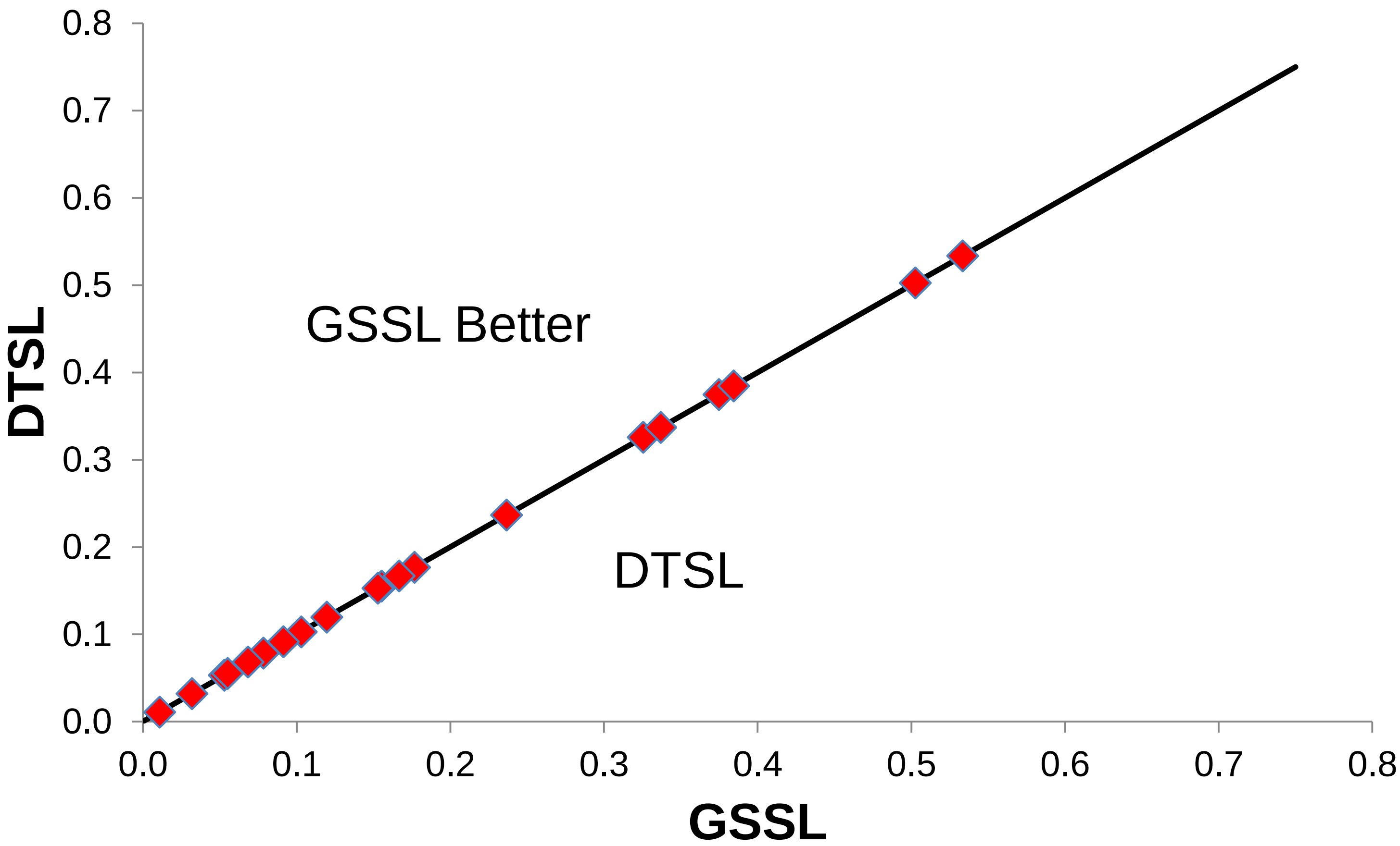
GSSL vs. L1

GSSL wins on 11 out of 20 domains

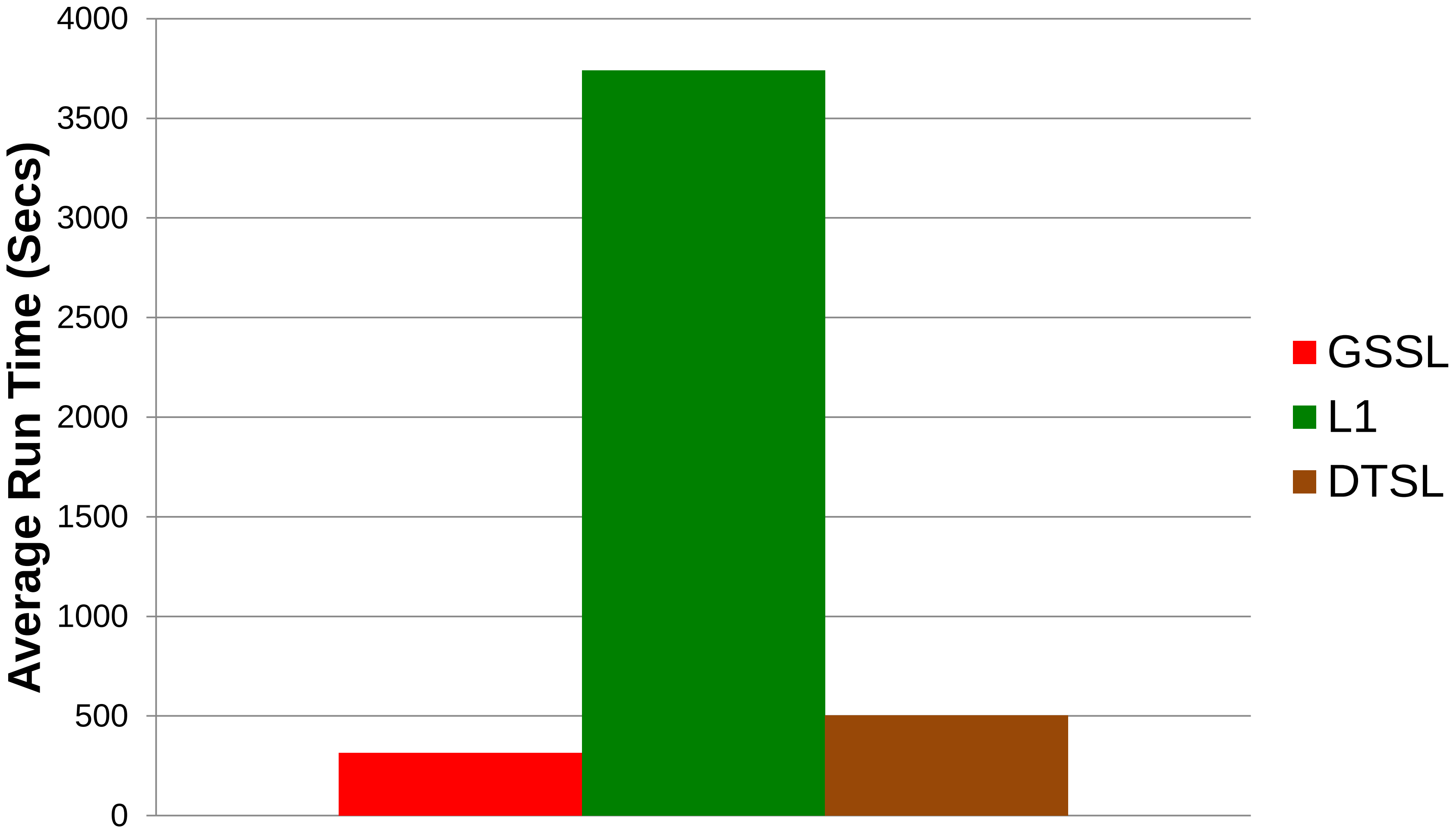


GSSL vs. DTSL

GSSL wins on 15 out of 20 domains



Run Time Comparison



Outline

- Learning while accounting for model use
- Learning the structure of propositional probabilistic graphical models
- Learning the structure of probabilistic relational models
- Deep transfer: Transferring across entirely different domains

Challenge: Complex Data

Patient

PID	Birthday	Gender
P1	2/2/63	M
P2	4/7/55	M

Drug

PID	Date	Medication	Dose
P1	5/1/02	Warfarin	10mg
P1	2/2/03	Terconazole	10mg

Diseases

PID	Date	Diag.
P1	2/1/01	Flu
P1	5/2/03	Bleeding

- Data are complexly structured
- Data are highly uncertain
- Etc.

Traditional Solution

Patient

PID	Birthday	Gender
P1	2/2/63	M
P2	4/7/55	M

Drug

PID	Date	Medication	Dose
P1	5/1/02	Warfarin	10mg
P1	2/2/03	Terconazole	10mg

Diseases

PID	Date	Diag.
P1	2/1/01	Flu
P1	5/2/03	Bleeding

Statistical Approach

+ Models uncertainty
- Ignores relations

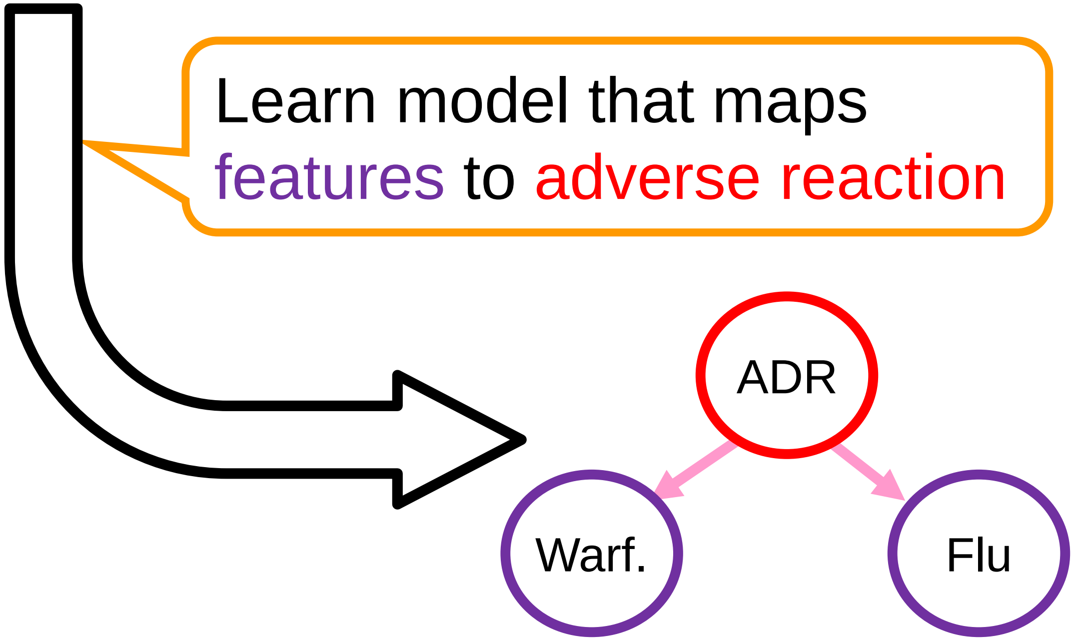
Logical Approach

+ Models relations
- Ignores uncertainty

Statistical Approach Overview

Data representation: i.i.d. vectors

Patient	Warfarin	Terconazole	…	Flu	ADR
P1	Yes	Yes		No	Yes
P2	No	No		Yes	No



Logical Approach Overview

Data representation: First-order logic

Drug

PID	Date	Medication	Dose
P1	5/1/02	Warfarin	10mg
P1	2/2/03	Terconazole	10mg

- **Constant:** Terconazole
- **Variable:** p
- **Literal:** Drug(P1, Terconazole)

Learn: Set of first-order logical rules

$$\mathrm{Drug}(\mathrm{p}, \mathrm{Terconazole}) \wedge \mathrm{Wt}(\mathrm{p}, \mathrm{w}) \wedge \mathrm{w} < 120 \Rightarrow \mathrm{ADR}(\mathrm{p})$$

Solution: Statistical Relational Learning

- Combine the statistical and logical approaches
- Intuition: Attach probabilities to first-order rules to capture uncertainty
- Example: Smoking causes cancer

$$\text{Smokes(person)} \Rightarrow \text{Cancer(person)} \quad : 0.15$$

VISTA: A SRL Framework

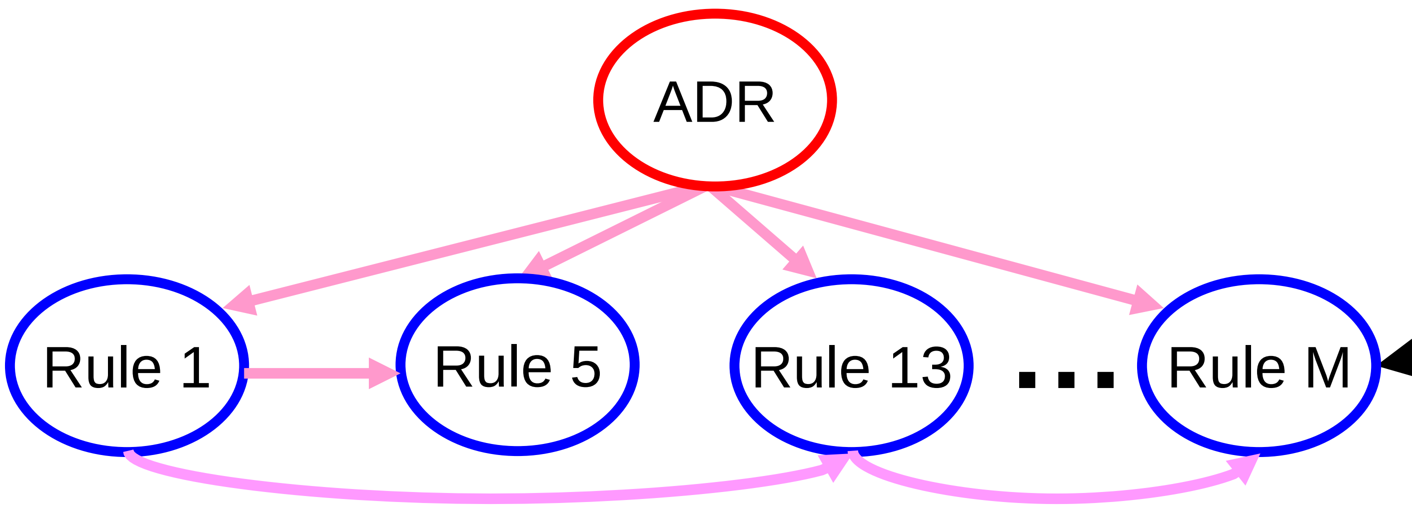
[Davis et al., IJCAI'07]

Integrates feature induction and model construction

- If-then rules capture implicit, relational features

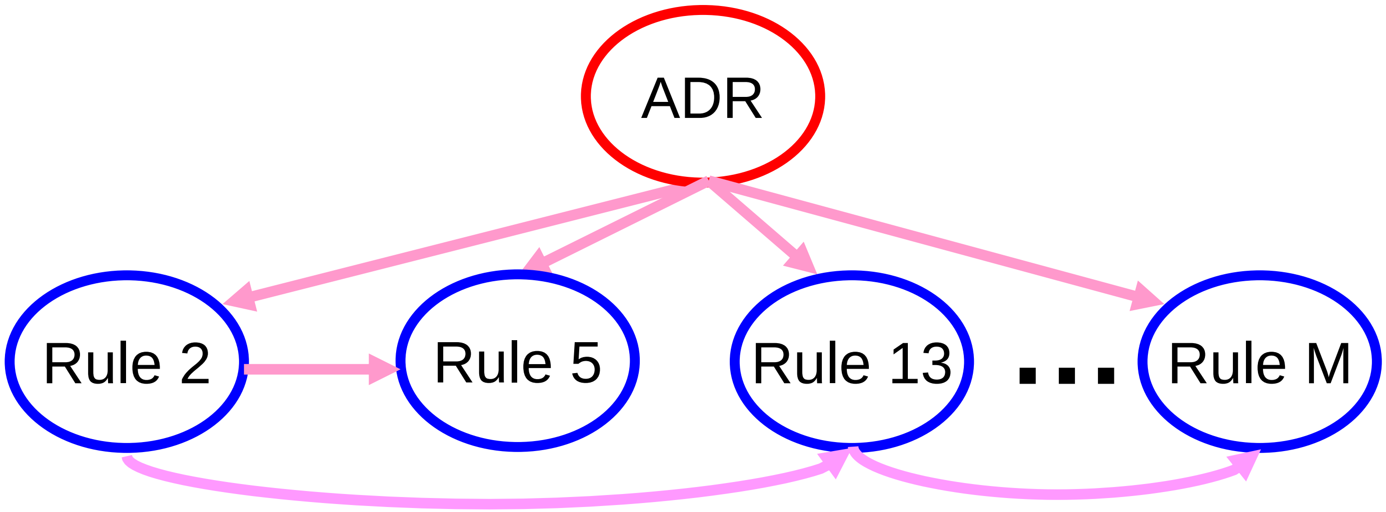
$$\boxed{\mathrm{Drug}(\mathrm{p},\mathrm{Terconazole}) \wedge \mathrm{Wt}(\mathrm{p},\mathrm{w}) \wedge \mathrm{w} < 120 \Rightarrow \mathrm{ADR}(\mathrm{p})}$$

- Rules become features in statistical model



VISTA: A SRL Framework

[Davis et al., IJCAI'07]

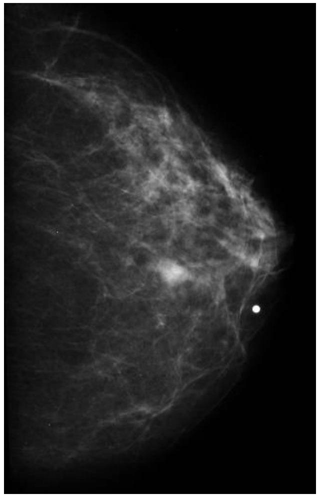


Candidate Rules:	R_1	R_2	R_3	R_4	R_5	…	R_n
Δ Model's score:	0.02	0.05	-0.01	0.01	0.03	…	-0.01

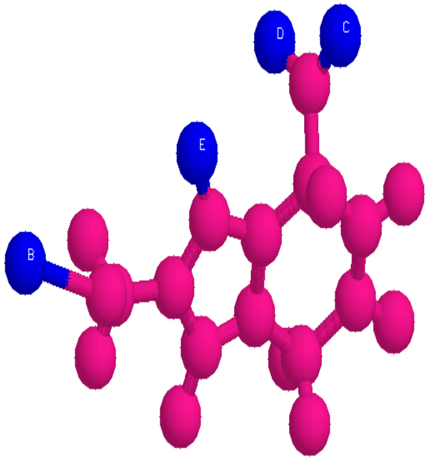
Iteratively add rules until stop criteria is met

Tasks Addressed

[Davis et al., IJCAI'07, ICML'07]



- **Given:** A radiologist's structured mammography report
- **Predict:** Abnormality is malignant



- **Given:** A set of 3D conformations of a small molecule
- **Predict:** Molecule's binding affinity to a target protein

Challenge: Hidden Structure

Drug

PID	Date	Medication	Dose
P1	5/1/02	Warfarin	10mg
P1	2/2/03	Terconazole	10mg

Diseases

PID	Date	Diag.
P1	2/1/01	Flu
P1	5/2/03	Bleeding

Observation

PID	Date	Weight
P2	2/2/03	120

Data and hence discovered patterns mention specific drugs or diseases

$$\boxed{\mathrm{Drug}(p, \mathrm{Terconazole})} \wedge \mathrm{Wt}(p, w) \wedge w < 120 \Rightarrow \mathrm{ADR}(p)$$

Regularities may involve drug or disease classes:
Enzyme inducers increase risk of internal bleeding

Solution: Clustering of Objects

$$\mathrm{Drug}(p, \mathrm{Terconazole}) \wedge \mathrm{Wt}(p, w) \wedge w < 120 \Rightarrow \mathrm{ADR}(p)$$

During learning, invent a clustering of objects that can appear in rules

$$\mathrm{Cluster2}(x) \wedge \mathrm{Drug}(p, x) \wedge \ldots \wedge \ldots \Rightarrow \mathrm{ADR}(p)$$

$$\mathrm{Cluster2}(x) = \{\mathrm{Terconazole}, \ldots, \mathrm{Ketoconazole}\}$$

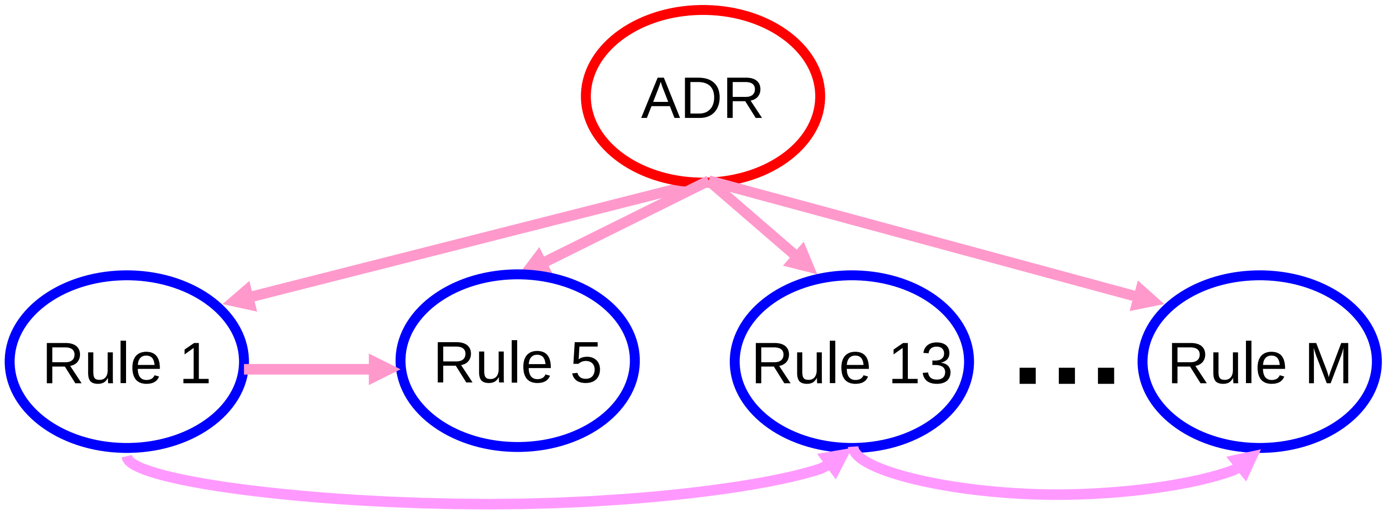
A group of “similar” objects

Motivation for Approach

- Why not use existing hierarchies?
- Why not cluster objects before learning?
- Inventing clusters during learning allows them:
  - To be tailored to specific prediction task
  - To exploit the context of the rule and the model

LUCID: Algorithmic Overview

[Davis et al., ICML'12]



Candidate Rules:	R_1	R_2	...	R_n	$R_{1,c}$	...	$R_{m,c}$
Δ Model's score:	0.02	0.05	...	-0.01	0.03	...	-0.01

Incorporating a Cluster in a Rule

If a candidate rule improves model's score **then**

$$\mathrm{Drug}(\mathrm{p}, \mathrm{Terconazole}) \wedge \mathrm{Wt}(\mathrm{p}, \mathrm{w}) \wedge \mathrm{w} < 120 \Rightarrow \mathrm{ADR}(\mathrm{p})$$

$$\mathrm{Cluster2}(\mathrm{d}) \wedge \mathrm{Drug}(\mathrm{p}, \mathrm{d}) \wedge \mathrm{Wt}(\mathrm{p}, \mathrm{w}) \wedge \mathrm{w} < 120 \Rightarrow \mathrm{ADR}(\mathrm{p})$$

1) Conjoin the invented predicate to the rule
2) Replace the object with a variable

Learning the Cluster Definition

- Which objects should be grouped together?
  - All constant of same type?
  - Slow because thousands of diagnosis and drugs
- Intuitively: Focus on similar constants, e.g., given Terconazole:
  - Which drugs can replace Terconazole?
  - Which drugs complement Terconazole?
- Idea: Use constants in “near miss” examples

Finding Relevant Objects: Near Miss Examples

$$\mathrm{Wt}(\mathrm{p},\mathrm{w}) \wedge \mathrm{w} < 120 \Rightarrow \mathrm{ADR}(\mathrm{p})$$

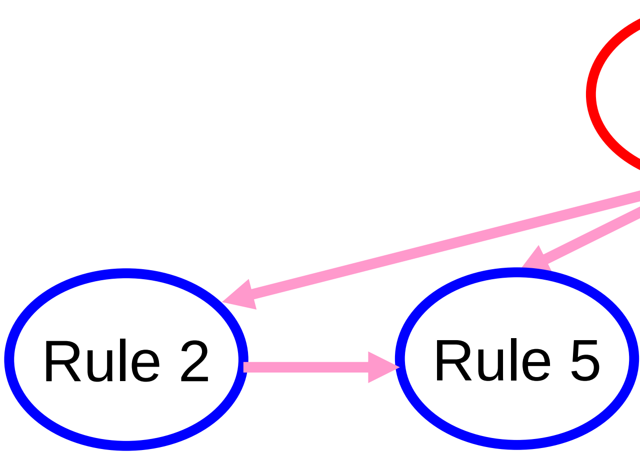
1. Find patients that
   I. Satisfy the more general rule
   II. Do not satisfy the more specific rule
2. Only consider drugs in this example set

$$\boxed{\mathrm{Drug}(\mathrm{p},\mathrm{Terconazole}) \wedge} \boxed{\mathrm{Wt}(\mathrm{p},\mathrm{w}) \wedge \mathrm{w} < 120} \Rightarrow \mathrm{ADR}(\mathrm{p})$$

Restricts which patients the rule applies to

Intuition: Context where Terconazole is prescribed

Evaluating Clusterings



Cluster Definition:

Cluster2(Terconazole)

Cluster2(Rifampicin)

Cluster2(Ketoconazole)

New Rule 5: $\text{Cluster2}(d) \wedge \text{Drug}(p, d) \wedge \ldots \Rightarrow \text{ADR}(p)$

Candidates:	Rifampicin	Ketocanazole	...	Alpranolol
Δ score:	0.04	0.02	...	-0.01

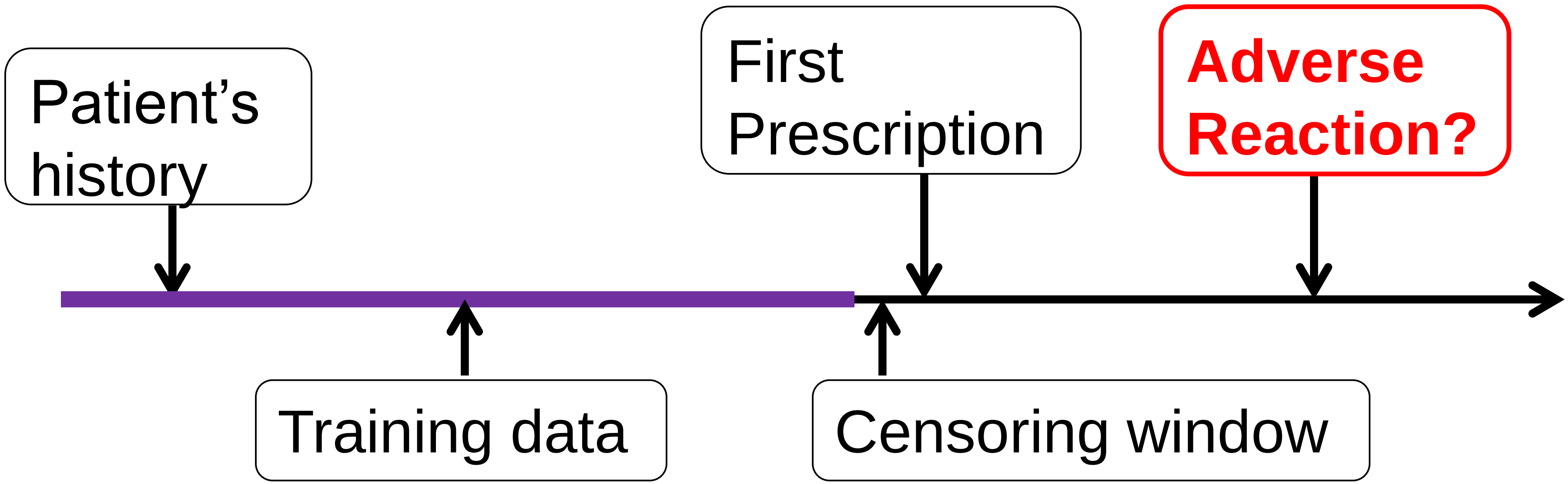
Add objects until none improves the score

Tasks and Data

- Tasks considered:
  - Myocardial infarction on selective Cox-2 inhibitors
  - Internal bleeding with Warfarin
  - Angioedema with ACE inhibitors

- Data from Marshfield Clinics
  - Diagnoses
  - Medications
  - Lab tests
  - Observations

Data Preparation

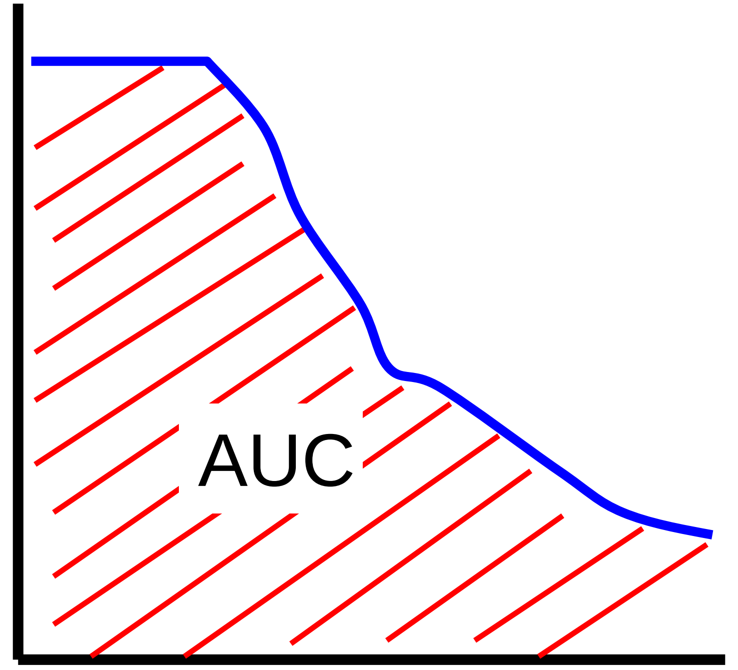


Positives: Adverse event after prescription

Negatives: Took medicine and no adverse event, matched on age and gender to positives

Evaluation Metric

$$\text{Precision} = \frac{TP}{TP + FP}$$



$$\text{Recall} = \frac{TP}{TP + FN}$$

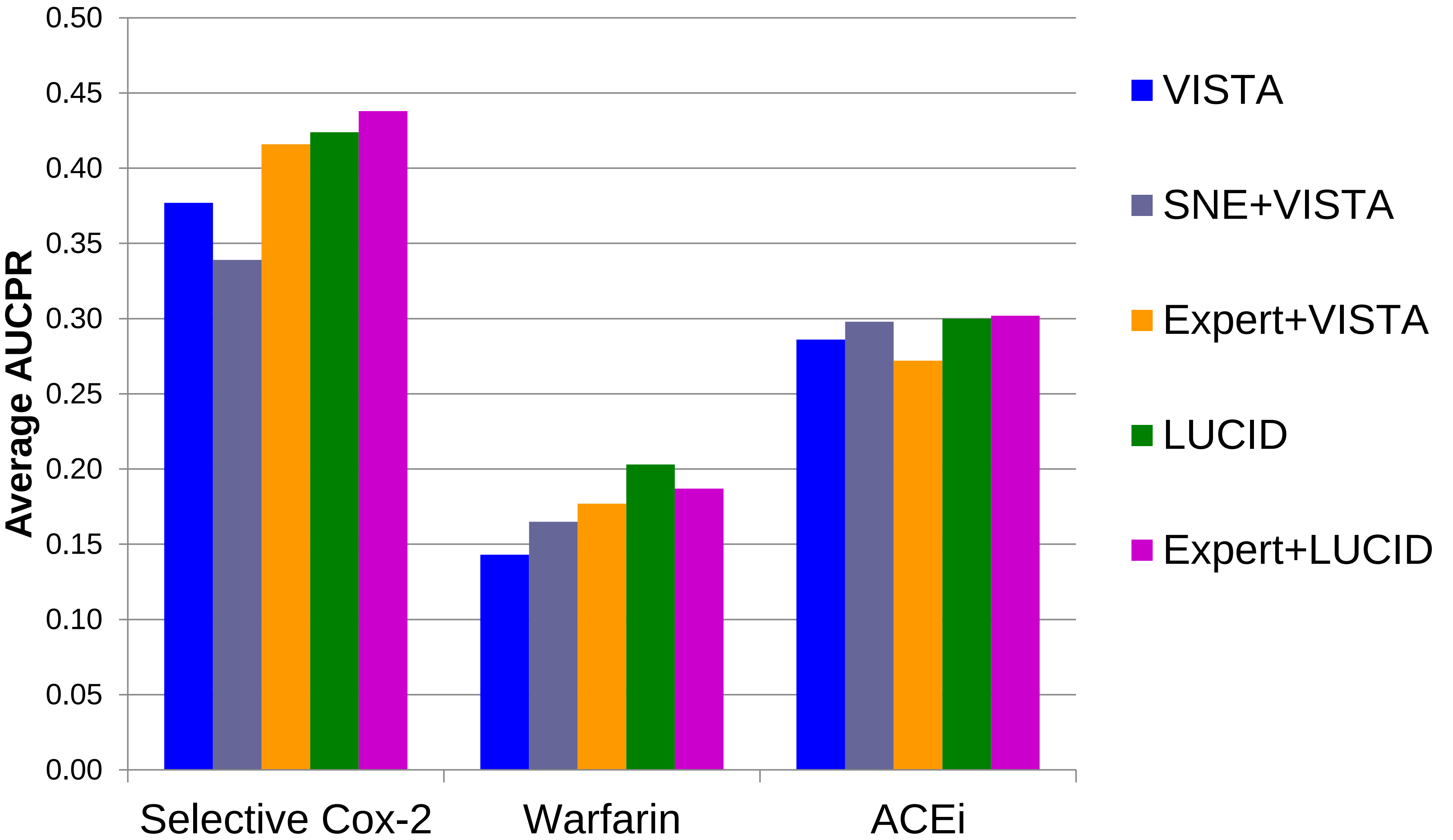
10 fold cross validated area under precision-recall curve

Systems Compared

Learned rules can contain

	Hand-Crafted Expert Hierarchy	Precluster	Dynamically Invented Clusters
VISTA			
SNE+VISTA		✓	
Expert+VISTA	✓		
LUCID			✓
Expert+Lucid	✓		✓

Results



Average AUCPR
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Selective Cox-2
Warfarin
ACEi
VISTA
SNE+VISTA
Expert+VISTA
LUCID
Expert+LUCID

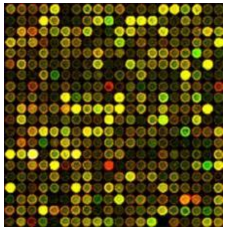
Outline

- Learning while accounting for model use
- Learning the structure of propositional probabilistic graphical models
- Learning the structure of probabilistic relational models
- Deep transfer: Transferring across entirely different domains

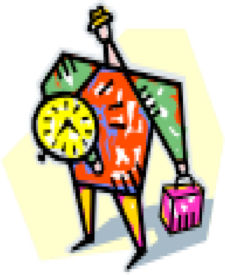
Challenge:
Acquiring Data Can Be Expensive

Costs include:

Emotional

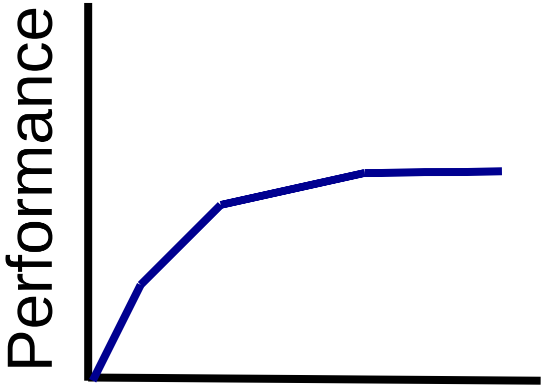


Money



Time

But more data is better…



Inductive Learning Cycle

- Get a task:

~~Learn first task~~

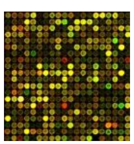
Forget what we learned!

Learn second task

- Collect data

Do this again

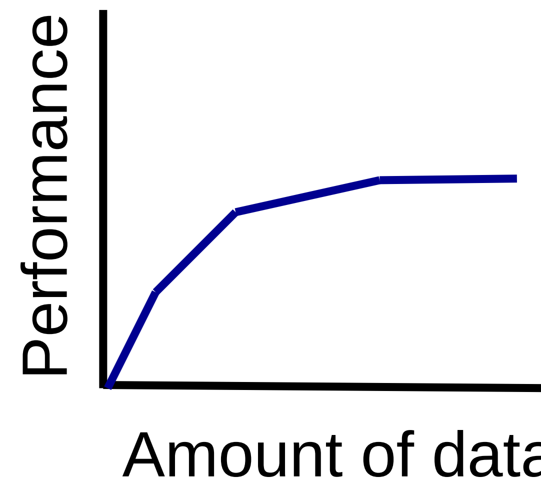
Emotional



Money

Time

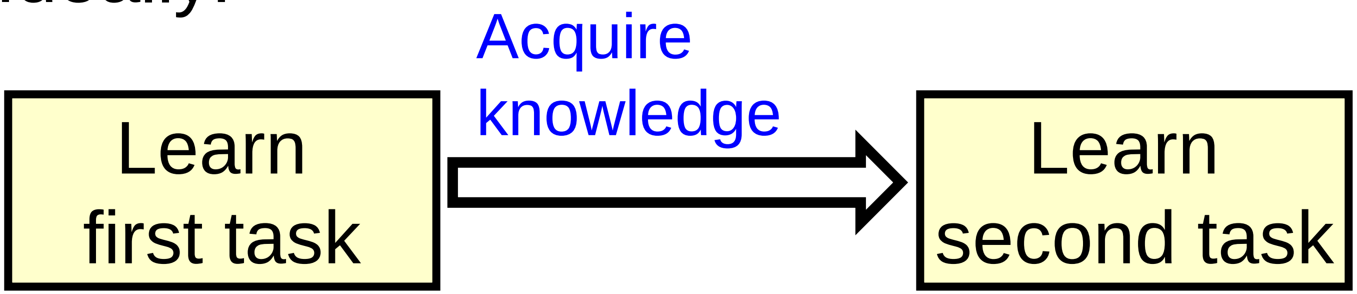
But more data is better…



Problem: One Off Solutions

- Interested in modeling many different domains

- Ideally:



- **Problem:** New domain looks “different”

Solution: Inductive transfer

Transfer Learning

Inductive Learning

- Given: Target Data
- Learn: Model

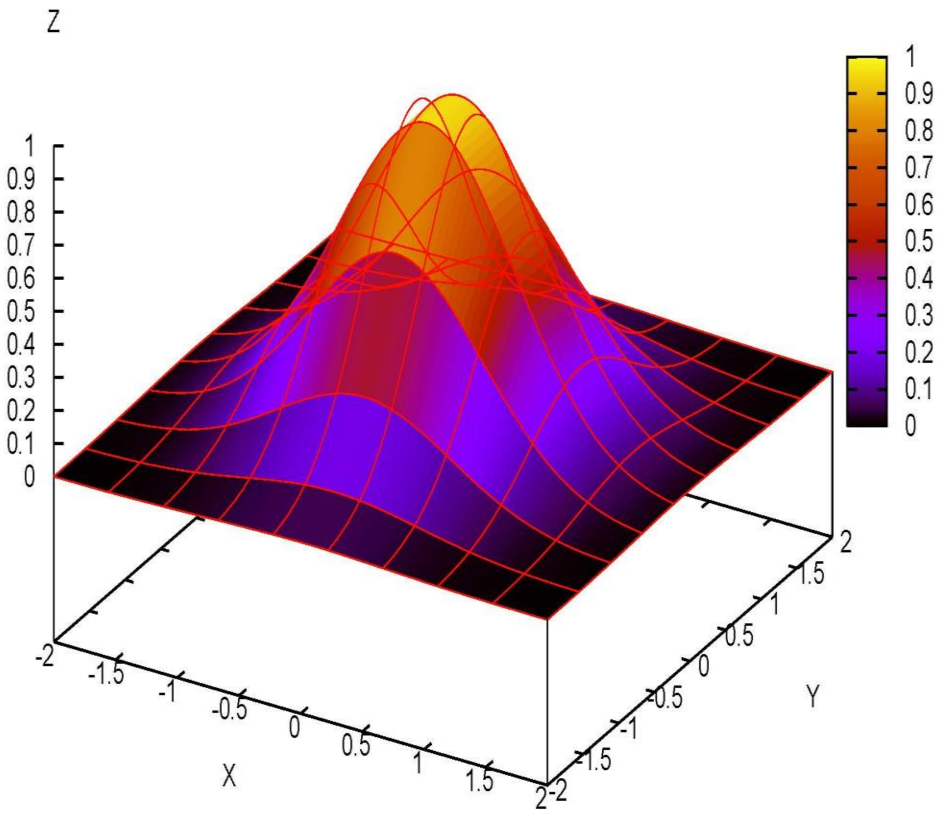
Transfer Learning

- Given: Target Data, Source Data
- Learn: Model

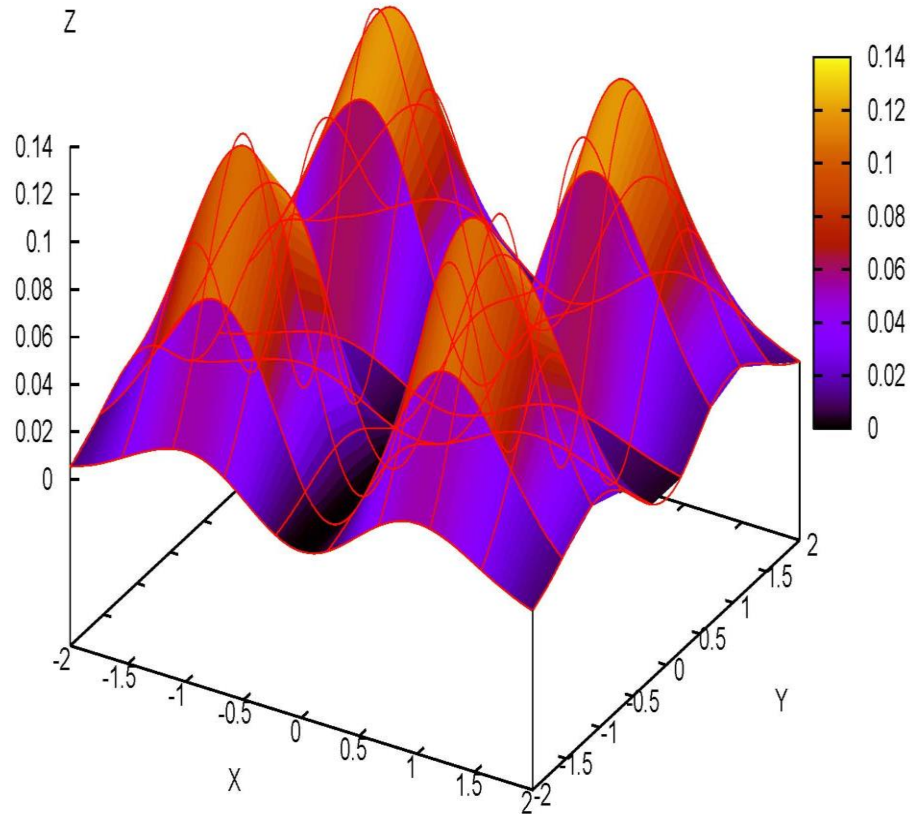
- Transfer: Makes use of data (or model or knowledge or …) from auxiliary domain
- Broadly speaking two types of transfer
  - Shallow: Same variables, different distributions
  - Deep: Different predicates, entities, properties

Same Variables, Different Distribution

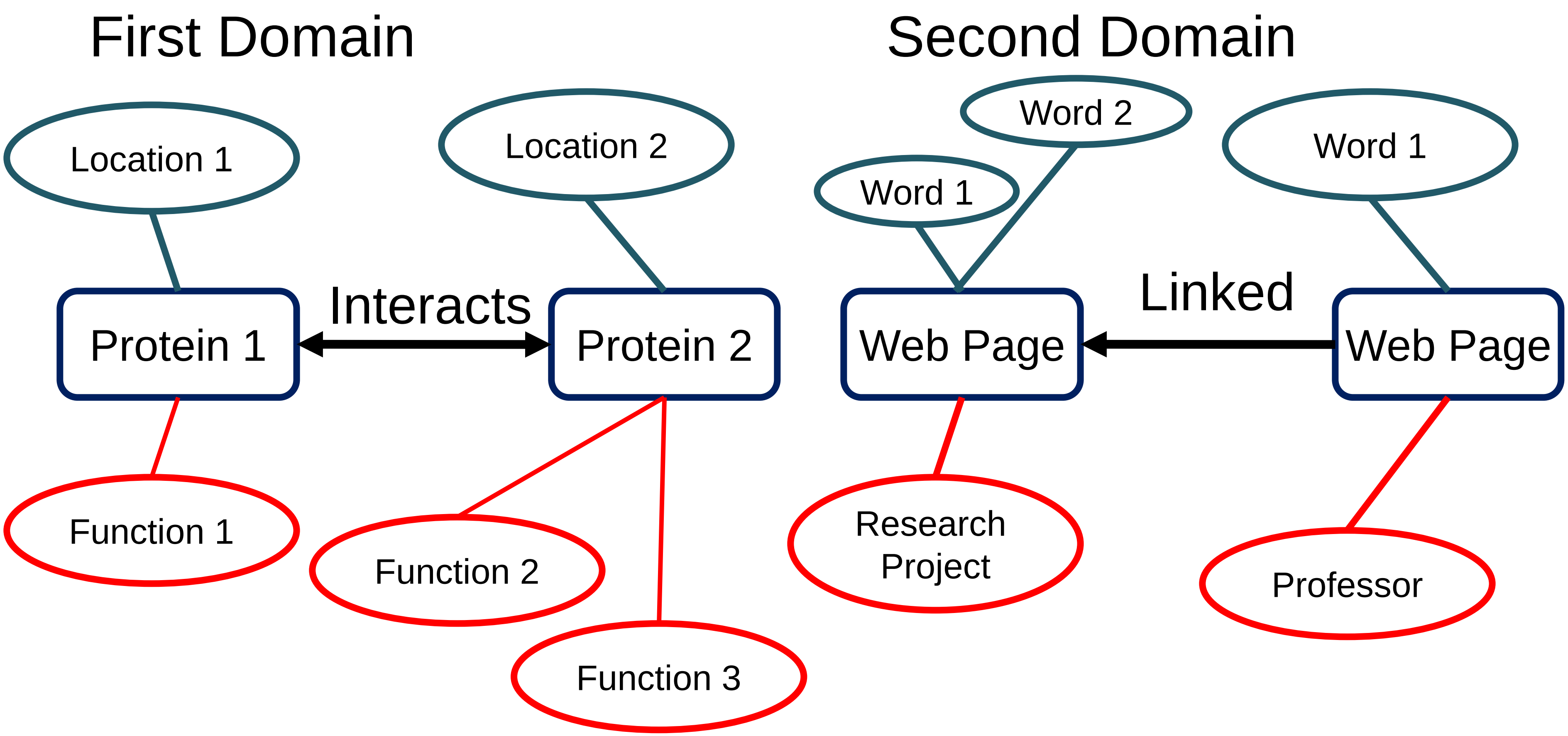
First Task



Second Task



Entirely Different Domains



Terminology

- Constants, variables, predicates, functions
  E.g.: Anna, x, Friends(x,y), MotherOf(x)
- **Grounding:** Replace all variables by constants
  E.g.: Friends(Anna,Bob)
- **Clause:**
  E.g.: Friends(x,y) ∨ Friends(y,z) ∨ Friends(x,z)
- **Predicate variable:** Variable instead of predicate name

$$r(x,y) \wedge s(x,z) \Rightarrow r(z,y)$$

$$\Big\downarrow\ r \rightarrow \text{Location},\ s \rightarrow \text{Interacts}$$

$$\text{Location}(x,y) \wedge \text{Interacts}(x,z) \Rightarrow \text{Location}(z,y)$$

Markov Logic Networks (MLNs)

[Richardson & Domingos, MLJ'06]

- A logical knowledge base is a set of hard constraints on the set of possible worlds

- Let us make them soft constraints
  - Give each formula a weight
  - Worlds that violate a formula become less probable

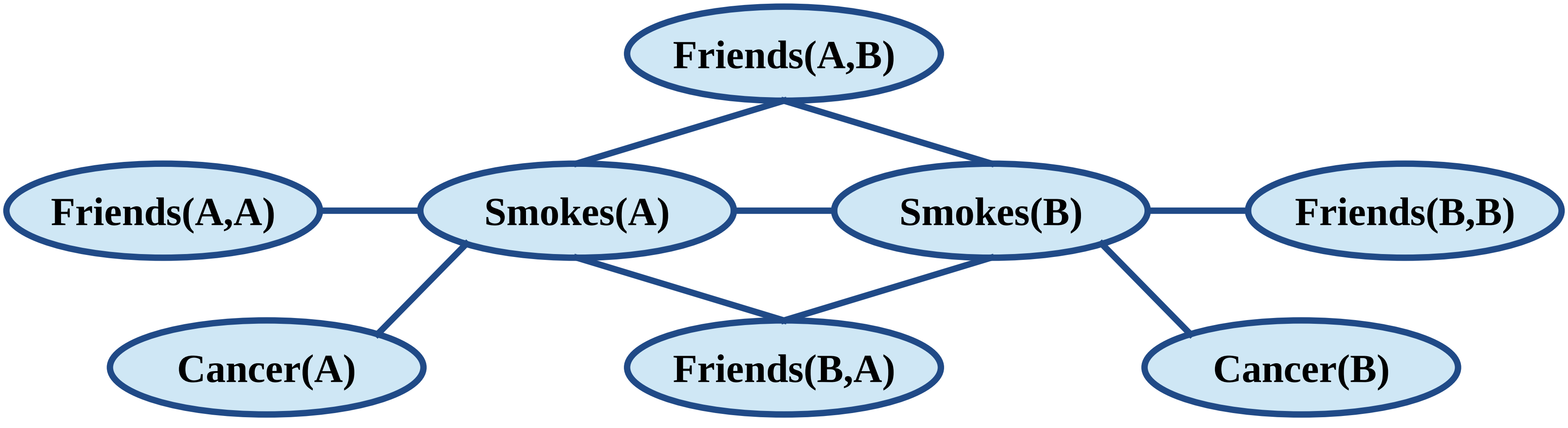
1.5	$\mathrm{Location}(x,y) \wedge \mathrm{Interacts}(x,z) \Rightarrow \mathrm{Location}(z,y)$

$$\mathrm{P(world)} \propto \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

MLN to Markov Network

$\forall X\ Smokes(X) \Rightarrow Cancer(X)$
$\forall X,Y\ Friends(X,Y) \Rightarrow [Smokes(X) \Leftrightarrow Smokes(Y)]$

Constants: Anna (A), Bob (B)



Markov Logic Networks: Learning

0.15 Smoke(x) $\Rightarrow$ Cancer(x)

Weight of Feature *i*

Formula *i*

$$P(x) = \frac{1}{Z} \exp\left(\sum_i \boxed{w_i} \boxed{f_i(x)} \right)$$

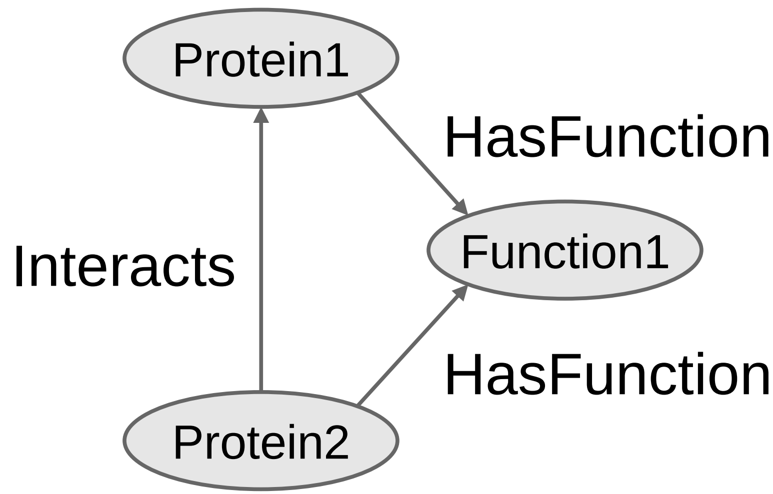
Structure Learning

- Given: Target Data
- Learn: Formulas, Weights

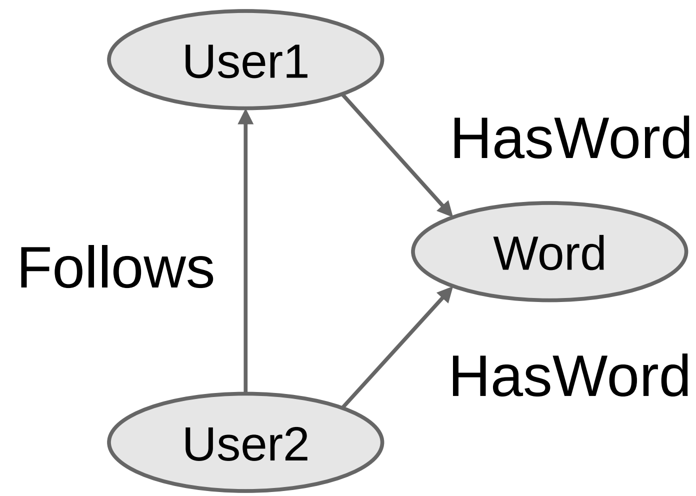
Search through spaces of clauses

Convex optimization of pseudolikelihood

Challenge: Domains Described by Different Predicates, Objects, Etc.



Protein-Protein Interaction

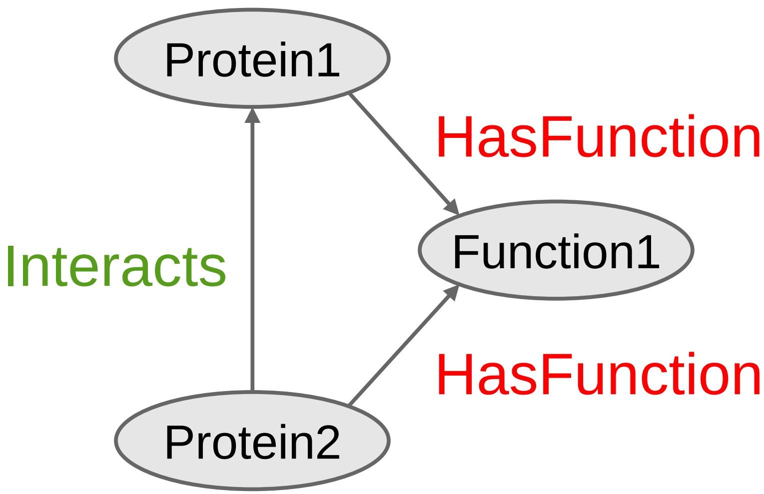


Twitter

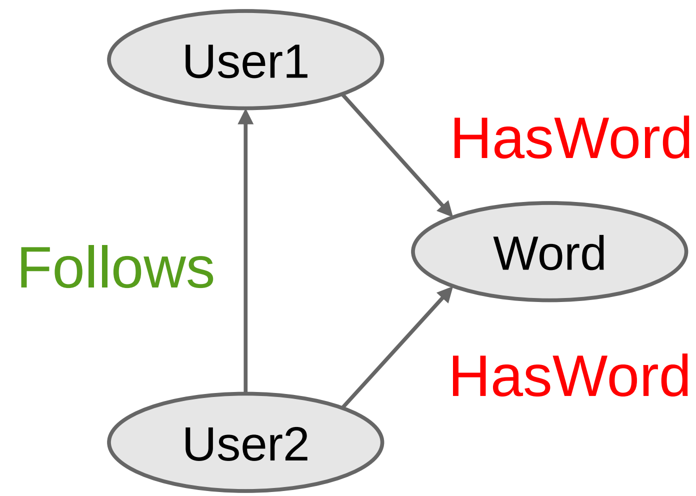
Challenge: Domains Described by Different Predicates, Objects, Etc.

$$r(x, z) \land s(x, y) \Rightarrow r(y, z)$$

Common templates used to model domains
Use variables and not predicate names



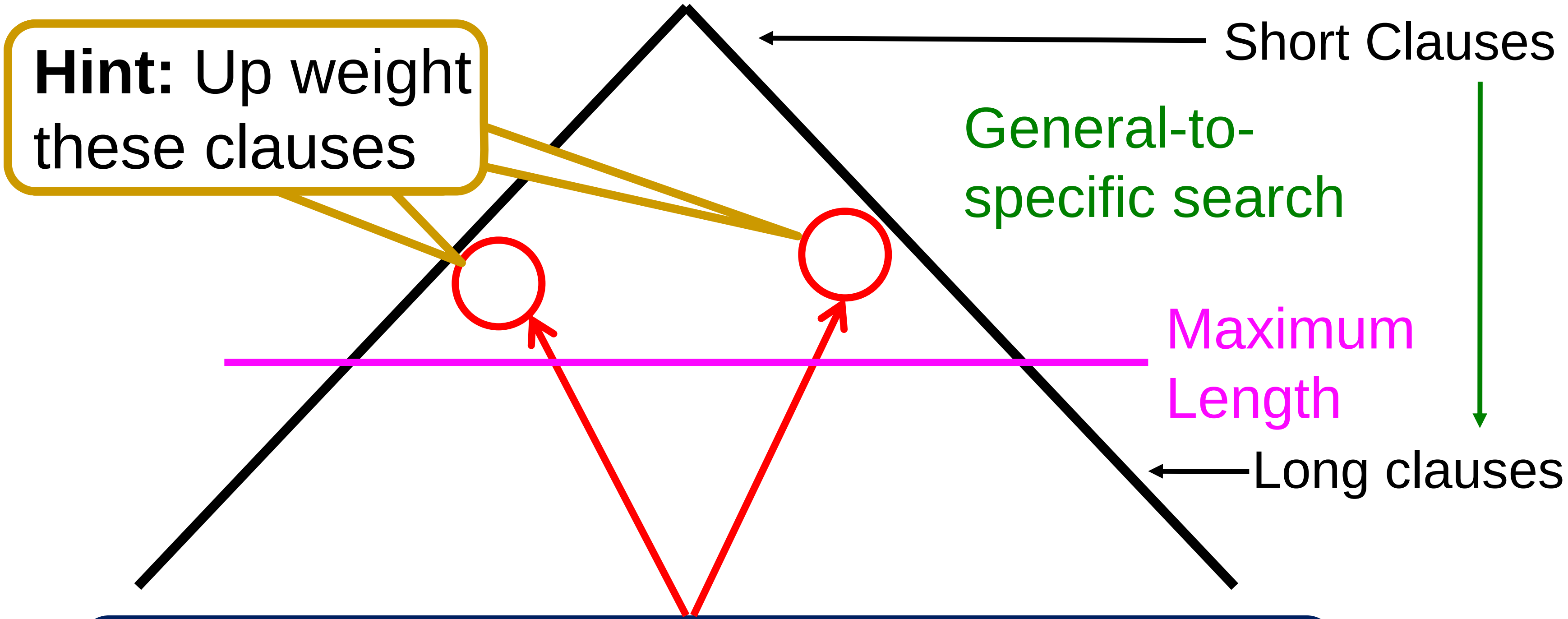
Protein-Protein Interaction



Twitter

Transfer as Declarative Bias

Search through (Large) Space of Possible Clauses



Intuition: Bias learning toward models that contain previously useful clauses

Overview of TODTLER

[Van Haaren, Kolobov, & Davis, AAAI'15]

$r(x, z) \wedge s(x, y) \Rightarrow r(y, z)$

$r(x, y) \Rightarrow r(y, x)$

....

$Word(a, w) \wedge Follows(a, b) \Rightarrow Word(b, w)$

$Follows(a, b) \Rightarrow Follows(b, a)$

...

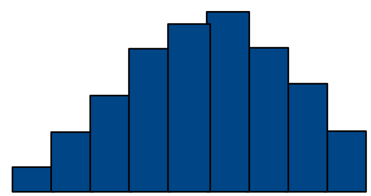
Follows(Jan, Jesse)

Follows(Jesse, Jan)

Word(Jesse, Basketball)

...

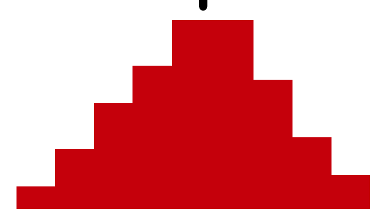
Learn distribution over 2nd-order clause templates in source and transfer it to target



Source domain distribution over **templates**

Target domain distribution over **formulas**

Combine



Adapted target domain distribution

Learning the Posterior

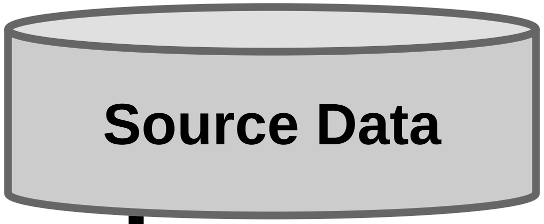
- Probabilistic inference for a posterior over 2nd-order clauses is hopelessly intractable

- Hence will use a heuristic approach
  - Generate second-order templates
  - For each template create all its first-order groundings
  - Treat each first-order clause independently and score its “usefulness” based on pseudolikelihood
  - Template score: Aggregation over its first-order groundings

Constructing Second-Order Clause Templates

- Generate all second-order clause templates
  - Maximum number of predicate variables
  - Maximum number of object variables
  - Maximum length
- Generate first-order clauses by grounding out predicate variables with predicate names
- Do this in source and target domain

Using the Source Data



Score first-order clauses

Word(a, w) ∧ Follows(a, b) ⇒ Word(b, w)
Type(a, t) ∧ Follows(a, b) ⇒ Type(b, t)
…
Follows(a, b) ⇒ Follows(b, a)

Improvement in PLL obtained by adding clause to empty MLN

Aggregate scores of template's first-order instantiations

Rescale PLLs between 0 and 1 and average

Ranking of Second-Order Templates

0.15 s(x, z) ∧ r(x, y) ⇒ s(y, z)

0.08 r(x, y) ⇒ r(y, z)

Learning in Target Domain

Target Data

Score first-order clauses

0.23 Loc(p, l) ∧ Interacts(p, q) ⇒ Loc (q, l)
0.14 Interacts(p, q) ⇒ Interacts(q, p)
0.12 Func(p, f) ∧ Interacts(p, q) ⇒ Func (q, f)

Ranked Templates

0.13 s(x, z) ∧ r(x, y) ⇒ s(y, z)
0.08 r(x, y) ⇒ r(y, z)

Combine Scores

0.30 Loc(p, l) ∧ Interacts(p, q) ⇒ Loc (q, l)
0.16 Func(p, f) ∧ Interacts(p, q) ⇒ Func (q, f)
0.15 Interacts(p, q) ⇒ Interacts(q, p)

Walk down list
Pick clauses

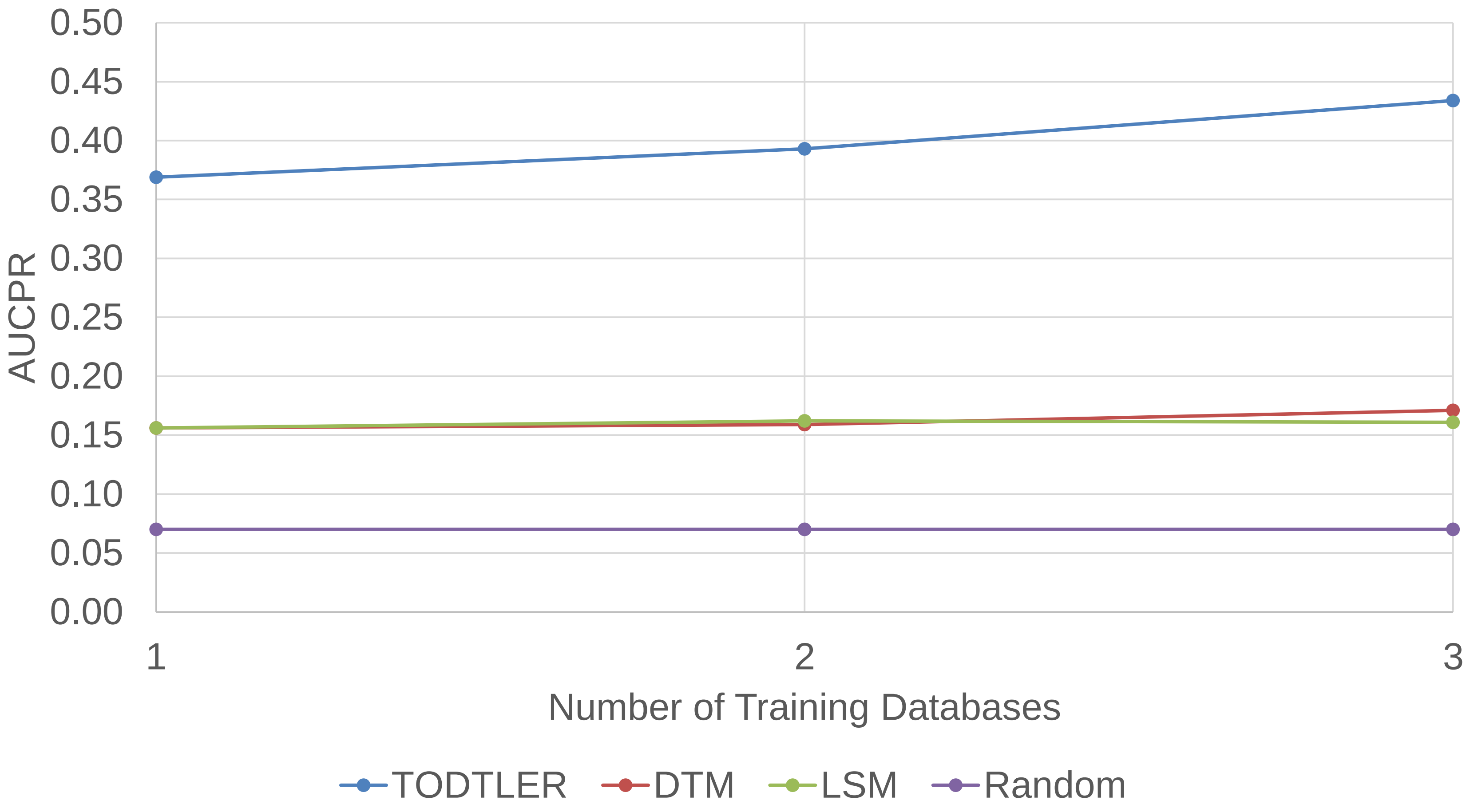
Empirical Evaluation

- Can we successully transfer among different domains?
- Will transfer outperform learning from scratch?
- Which transfer approach is the best?
- Will we discover and transfer relevant templates?

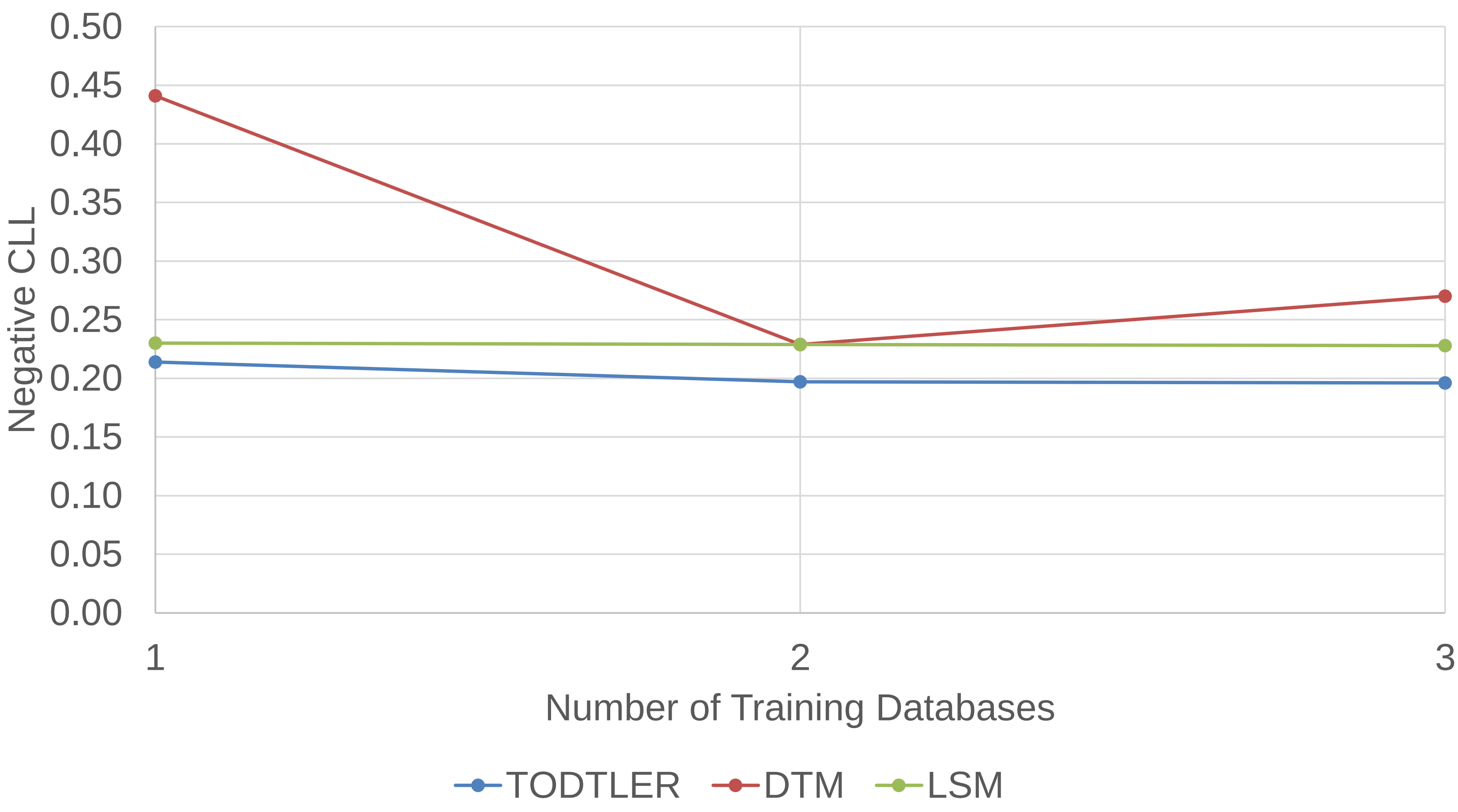
Data and Methodology

- Transfer among three domains:
  - Yeast protein: 7 predicates, 1.4M ground atoms
    [Davis et al., ECML'05]
  - WebKB: 3 predicates, 4.4M ground atoms
    [Craven & Slattery, MLJ'01]
  - Twitter: 3 predicates, 50K ground atoms
- Evaluation metrics
  - Area under the precision recall curve (AUC PR)
  - Negative conditional log likelihood (CLL)
  - Run time

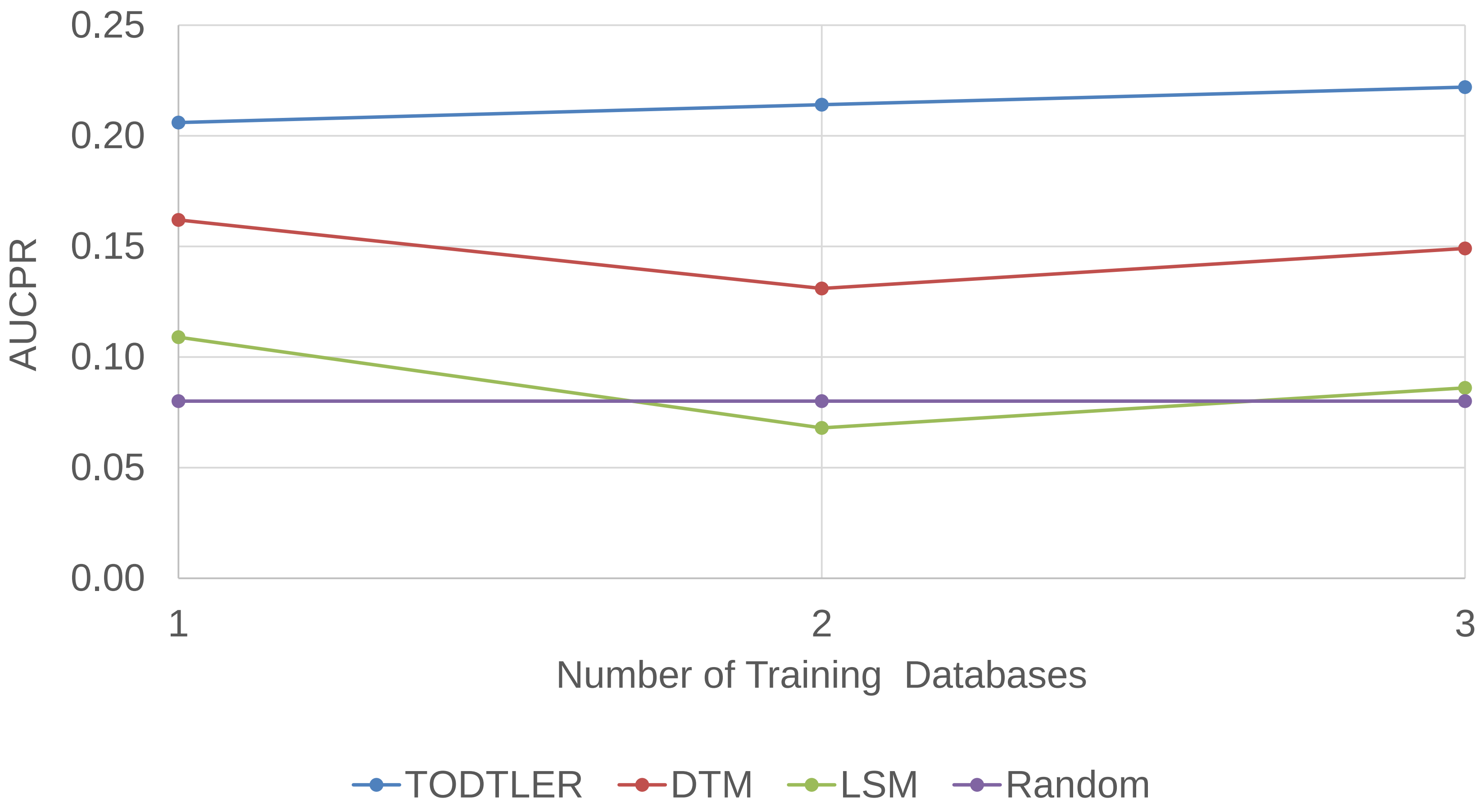
Twitter to WebKB



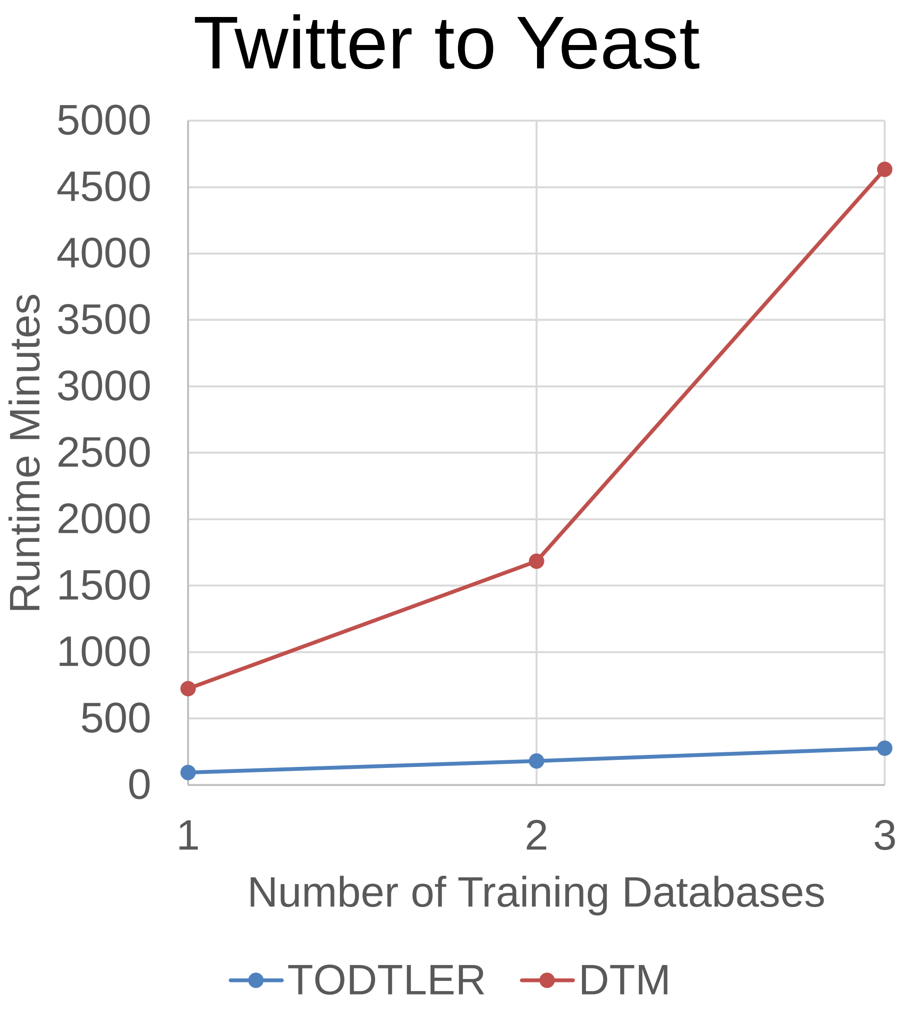
Twitter to WebKB

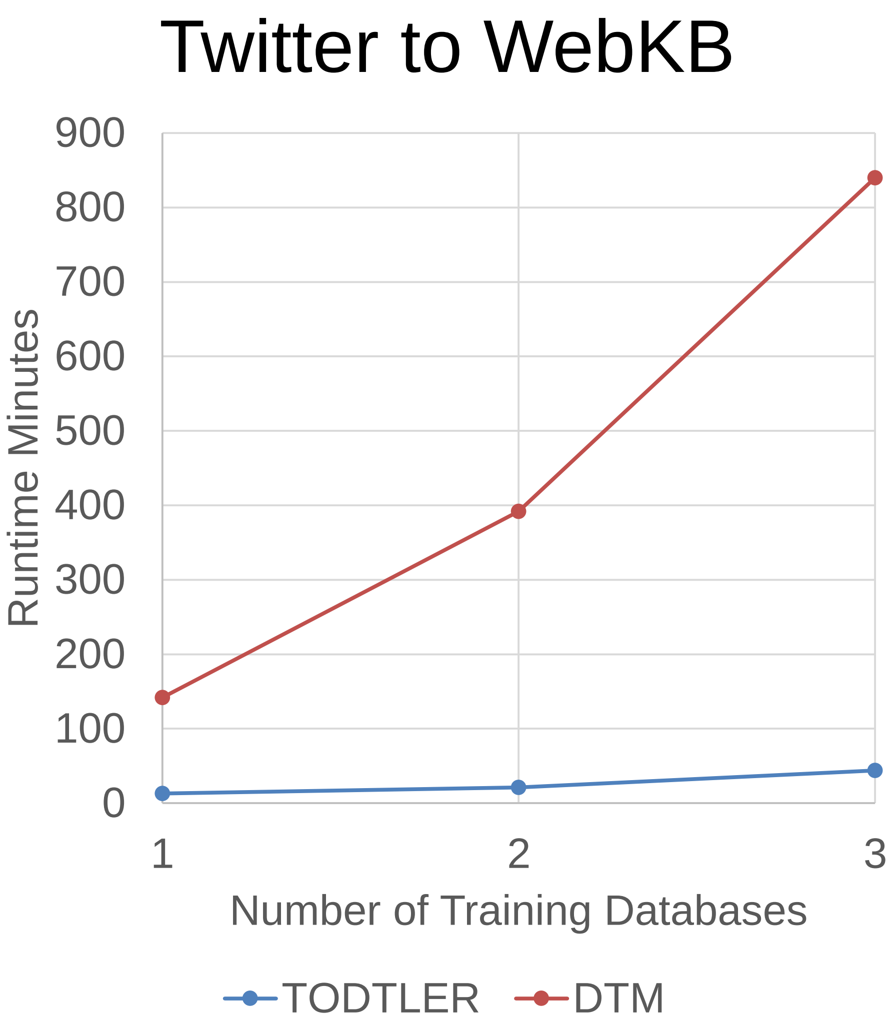


WebKB to Yeast



Run Time


Twitter to Yeast
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
Runtime Minutes
1
2
3
Number of Training Databases
TODTLER
DTM


Twitter to WebKB
900
800
700
600
500
400
300
200
100
0
Runtime Minutes
1
2
3
Number of Training Databases
TODTLER
DTM

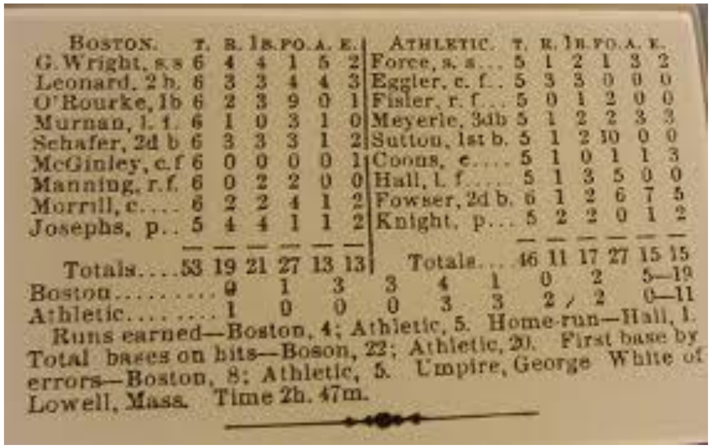
Templates Ranked in Top 10

	Yeast	WebKB	Twitter
Symmetry: $r(x, y) \Rightarrow r(y, x)$	1^{st}	1^{st}	2^{nd}
Homophily $s(x, y) \wedge r(z, y) \Rightarrow s(z, y)$	3^{rd}	8^{th}	6^{th}
Transitivity $r(x, y) \wedge r(y, z) \Rightarrow r(x, y)$	6^{th}	2^{nd}	-
Triangle Completion $r(x, z) \wedge r(y, z) \Rightarrow r(x, y)$	10^{th}	-	5^{th}
Cycle $r(x, y) \wedge r(y, z) \Rightarrow r(z, x)$	-	4^{th}	-

Part II: Applications to Sports

Traditional Sports Data: Box Scores

Box Score from 1876



Boston.	T.	R.	1B.	PO.	A.	E.
G. Wright, s. s	6	4	4	1	5	2
Leonard, 2 b.	6	3	3	4	4	3
O'Rourke, 1b	6	2	3	9	0	1
Murnan, l. f.	6	1	0	3	1	0
Schafer, 2d b	6	3	3	3	1	2
McGinley, c. f	6	0	0	0	0	1
Manning, r. f.	6	0	2	2	0	0
Morrill, c.	6	2	2	4	1	2
Josephs, p.	5	4	4	1	1	2
Totals	53	19	21	27	13	13

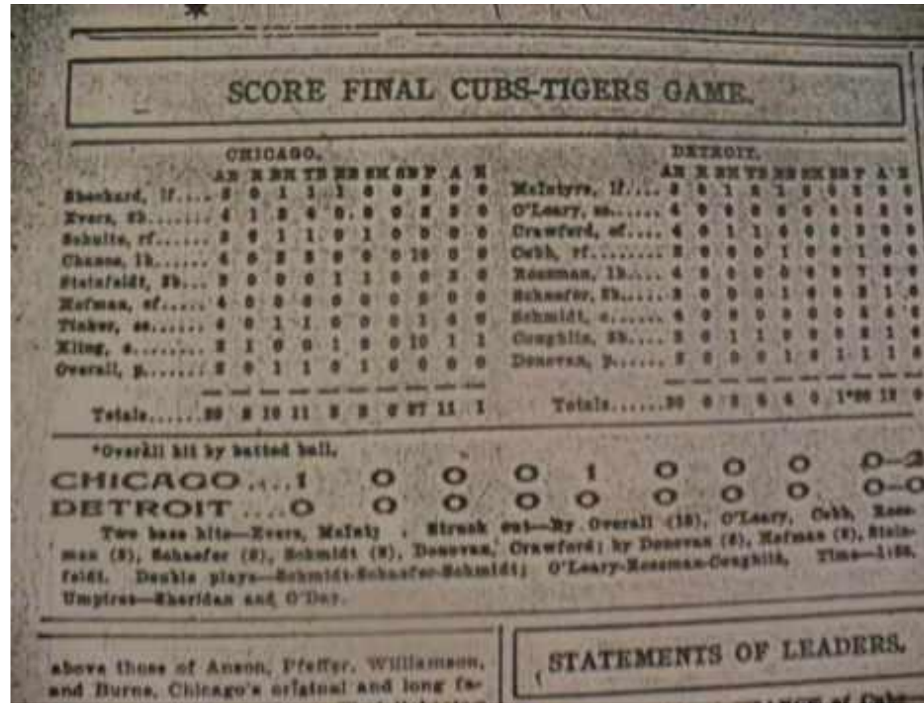
Athletic.	T.	R.	1B.	PO.	A.	E.
Force, s. s.	5	1	2	1	3	2
Eggler, c. f.	5	3	3	0	0	0
Fisler, r. f.	5	0	1	2	0	0
Meyerle, 3d b	5	1	2	2	3	3
Sutton, 1st b.	5	1	2	10	0	0
Coons, c.	5	1	0	1	1	3
Hall, l. f.	5	1	3	5	0	0
Fowser, 2d b.	6	1	2	6	7	5
Knight, p.	5	2	2	0	1	2
Totals	46	11	17	27	15	15

Boston	0	1	3	3	4	1	0	2	5—19
Athletic	1	0	0	0	3	3	2	2	0—11

Runs earned—Boston, 4; Athletic, 5. Home-run—Hall, 1. Total bases on hits—Boston, 22; Athletic, 20. First base by errors—Boston, 8; Athletic, 5. Umpire, George White of Lowell, Mass. Time 2h. 47m.

https://en.wikipedia.org/wiki/Box_score_(baseball)

Box Score from 1908



SCORE FINAL CUBS-TIGERS GAME.

CHICAGO.

	AB	R	BH	TB	BB	SH	SB	P	A	E
Sheckard, lf	5	0	1	1	1	0	0	3	0	0
Evers, 2b	4	1	3	4	0	0	0	2	3	0
Schulte, rf	3	0	1	1	0	1	0	0	0	0
Chance, 1b	4	0	2	2	0	0	0	10	0	0
Steinfeldt, 3b	3	0	0	0	1	1	0	0	2	0
Hofman, cf	4	0	0	0	0	0	0	0	0	0
Tinker, ss	4	0	1	1	0	0	0	1	4	0
Kling, c	3	1	0	0	1	0	0	10	1	1
Overall, p	3	0	1	1	0	1	0	0	0	0
Totals	30	2	10	11	3	3	0	27	11	1

DETROIT.

	AB	R	BH	TB	BB	SH	SB	P	A	E
McIntyre, lf	3	0	1	1	1	0	0	2	0	0
O'Leary, ss	4	0	0	0	0	0	0	0	3	0
Crawford, cf	4	0	1	1	0	0	0	3	0	0
Cobb, rf	3	0	0	0	1	0	0	1	0	0
Rossman, 1b	4	0	0	0	0	0	0	7	2	0
Schaefer, 2b	3	0	0	0	1	0	0	3	1	0
Schmidt, c	4	0	0	0	0	0	0	5	4	0
Coughlin, 3b	3	0	1	1	0	0	0	2	1	0
Donovan, p	3	0	0	0	1	0	1	1	1	0
Totals	30	0	3	3	4	0	1	*26	12	0

*Overall hit by batted ball.

CHICAGO 1	0	0	0	1	0	0	0	0	0—2
DETROIT 0	0	0	0	0	0	0	0	0	0—0

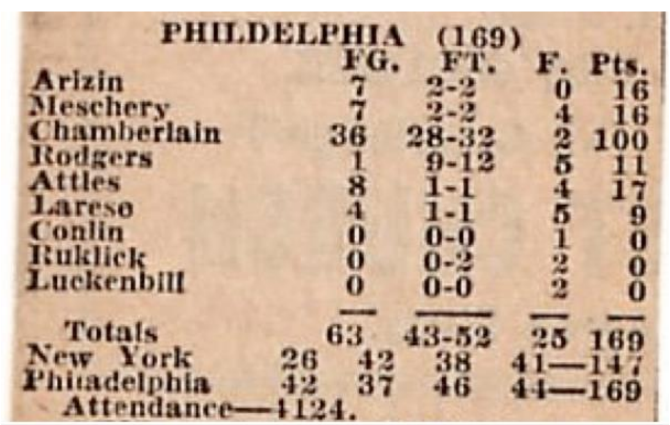
Two base hits—Evers, McIntyre. Struck out—By Overall (10), O'Leary, Cobb, Rossman (2), Schaefer (2), Schmidt (2), Donovan, Crawford; by Donovan (5), Hofman (2), Steinfeldt. Time—1:25. Double plays—Schmidt-Schaefer-Schmidt; O'Leary-Rossman-Coughlin. Umpires—Sheridan and O'Day.

above those of Anson, Pfeffer, Williamson, and Burns, Chicago's original and long fa-

STATEMENTS OF LEADERS.

https://miscbaseball.wordpress.com/2009/10/11/1908-the-cubs-win-the-world-series/

Box Score from 1962



PHILDELPHIA (169)

	FG.	FT.	F.	Pts.
Arizin	7	2-2	0	16
Meschery	7	2-2	4	16
Chamberlain	36	28-32	2	100
Rodgers	1	9-12	5	11
Attles	8	1-1	4	17
Lareso	4	1-1	5	9
Conlin	0	0-0	1	0
Ruklick	0	0-2	2	0
Luckenbill	0	0-0	2	0
Totals	63	43-52	25	169

New York	26	42	38	41—147
Philadelphia	42	37	46	44—169

Attendance—4124.

Sports Analytics

"Traditional" approach to evaluating players

- Scouts evaluate subjectively on gut

NAME	P	HT	WT	SCHOOL--CITY--STATE	COACH
NAT ARCHIBALD (2/4)	G	5-10	140	DE WITT CLINTON--BRONX, N.Y.	RICHARD BUCKNER

Speed......... 9 Off. Moves..... 6/7
Spring......... 9 Court Savvy.... 7
Shooting....... 6 "D" Potential.. 4
Dribbling...... 10 Aggressivness.. 7
Playmaking..... 7 Attitude....... 7
GRADES: rank ?/Gen/75..no Boards

SUMMATION: lightening-quick guard with Globie-dribbling talent lacks strength/size/defensive foundation for top majors--slim southpaw has terrific moves to basket & is blur on break but has limited shooting range & consistency (nice tough 15 feet & in when "on") & must learn moves for outside game--FINE FOR PRESSING/RUNNING LM OUTFIT--N

- Traditional statistics

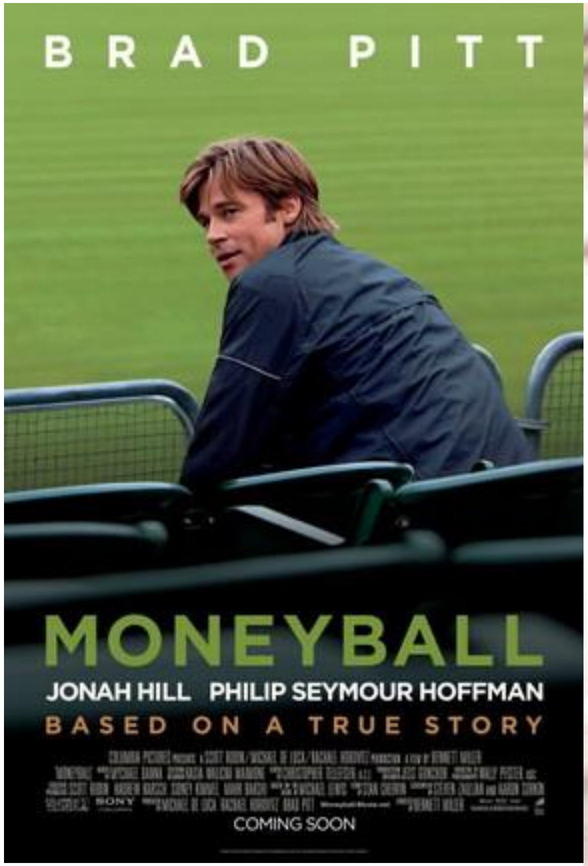
PHILDELPHIA (169)

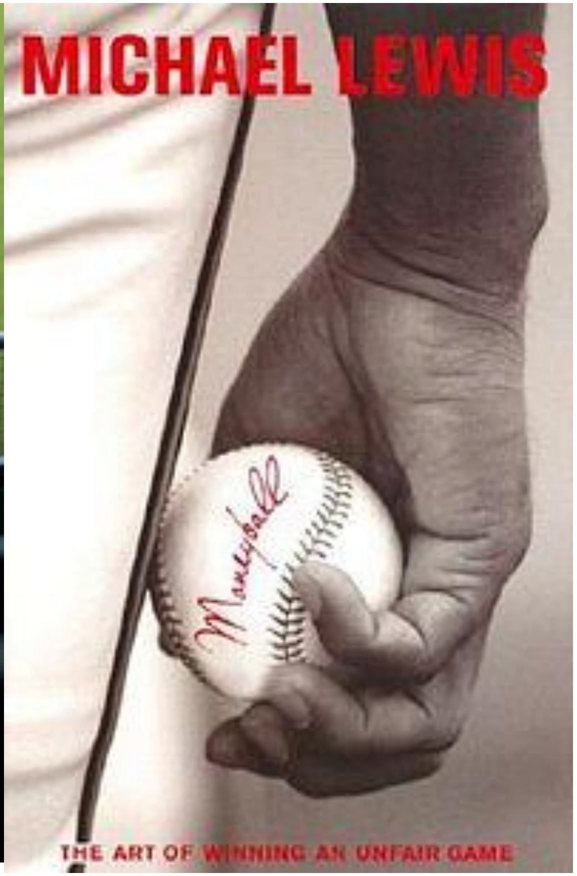
	FG.	FT.	F.	Pts.
Arizin	7	2-2	0	16
Meschery	7	2-2	4	16
Chamberlain	36	28-32	2	100
Rodgers	1	9-12	5	11
Attles	8	1-1	4	17
Larese	4	1-1	5	9
Conlin	0	0-0	1	0
Ruklick	0	0-2	2	0
Luckenbill	0	0-0	2	0
Totals	63	43-52	25	169

New York	26	42	38	41	—147
Philadelphia	42	37	46	44	—169

Attendance—4124.

Sabremetrics: A Better Idea





“Bill James…asked the question why” – Paul DePodesta, “Moneyball”

Why are common statistics meaningful?

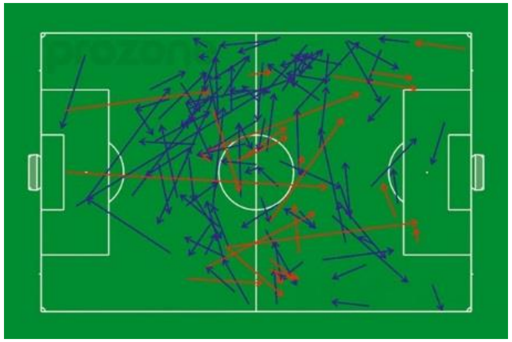
- Question 1: Which statistics best quantify various aspects of team or player performance?
- Question 2: Can we come up with a single statistic to rank players?
- Question 3: How can we project future team or player performance?

Assumption: Available data is box score like

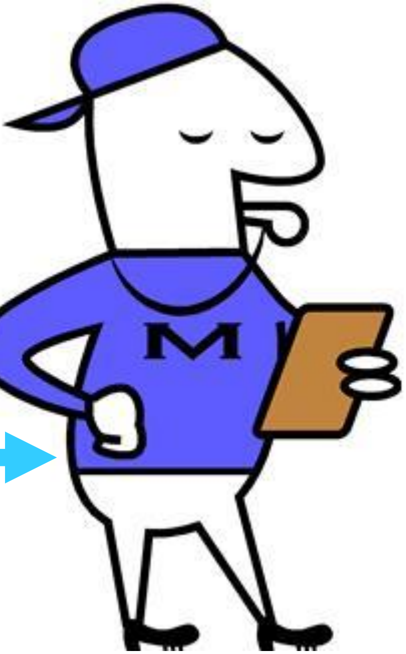
Sports Data Today

Train
Scout
Identify players
Etc.

Complex Data



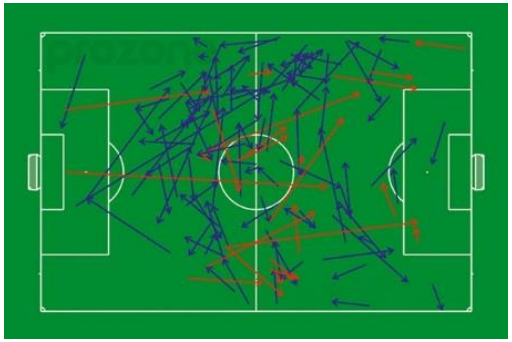
Automated data analysis techniques



How can we exploit all the collected data?

How Can Analytics Help?

Complex Data



Compute relevant metrics

Number of shots on target: 10

Learn predictive models

Heart Rate > 140 AND
Distance > 8KM $\Rightarrow$ Tired

Discover novel patterns

Player 1 pass to Player 2 AND
Player 2 dribbles …

Three New Types of Data

- Event stream: Events with time and location

…	**Pass** (60,10)	**Out** (75,0)	**Throw** (75,0)	**Run** (80,5)	**Pass** (86,15)	**Cross** (90,20)	**Shot** (88,43)	…

- Athlete monitoring:
  GPS, accelerometer, etc.

- Optical tracking:
  X, Y locations of players

Outline

- **Rating players:** Assign a rating to each action a player performs in a match

- **Understand strategy:** Discover patterns from player tracking data

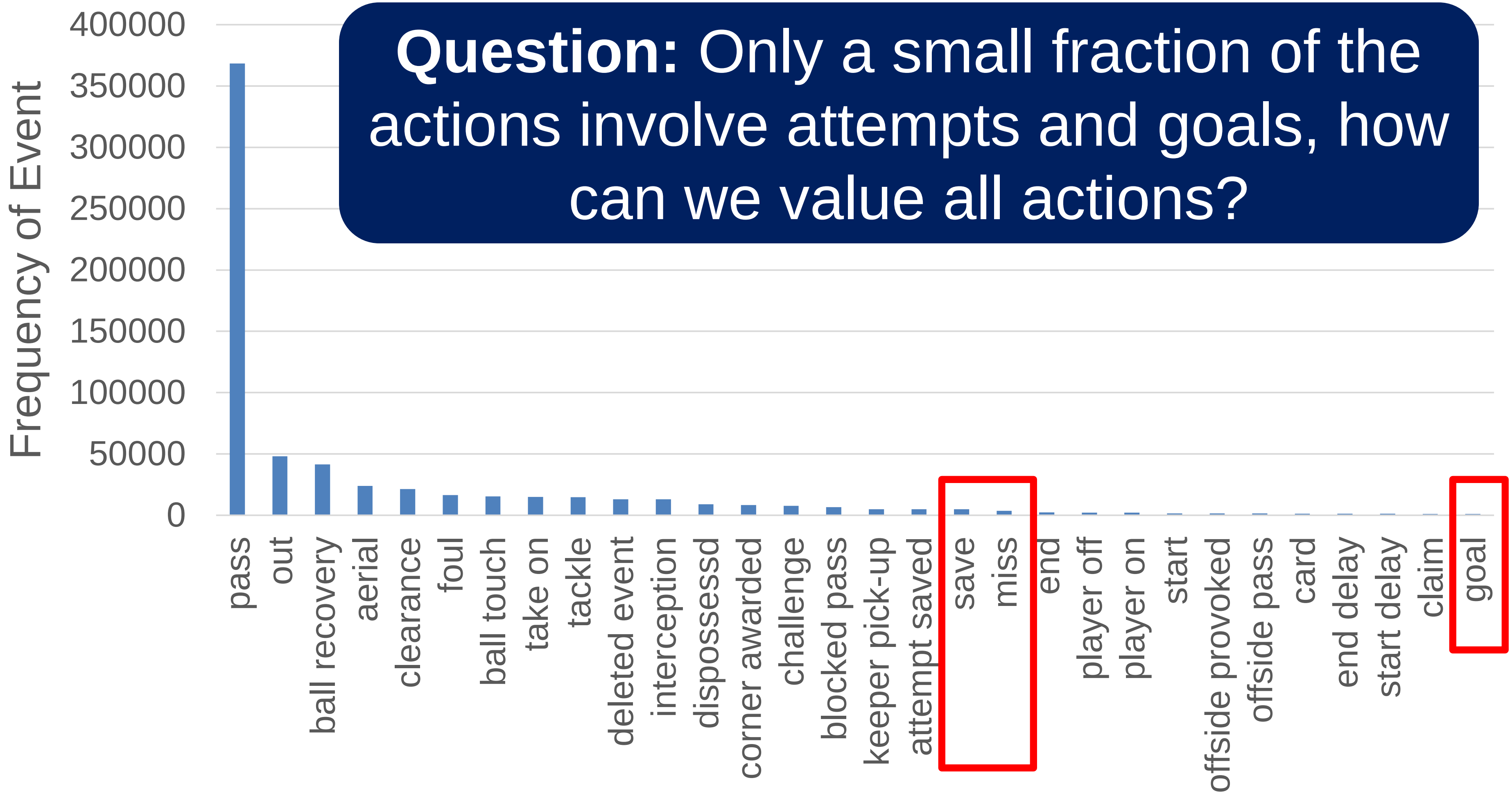
Outline

- **Rating players:** Assign a rating to each action a player performs in a match

- **Understand strategy:** Discover patterns from player tracking data

Distribution of Some Events

Question: Only a small fraction of the actions involve attempts and goals, how can we value all actions?

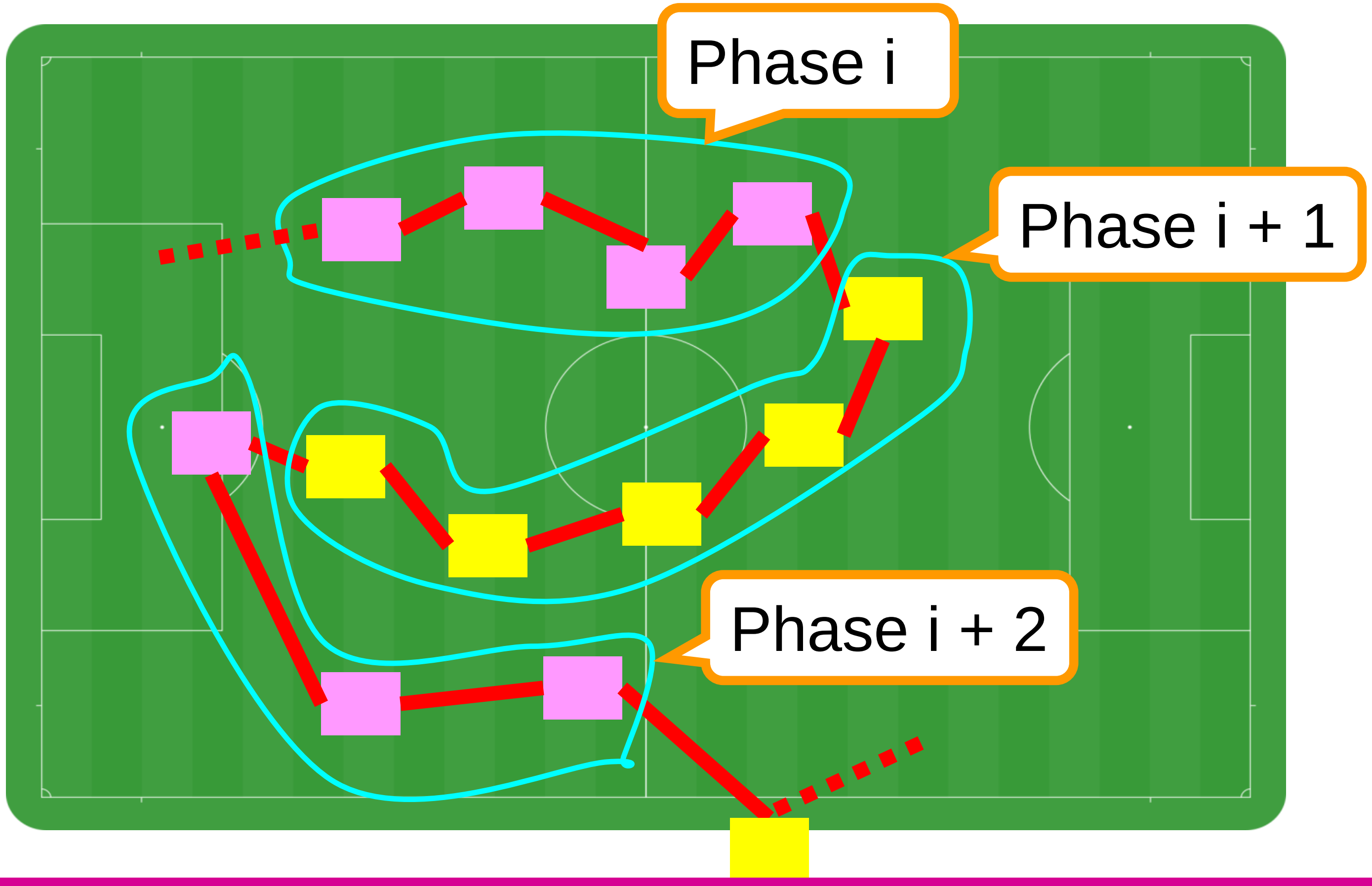


Our Approach: STARSS

- Given: Event stream with type and location of all events (e.g., passes and shots)
- Do: Assign rating to each action

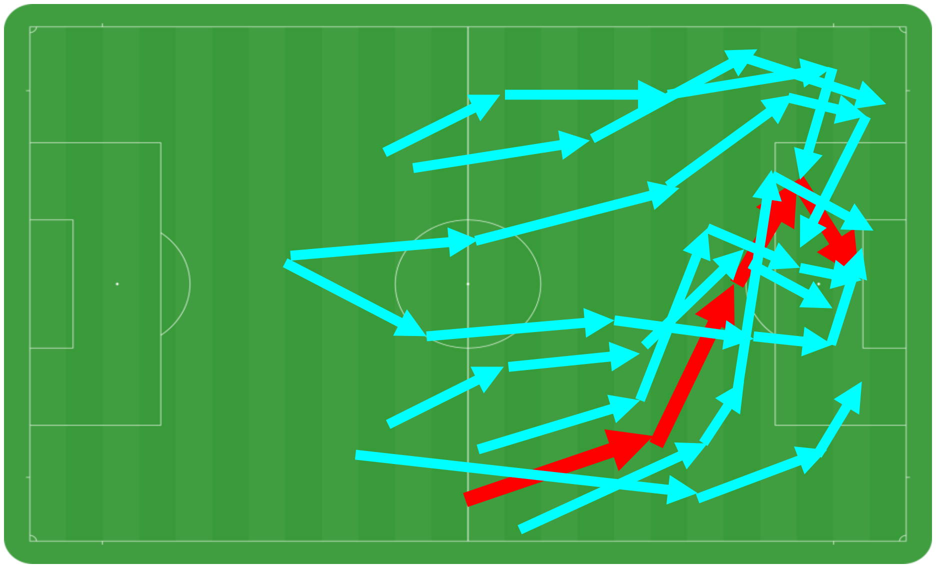
- Approach
  1. Split matches in phases
  2. Rate phases
  3. Distribute phase rating over individual actions
  4. Aggregate players ratings over season

Divide Match into Phases



Split event stream based on change of possession

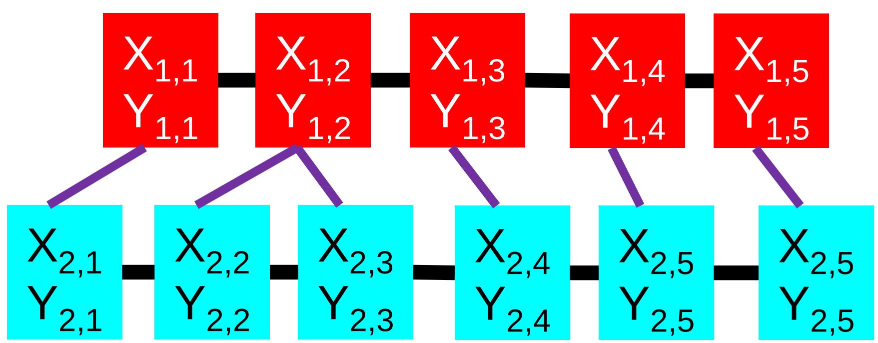
Rating Phases



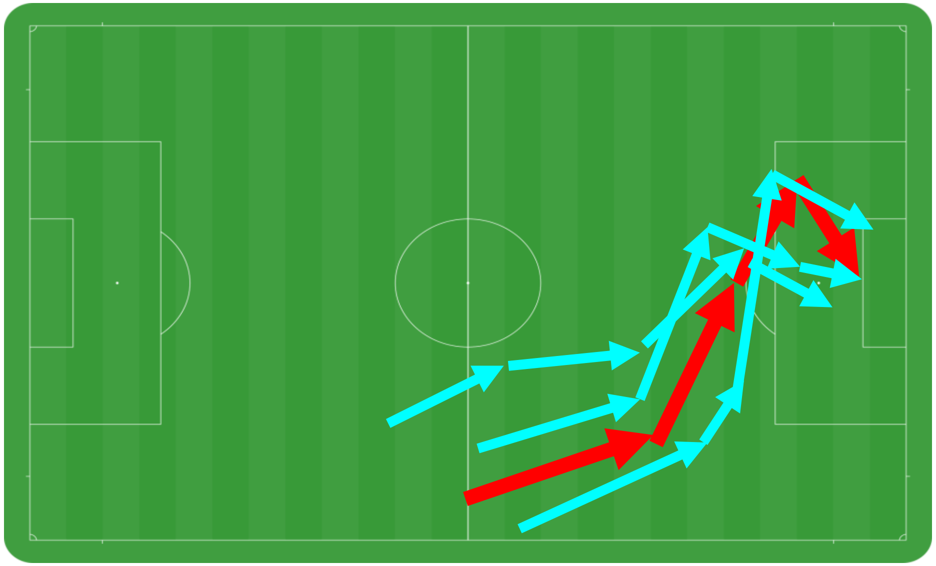
Question: How good is this phase?

Answer: Compare to what happened in similar phases

Similarity metric: Dynamic Time Warping on event positions



Rating Phases

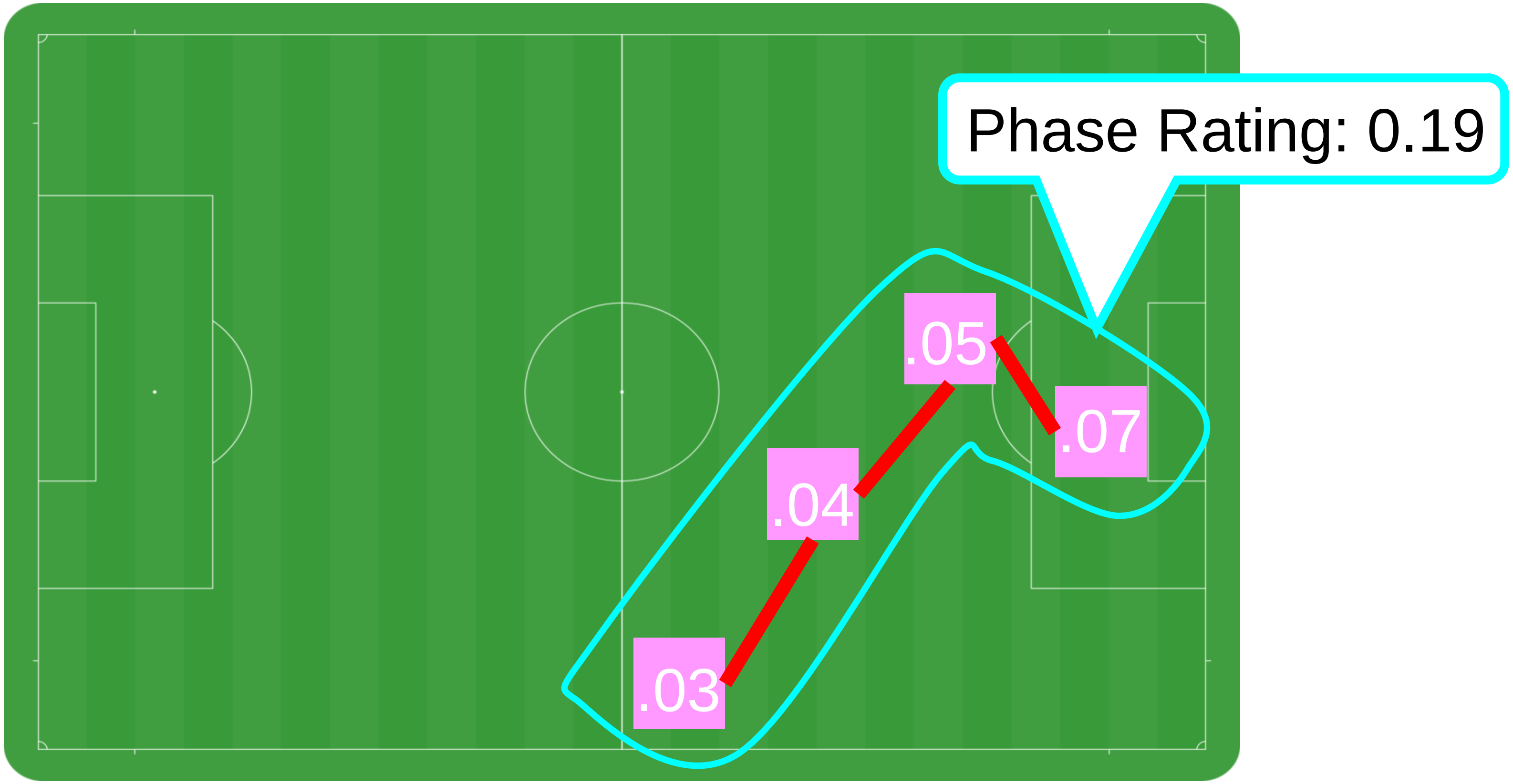


Rating phases:

1. Find *k* most similar phases (e.g., 100)
2. Of these, count how many result in a goal (e.g., 6)

$$Rating(phase) = \frac{6\ goals}{100\ similar\ phases} = 0.06$$

Distribute Phase Rating Across Its Constituent Actions



Actions at end are more important: Exponential decay

Top 10 Players: EPL 2016-2017

Rank	Team	Player	Rating Per 90	Goals Per 90	Assists Per 90
1	Arsenal	Alexis Sanchez	0.289	0.478	0.147
2	West Ham	Dimitri Payet	0.279	0.315	0.420
3	West Ham	Mauro Zarate	0.262	0.342	0.000
4	Chelsea	Willian	0.249	0.164	0.196
5	Liverpool	Philippe Coutinho	0.244	0.359	0.225
6	Arsenal	Santi Cazorla	0.240	0.000	0.209
7	Arsenal	Mesut Ozil	0.240	0.177	0.561
8	Sunderland	Wahbi Khazri	0.240	0.167	0.084
9	Aston Villa	Rudy Gestede	0.237	0.272	0.109
10	Man City	Kevin De Bruyne	0.233	0.315	0.404

Outline

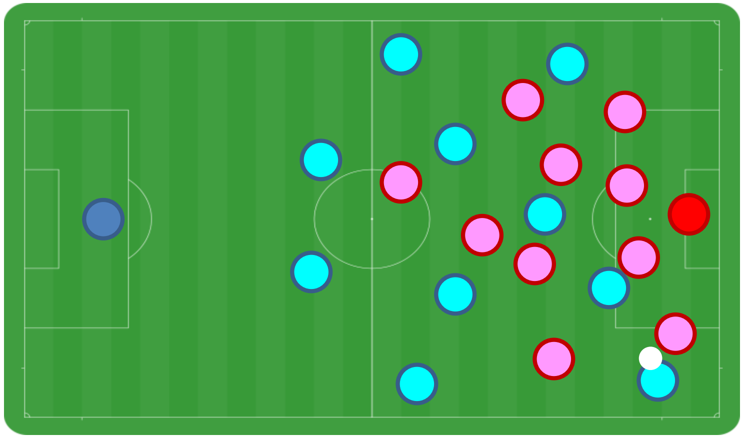
- **Rating players:** Assign a rating to each action a player performs in a match

- **Understand strategy:** Discover patterns from player tracking data

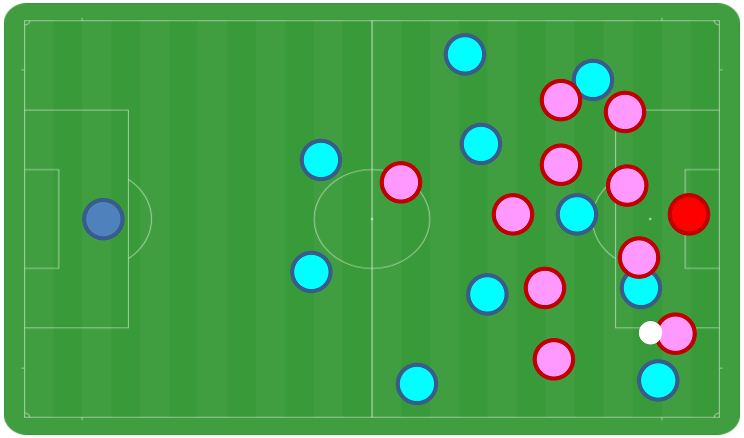
Discover Offensive Strategies in Football Matches

- **Given:**
  - Event stream with type and location of all events (e.g., passes and shots)
  - Locations of all players and the ball (10 hz sample)
- **Find:** Typical offensive strategies

Time t



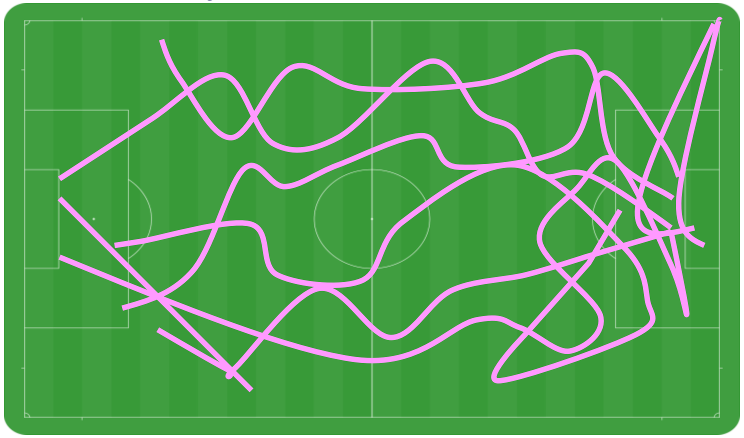
Time t+1

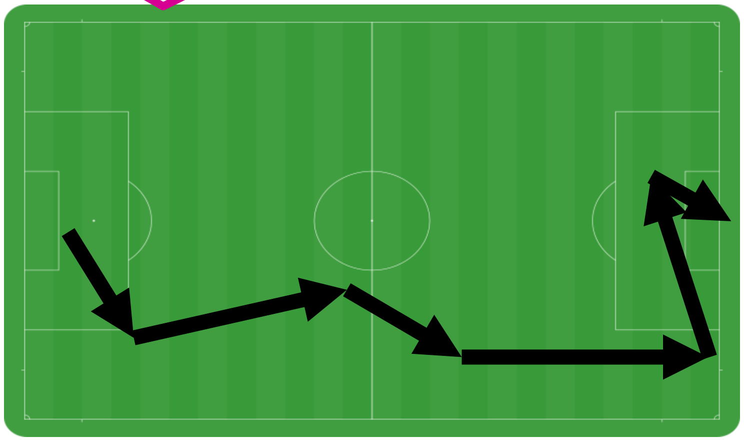


Big Picture Problem

Lots and lots of game play sequences

Subset of actions that commonly lead to shots





- Film study is time consuming
- Automation can help speed this up
- Computers good at finding patterns in large data sets

Challenges

- Relationships and how they change over time are important
  - Space
  - Interactions between players
- Order of events is important
- May want to generalize over players involved
- Exact same sequence of events unlikely to occur multiple times

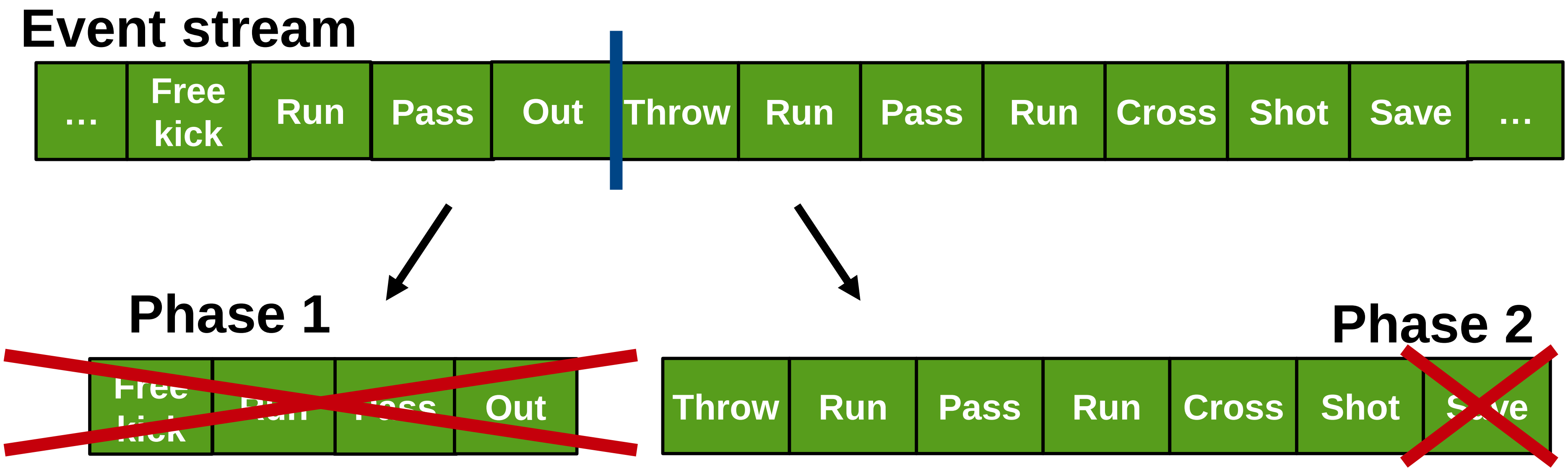
Important Steps

1. Data cleaning
2. Event stream preprocessing
3. Clustering data
4. Identifying important strategies

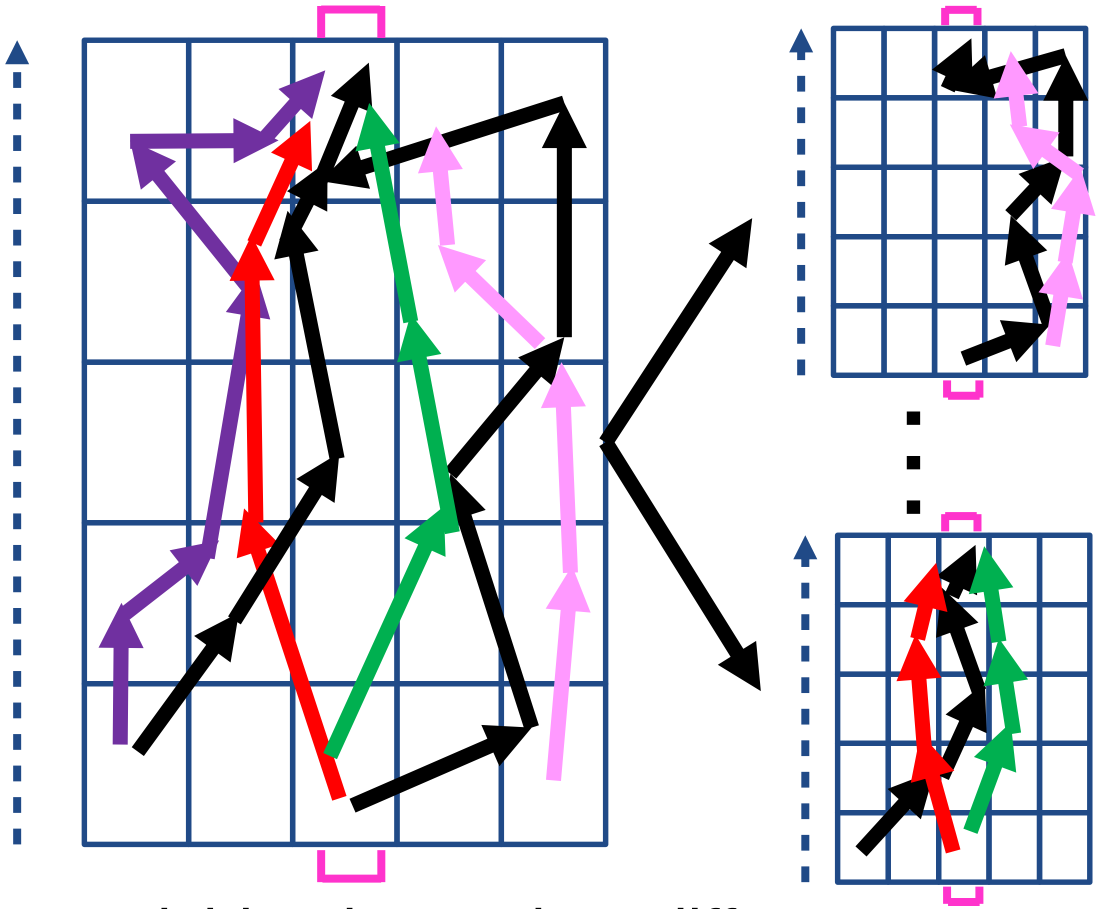
Step 1: Cleaning Data

- Outliers and incorrect values
  - Valid field coordinates
  - Player and ball movements seem “possible”
- Teams switch direction at half time: Normalize data such that team always attacks same goal
- Account for changes in data (e.g., position switches, new players, etc.)

Step 2: Event Stream Preprocessing



Step 3: Clustering



Three Benefits

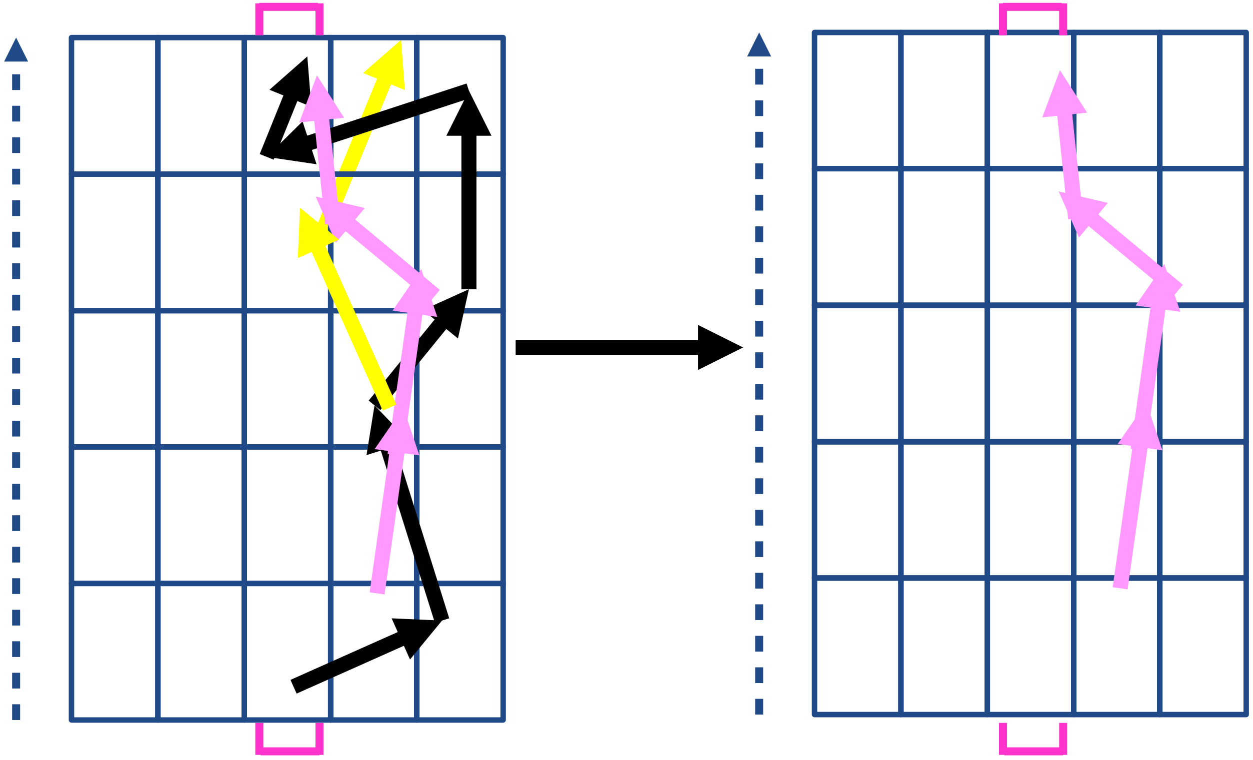
1. Teams employ multiple strategies
2. Generalize from a specific location
3. Subsequent step more computationally efficient

Divide phases into different groups such that the phases in a group are “similar”

Step 4: Finding Interesting Sequences

Within each cluster, find frequently occurring subsequences

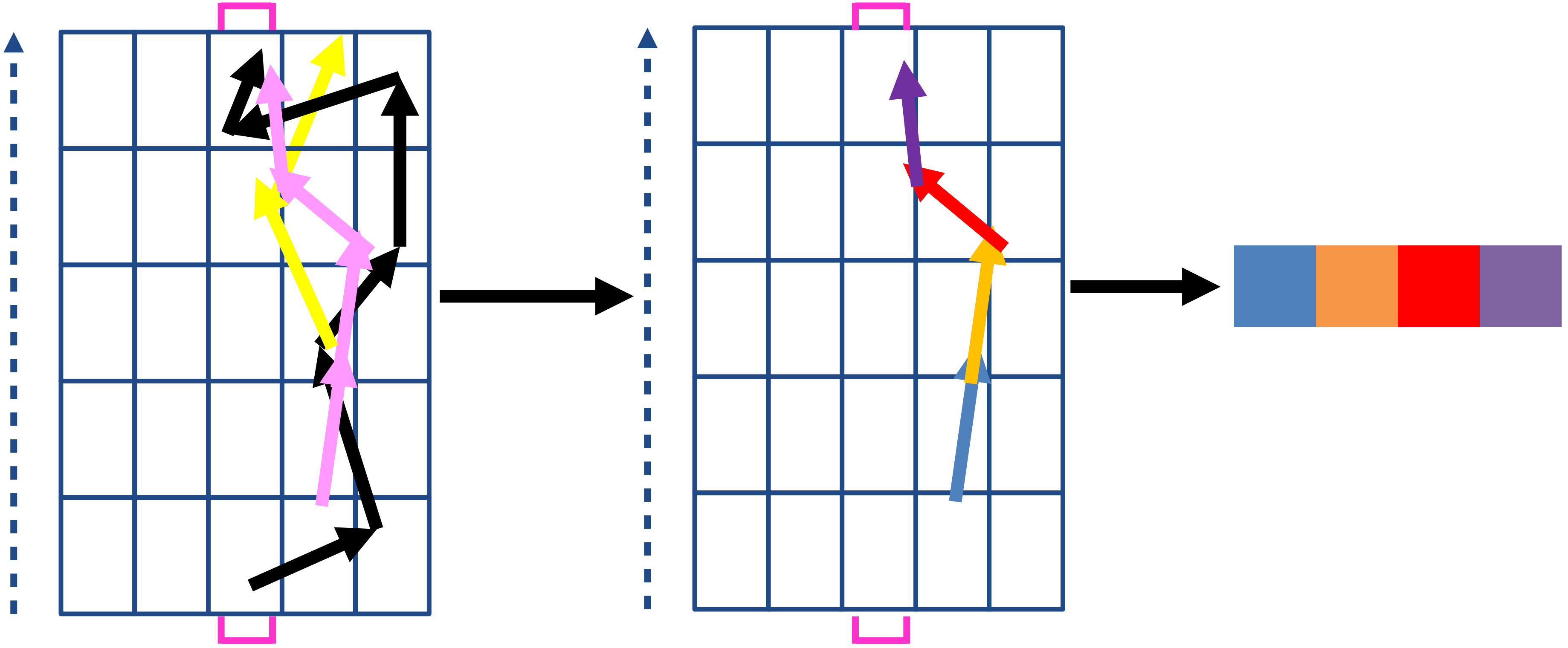
Cluster 1



Step 4: Finding Interesting Sequences

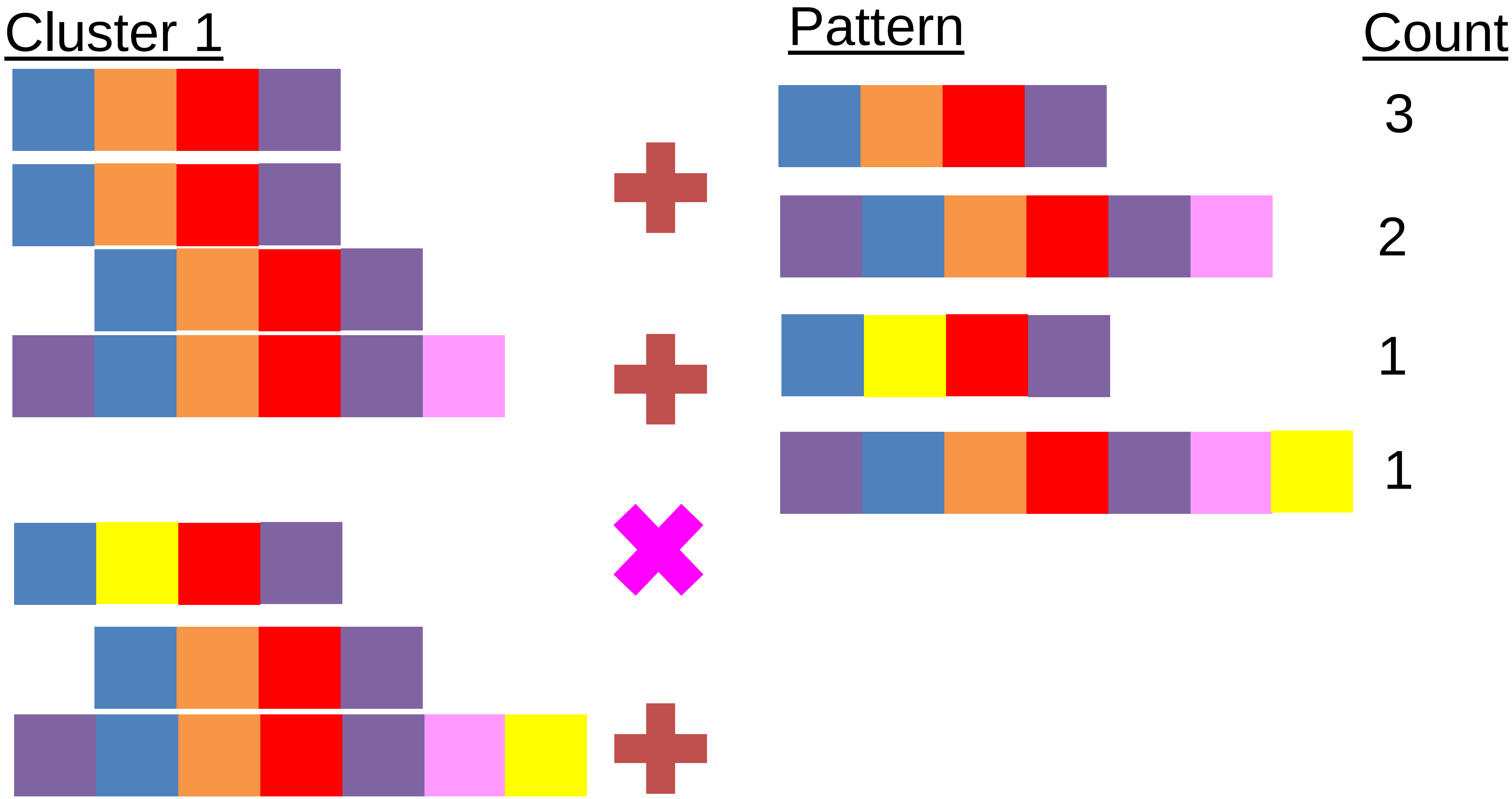
Within each cluster, find frequently occurring subsequences

Cluster 1

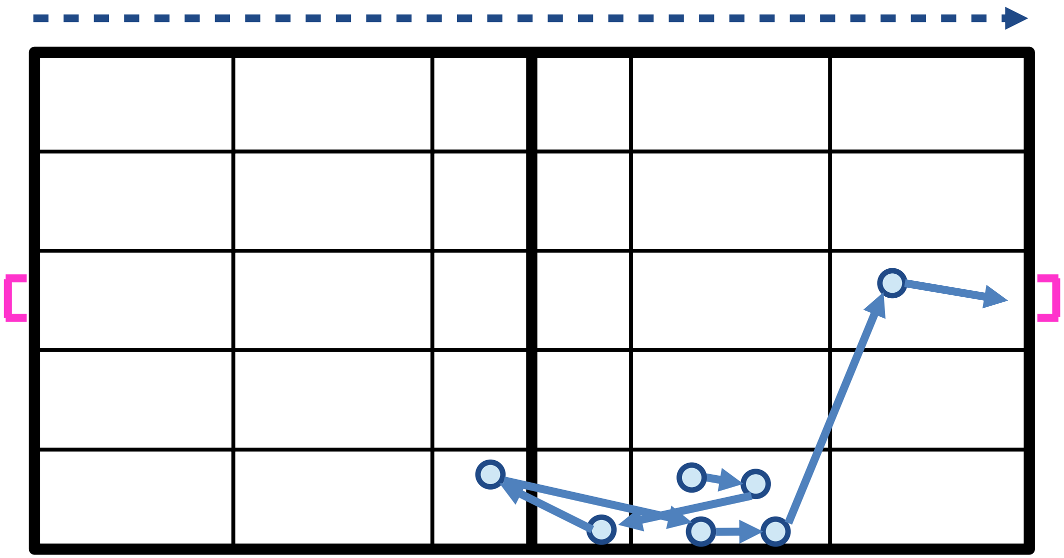


Step 4: Finding Interesting Sequences

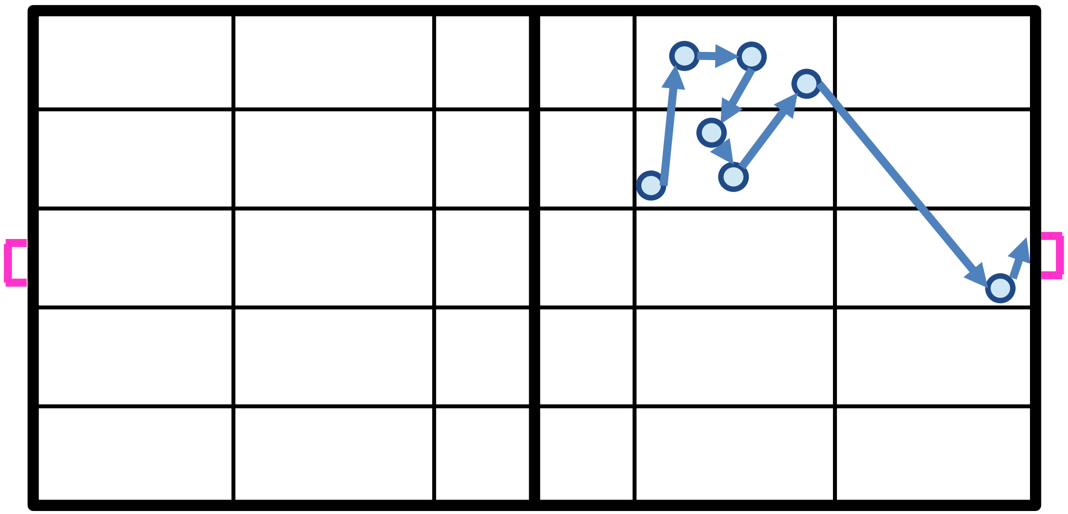
Within each cluster, find frequently occurring subsequences



Two Representative Patterns



An attack down the right flank



An attack down the left flank

Summary

- Focus on learning models from data
  - Expand frontiers of what is possible
  - Account for real world problems
  - Modeling structured data

- Applications drive research agenda
  - Health: ADRs
  - Sports
  - …

Questions?