Learning from Labeled and Unlabeled Data part 2: coupled training

Machine Learning 10-601

April 1, 2009

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

Last Time…

1. Unlabeled can help EM learn Bayes nets for P(X,Y)
   - If we assume the Bayes net structure is correct

2. Using unlabeled data to reweight labeled examples gives better approximation to true error
   - If we assume examples drawn from stationary P(X)

3. Use unlabeled data to detect/preempt overfitting
   - Based on distance metric, triangle inequality
   - If we assume priors over H that correctly order hypotheses

When can Unlabeled Data help supervised learning?

Problem setting (the PAC learning setting):

- Set *X* of instances drawn from unknown distribution *P(X)*
- Wish to learn target function *f: X→ Y* (or, P(Y|X))
- Given a set *H* of possible hypotheses for *f*

Given:

- i.i.d. labeled examples $L = \{\langle x_1, y_1\rangle \ldots \langle x_m, y_m\rangle\}$
- i.i.d. unlabeled examples $U = \{x_{m+1}, \ldots x_{m+n}\}$

Wish to find hypothesis with lowest true error:

$$\hat{f} \leftarrow \arg\min_{h \in H} \Pr_{x \in P(X)}[h(x) \neq f(x)]$$

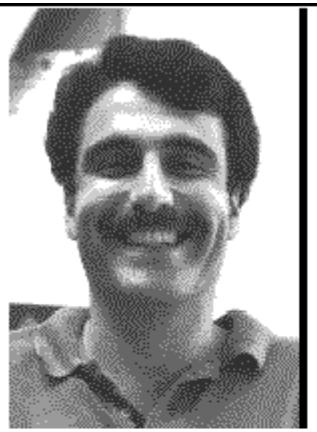
Idea 4: CoTraining, Coupled Training

- In some settings, available data features are redundant and we can train two classifiers based on disjoint features
- In this case, the two classifiers should agree on the classification for each unlabeled example
- Therefore, we can use the unlabeled data to constrain joint training of both classifiers

Redundantly Sufficient Features

Professor Faloutsos

my advisor



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

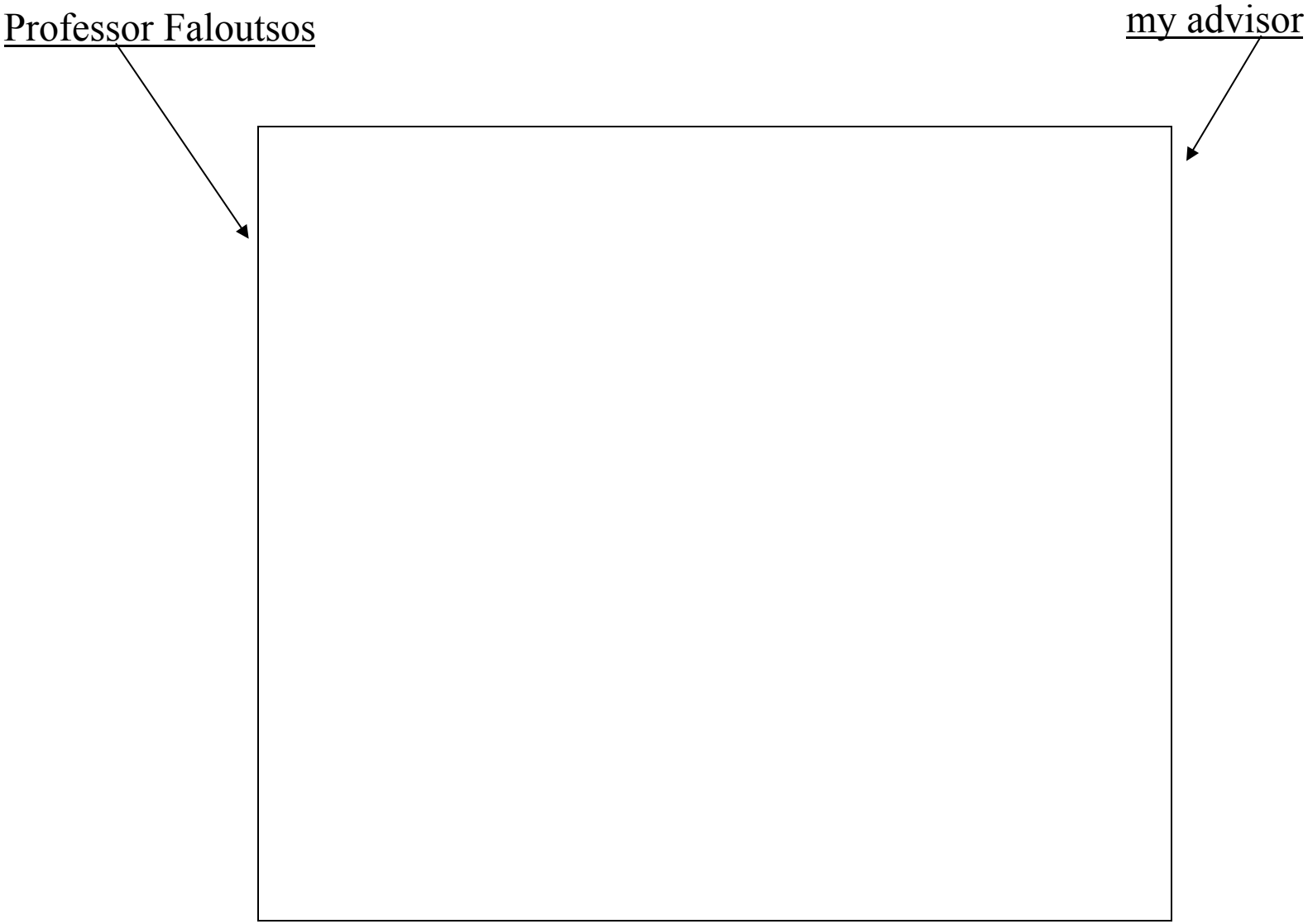
Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

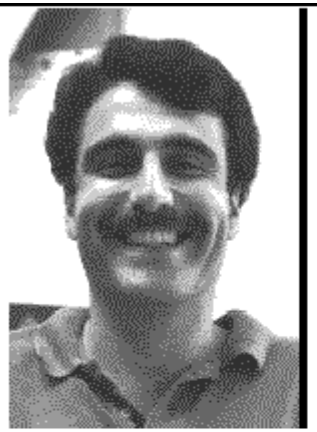
Research Interests:

- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

Redundantly Sufficient Features



Redundantly Sufficient Features



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

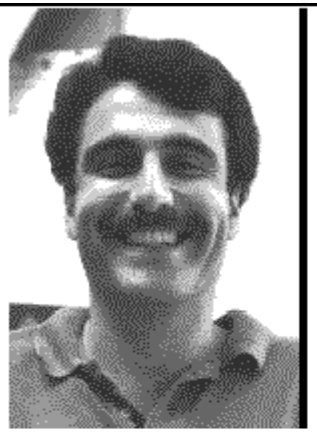
Research Interests:

- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

Redundantly Sufficient Features

Professor Faloutsos

my advisor



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

Research Interests:

- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

CoTraining Algorithm #1

[Blum&Mitchell, 1998]

Given: labeled data L,

unlabeled data U

Loop:

Train g1 (hyperlink classifier) using L

Train g2 (page classifier) using L

Allow g1 to label *p* positive, *n* negative examps from U

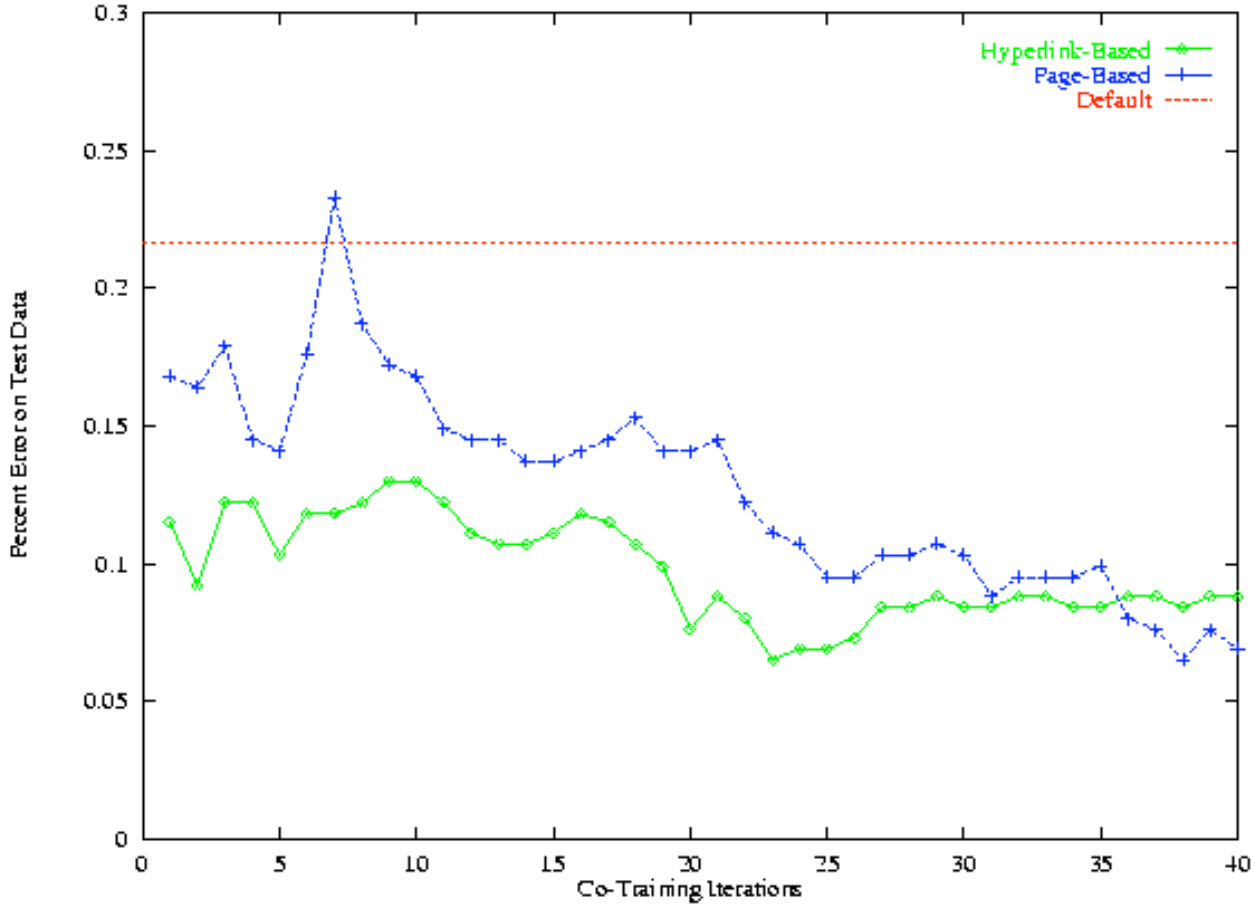
Allow g2 to label *p* positive, *n* negative examps from U

Add these self-labeled examples to L

CoTraining: Experimental Results

- begin with 12 labeled web pages (academic course)
- provide 1,000 additional unlabeled web pages
- average error: learning from labeled data 11.1%;
- average error: cotraining 5.0%

Typical run:



CoTraining setting:

- wish to learn f: X → Y, given L and U drawn from P(X)
- features describing X can be partitioned (X = X1 x X2)

  such that f can be computed from either X1 or X2

$$(\exists g_1, g_2)(\forall x \in X) \quad g_1(x_1) = f(x) = g_2(x_2)$$

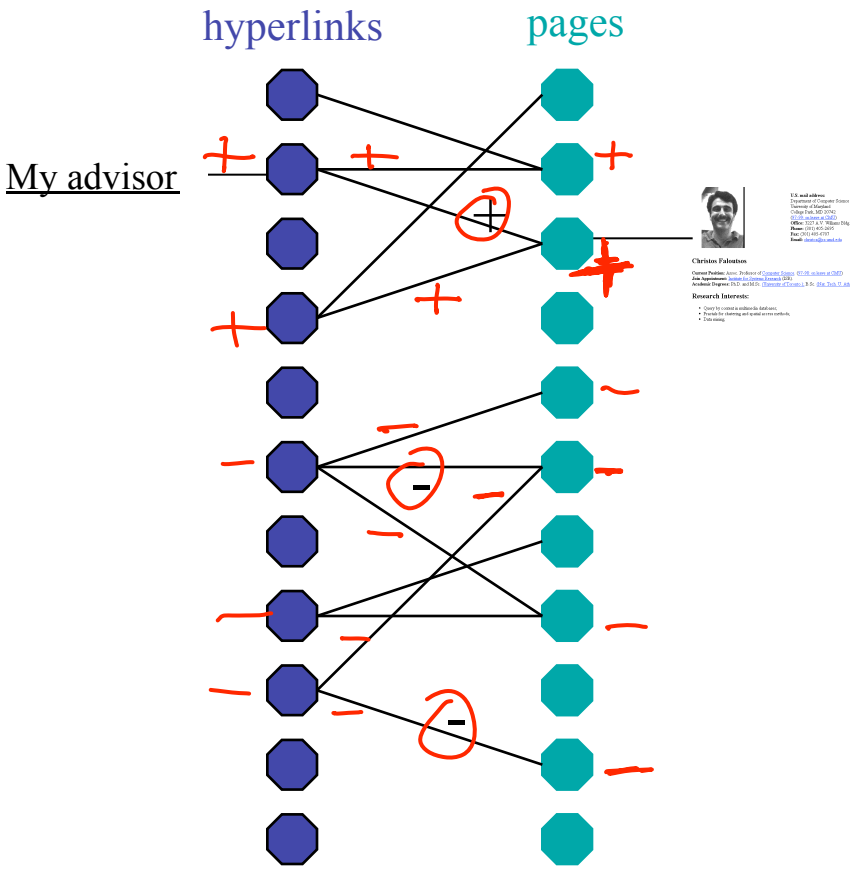
One result [Blum&Mitchell 1998]:

- If
  - X1 and X2 are conditionally independent given Y
  - f is PAC learnable from noisy *labeled* data
- Then
  - f is PAC learnable from weak initial classifier plus polynomial number of *unlabeled* examples

Classifier with accuracy > 0.5

Example: Co-Training Rote Learners

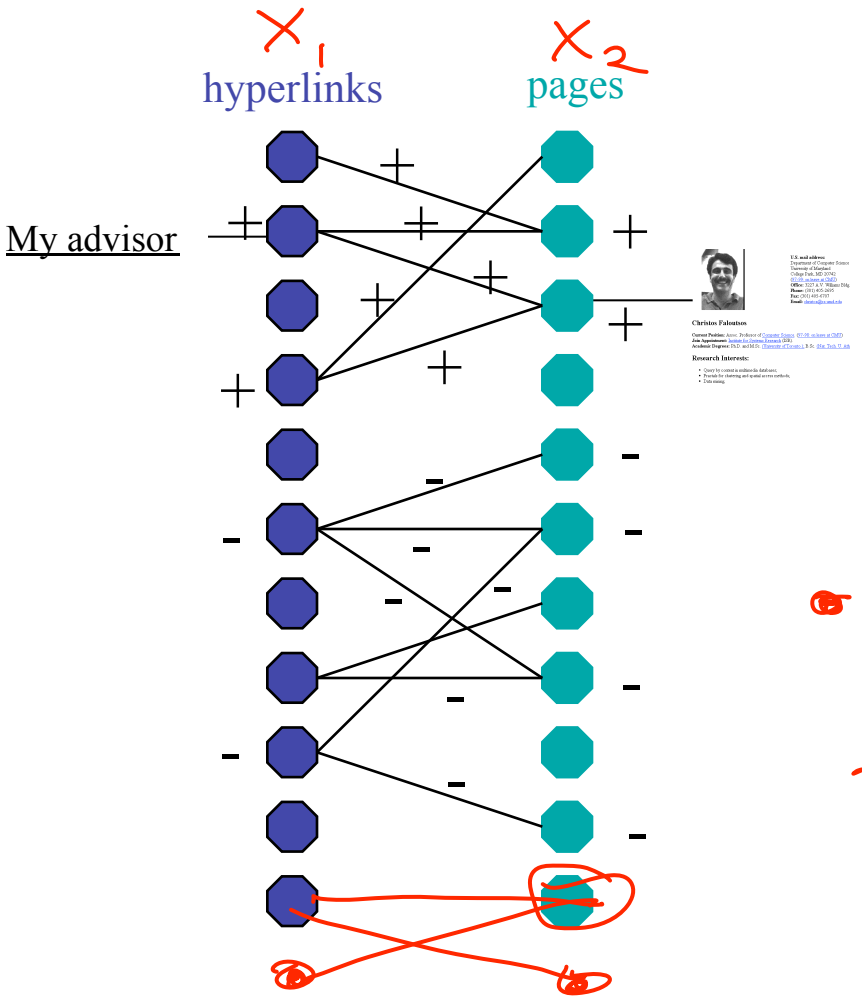
f1:hyperlink → Y, f2: page → Y



Example: Co-Training Rote Learner

m labeled examps.

$E[err] \rightarrow$



- $P(x)$

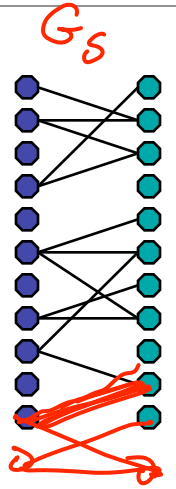
$\langle x_1, x_2 \rangle$

Expected Rote CoTraining error given *m* examples

$$CoTraining \quad setting:$$
$$learn \quad f: X \rightarrow Y$$
$$where \quad X = X_1 \times X_2$$
$$where \quad x \quad drawn \quad from \quad unknown \quad distribution$$
$$and \quad \exists g_1, g_2 \quad (\forall x) g_1(x_1) = g_2(x_2) = f(x)$$

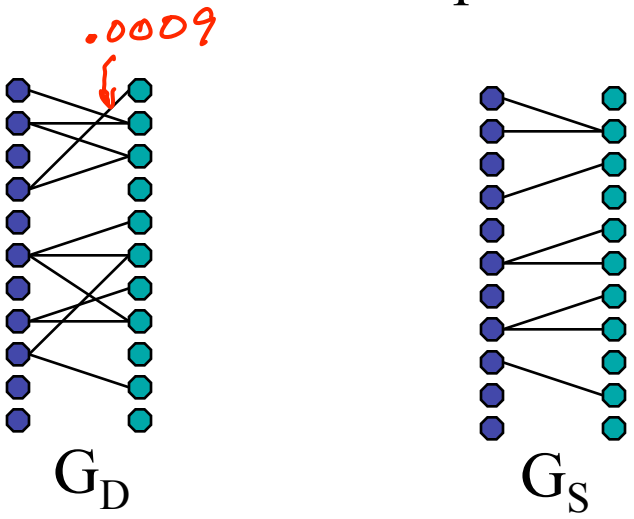
$$E[error] \leq \sum_j P(x \in g_j)(1 - P(x \in g_j))^m$$

Where g_j is the jth connected component of graph of L+U, m is number of labeled examples



How many *unlabeled* examples suffice?

Want to assure that connected components in the underlying distribution, G_D, are connected components in the observed sample, G_S



$O(\log(N)/\alpha)$ examples assure that with high probability, G_S has same connected components as G_D [Karger, 94]

N is size of G_D, α is min cut over all connected components of G_D

PAC Generalization Bounds on CoTraining

[Dasgupta et al., NIPS 2001]

This theorem assumes X1 and X2 are conditionally independent given Y

Theorem 1 *With probability at least* $1-\delta$ *over the choice of the sample* S*, we have that for all* h_1 *and* h_2*, if* $\gamma_i(h_1,h_2,\delta) > 0$ *for* $1 \leq i \leq k$ *then (a)* f *is a permutation and (b) for all* $1 \leq i \leq k$*,*

$$P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp) \leq \frac{\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp) + \epsilon_i(h_1,h_2,\delta)}{\gamma_i(h_1,h_2,\delta)}.$$

The theorem states, in essence, that if the sample size is large, and h_1 and h_2 largely agree on the unlabeled data, then $\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp)$ is a good estimate of the error rate $P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp)$.

$$\gamma_i(h_1,h_2,\delta) = \widehat{P}(h_1 = i \mid h_2 = i, h_1 \neq \perp) - \widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp) - 2\epsilon_i(h_1,h_2,\delta)$$

$$\epsilon_i(h_1,h_2,\delta) = \sqrt{\frac{(\ln 2)(|h_1|+|h_2|) + \ln\frac{2k}{\delta}}{2|S(h_2 = i,\ h_1 \neq \perp)|}}$$

PAC Generalization Bounds on CoTraining

[Dasgupta et al., NIPS 2001]

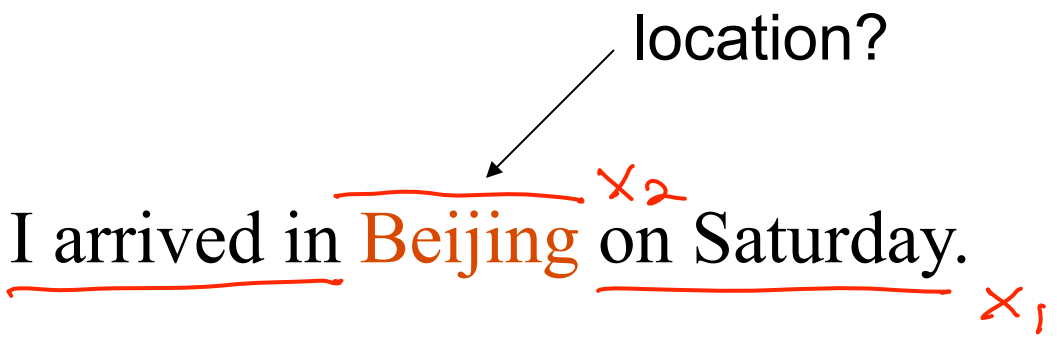
This theorem assumes X1 and X2 are conditionally independent given Y

Theorem 1 *With probability at least* $1-\delta$ *over the choice of the sample* S*, we have that for all* h_1 *and* h_2*, if* $\gamma_i(h_1, h_2, \delta) > 0$ *for* $1 \leq i \leq k$ *then (a)* f *is a permutation and (b) for all* $1 \leq i \leq k$*,*

$$P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp) \leq \frac{\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp) + \epsilon_i(h_1, h_2, \delta)}{\gamma_i(h_1, h_2, \delta)}.$$

The theorem states, in essence, that if the sample size is large, and h_1 and h_2 largely agree on the unlabeled data, then $\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp)$ is a good estimate of the error rate $P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp)$.

Example 2: Learning to extract named entities

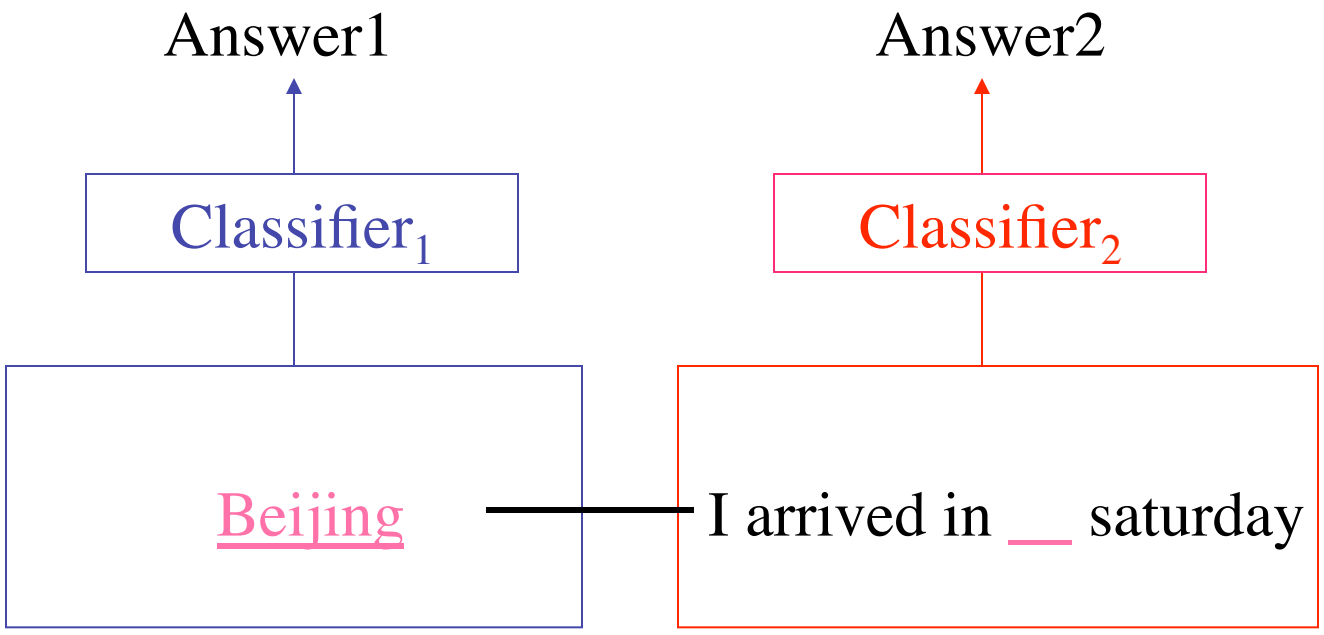


If: "I arrived in <X> on Saturday."

Then: Location(X)

Co-Training for Named Entity Extraction (i.e.,classifying which strings refer to people, places, dates, etc.)

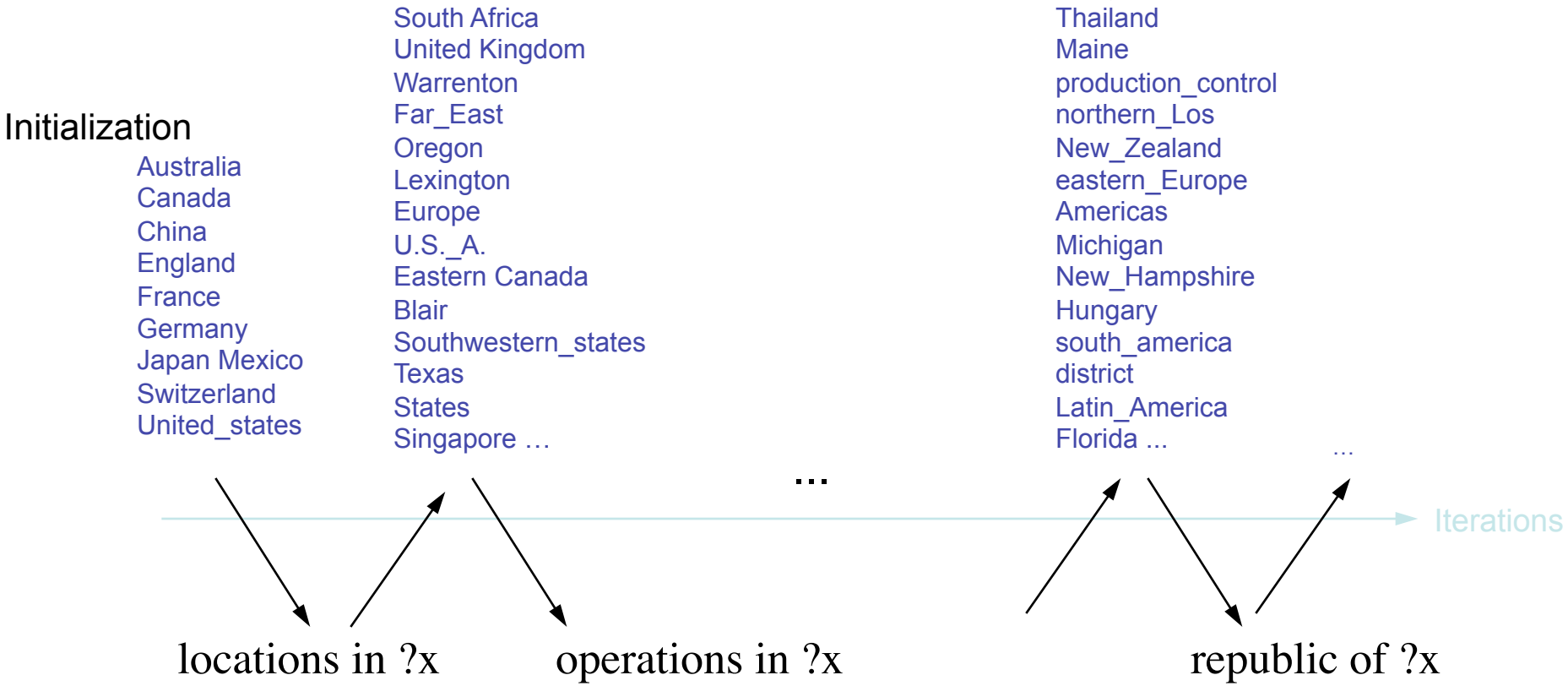
[Riloff&Jones 98; Collins et al., 98; Jones 05]



I arrived in **Beijing** saturday.

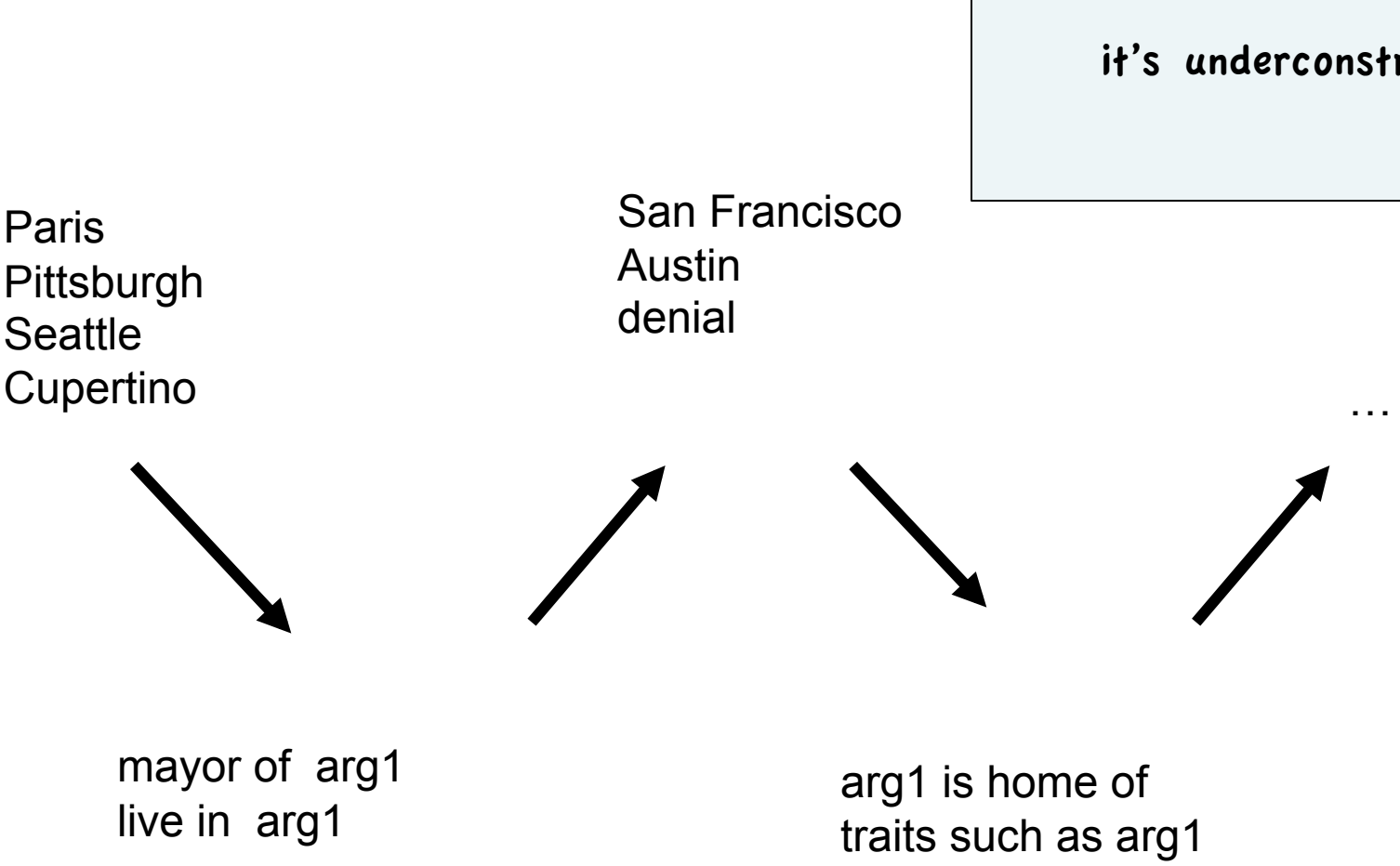
Bootstrap learning to extract named entities

[Riloff and Jones, 1999], [Collins and Singer, 1999], ...

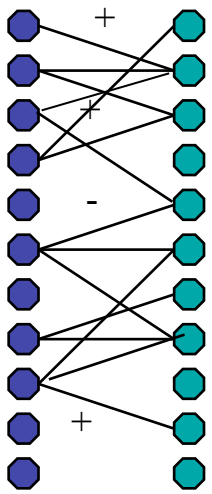


The Problem with Semi-Supervised Bootstrap Learning

it's underconstrained!!



What if CoTraining Assumption Not Perfectly Satisfied?



- Idea: Want classifiers that produce a *maximally consistent* labeling of the data
- If learning is an optimization problem, what function should we optimize?

What Objective Function?

$$E = E1 + E2$$

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

Error on labeled examples

What Objective Function?

$$E = E1 + E2 + c_3 E3$$

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

$$E3 = \sum_{x\in U} (\hat{g}_1(x_1) - \hat{g}_2(x_2))^2$$

Error on labeled examples

Disagreement over unlabeled

What Objective Function?

$$E = E1 + E2 + c_3 E3 + c_4 E4$$

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

Error on labeled examples

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

Disagreement over unlabeled

$$E3 = \sum_{x\in U} (\hat{g}_1(x_1) - \hat{g}_2(x_2))^2$$

Misfit to estimated class priors

$$E4 = \left(\left(\frac{1}{|L|} \sum_{<x,y>\in L} y \right) - \left(\frac{1}{|L|+|U|} \sum_{x\in L\cup U} \frac{\hat{g}_1(x_1) + \hat{g}_2(x_2)}{2} \right) \right)^2$$

What Function Approximators?

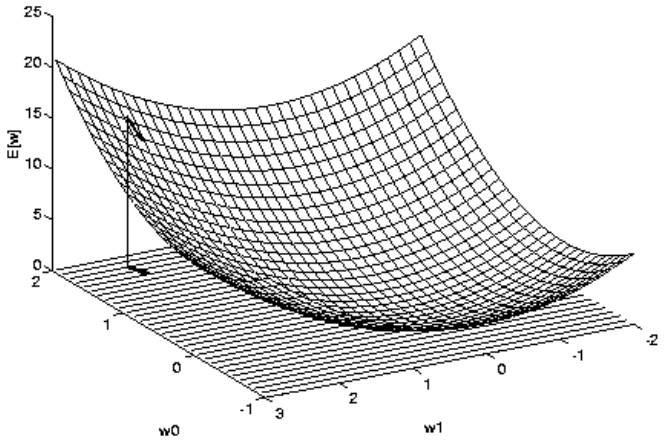
What Function Approximators?

$$\hat{g}_1(x) = \frac{1}{1 + e^{\sum_j w_{j,1} x_j}} \qquad \hat{g}_2(x) = \frac{1}{1 + e^{\sum_j w_{j,2} x_j}}$$

- Same fn form as Naïve Bayes, Max Entropy
- Use gradient descent to simultaneously learn g1 and g2, directly minimizing E = E1 + E2 + E3 + E4
- No word independence assumption, use both labeled and unlabeled data

Gradient CoTraining

$$\hat{g}_1(x) = \frac{1}{1 + e^{\sum_j w_{j,1} x_j}} \qquad \hat{g}_2(x) = \frac{1}{1 + e^{\sum_j w_{j,2} x_j}}$$



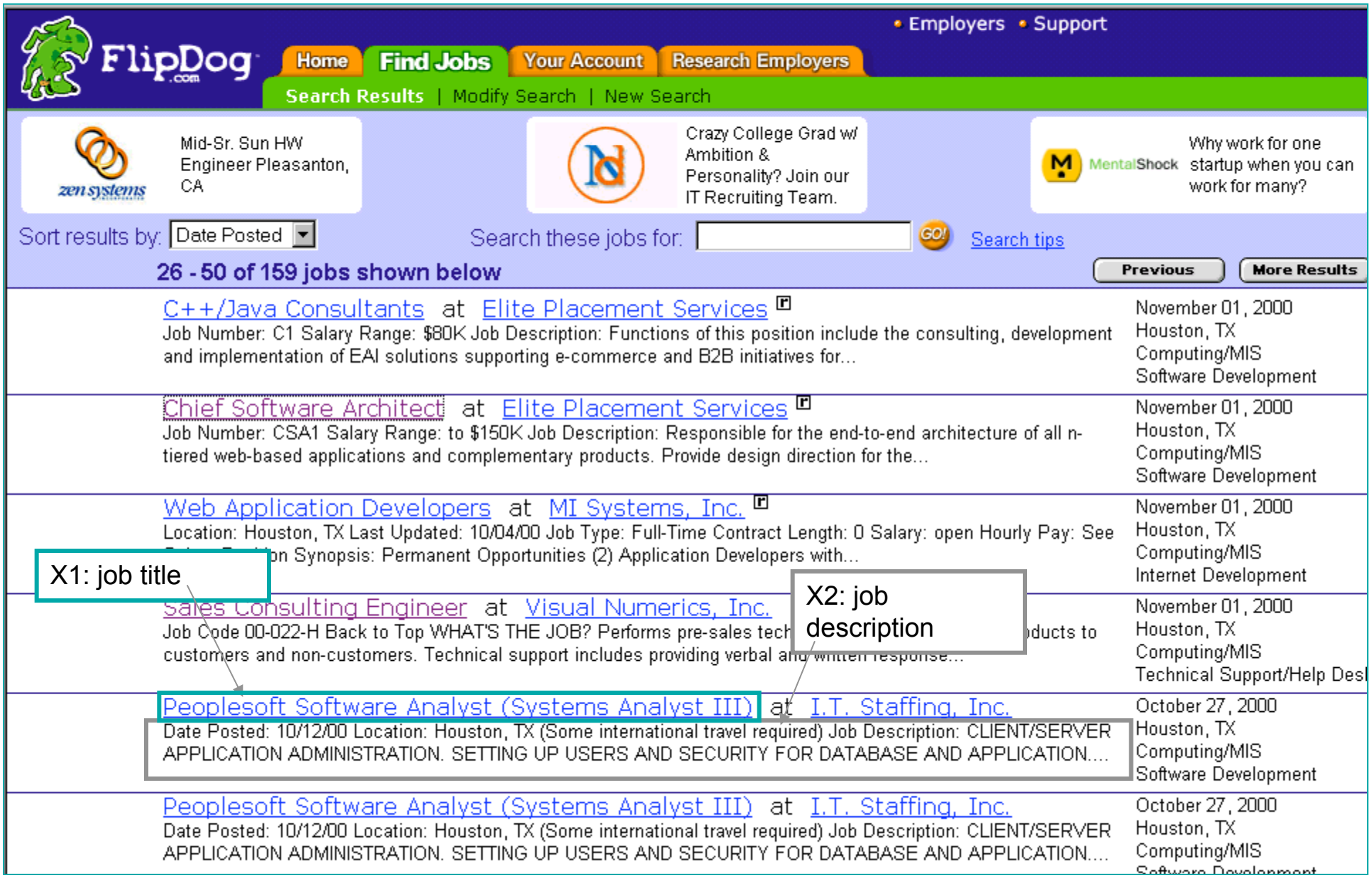
Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

Training rule:

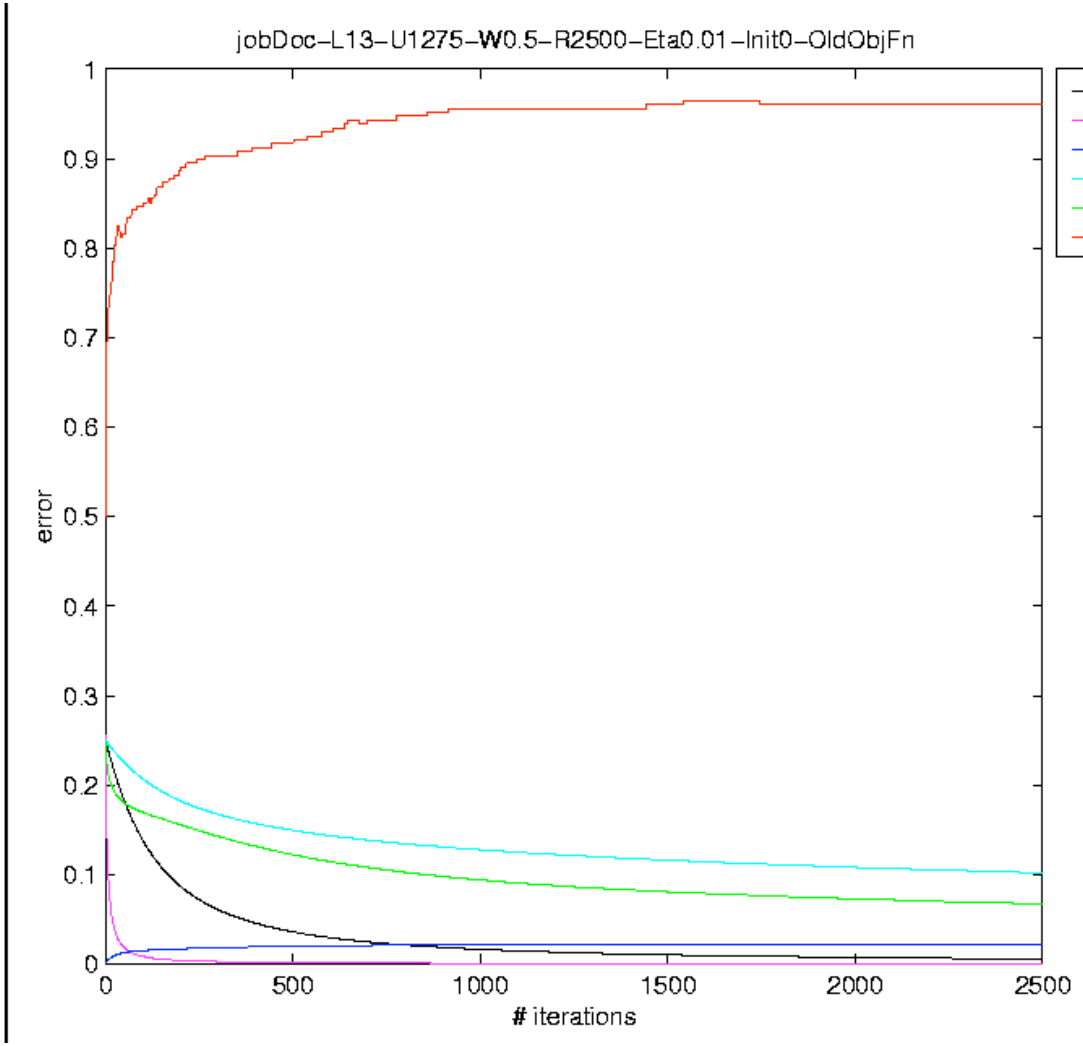
$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

Classifying Jobs for FlipDog



Gradient CoTraining

Classifying FlipDog job descriptions: SysAdmin vs. WebProgrammer



Final Accuracy

Labeled data alone: 86%

CoTraining: 96%

CoTraining Summary

- Unlabeled data improves supervised learning when example features are redundantly sufficient
  - Family of algorithms that train multiple classifiers

- Theoretical results
  - Expected error for rote learning
  - If X1,X2 conditionally independent given Y, Then
    - PAC learnable from weak initial classifier plus unlabeled data
    - disagreement between g1(x1) and g2(x2) bounds final classifier error

- Many real-world problems of this type
  - Semantic lexicon generation [Riloff, Jones 99], [Collins, Singer 99]
  - Web page classification [Blum, Mitchell 98]
  - Word sense disambiguation [Yarowsky 95]
  - Speech recognition [de Sa, Ballard 98]
  - Visual classification of cars [Levin, Viola, Freund 03]

What you should know

1. Unlabeled can help EM learn Bayes nets for P(X,Y)
   - If we assume the Bayes net structure is correct

2. Using unlabeled data to reweight labeled examples gives better approximation to true error
   - If we assume examples drawn from stationary P(X)

3. Use unlabeled data to detect/preempt overfitting
   - Based on distance metric, triangle inequality
   - If we assume priors over H that correctly order hypotheses

4. CoTraining multiple classifiers, using unlabeled data as constraints
   - If we assume redundantly sufficient features, with different conditional distributions given the class

Further Reading

- Semi-Supervised Learning, O. Chapelle, B. Sholkopf, and A. Zien (eds.), MIT Press, 2006. (excellent book)

- EM for Naïve Bayes classifiers: K.Nigam, et al., 2000. "Text Classification from Labeled and Unlabeled Documents using EM", *Machine Learning,* 39, pp.103—134.

- CoTraining: A. Blum and T. Mitchell, 1998. "Combining Labeled and Unlabeled Data with Co-Training," *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98).*
- S. Dasgupta, et al., "PAC Generalization Bounds for Co-training", *NIPS 2001*

- Model selection: D. Schuurmans and F. Southey, 2002. "Metric-Based methods for Adaptive Model Selection and Regularization," Machine Learning, 48, 51—84.

Toward Never-Ending Learning of Semantic Knowledge

Justin Betteridge, Andrew Carlson, Estevam R. Hruschka Jr.,
Tom M. Mitchell

(with help from Sue Ann Hong, Sophie Wang, Richard Wang)

Carnegie Mellon University

March 2009

Our Goal: Never-Ending Language Learning

Goal:

- run 24x7, forever
- each day:
    1. extract more facts from the web to populate and extend initial ontology
    2. learn to read better than yesterday

Our Goal: Never-Ending Language Learning

Goal:

- run 24x7, forever
- each day:
    1. extract more facts from the web to populate initial ontology
    2. learn to read better than yesterday

Today…

Given:

- initial ontology defining dozens of classes and relations
- 10-20 seed examples of each

Task:

- learn to extract / extract to learn
- running over 200M web pages, for a few days

Browse the KB

- ~ 18,000+ entities, ~ 30,000 extracted beliefs
- learned from 10-20 seed examples, 200M unlabeled web pages
- ~ 2 days computation on M45 cluster (thanks Yahoo!)

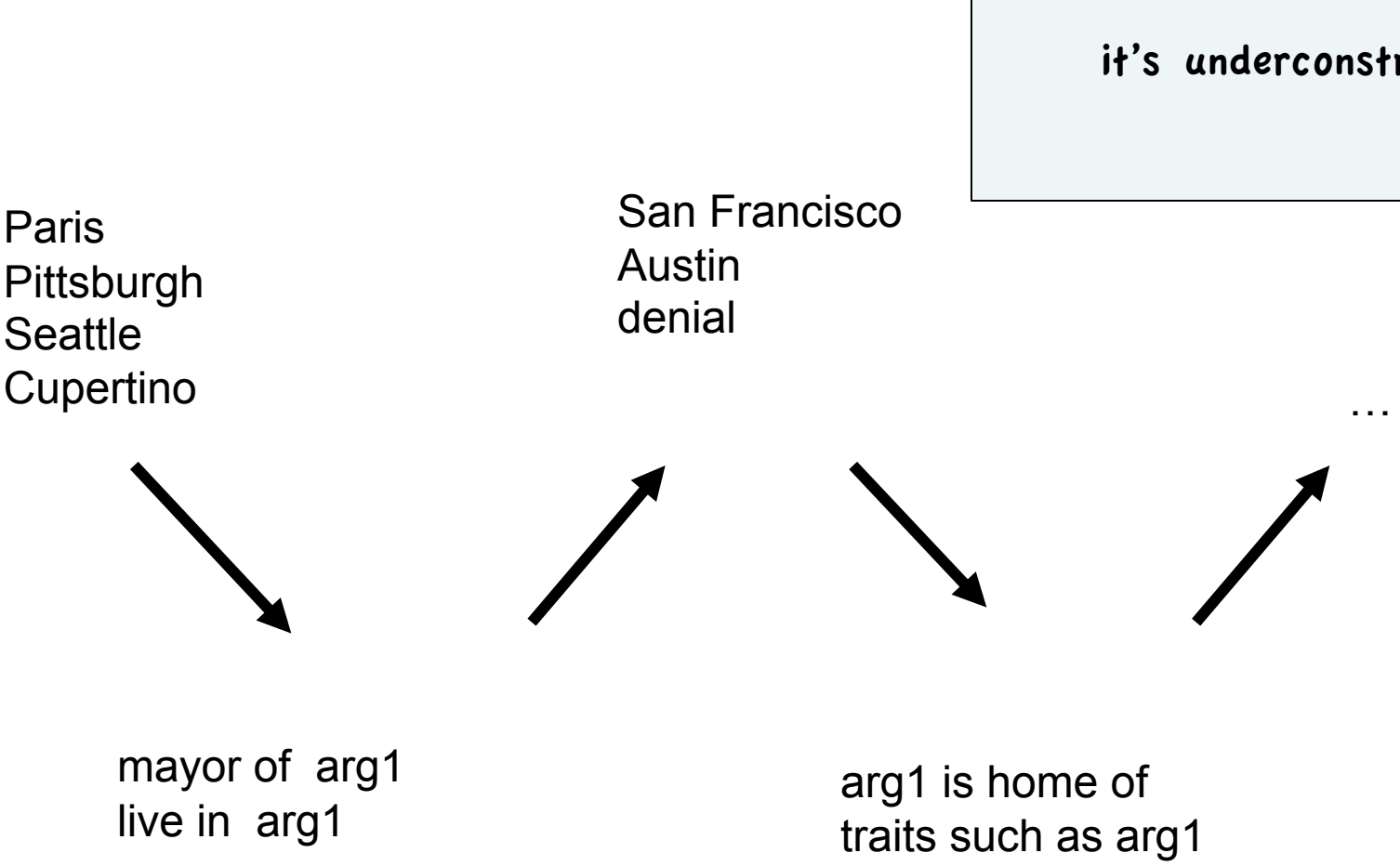
on the web:

initial

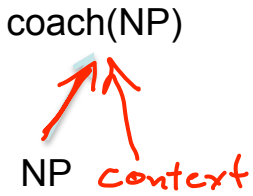
populated

The Problem with Semi-Supervised Bootstrap Learning

it's underconstrained!!

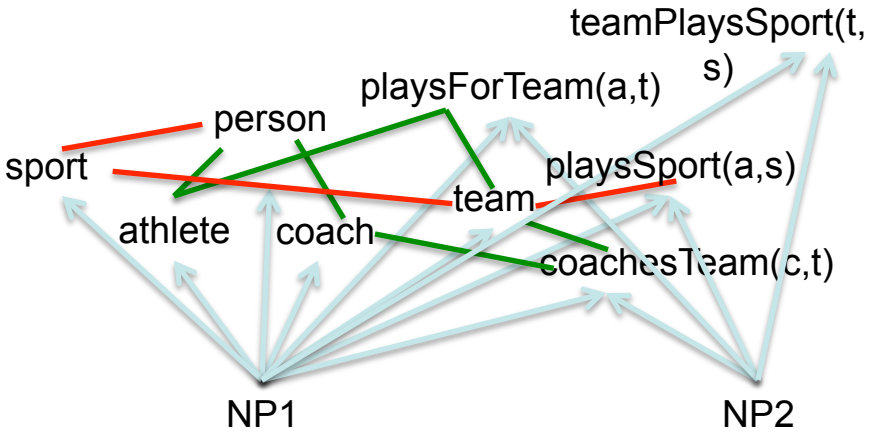


The Key to Accurate Semi-Supervised Learning



Krzyzewski coaches the Blue Devils.

hard (underconstrained) semi-supervised learning problem



Krzyzewski coaches the Blue Devils.

much easier (more constrained) semi-supervised learning problem

Constraining semi-supervised learning 1

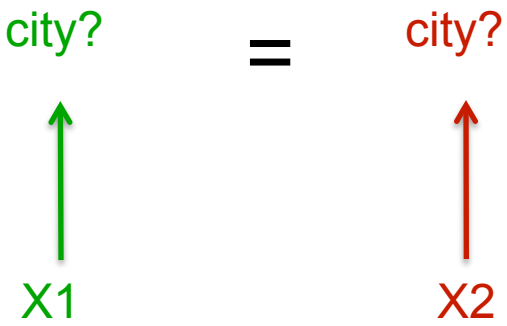
Wish to learn f : X → Y

e.g., city : NounPhraseInSentence → {0,1}

Constraint type 1 (co-training):

if X can be split into redundantly sufficient X1, X2

then learn both f1: X1 → Y, and f2: X2 → Y



X: Luke is mayor of Pittsburgh.

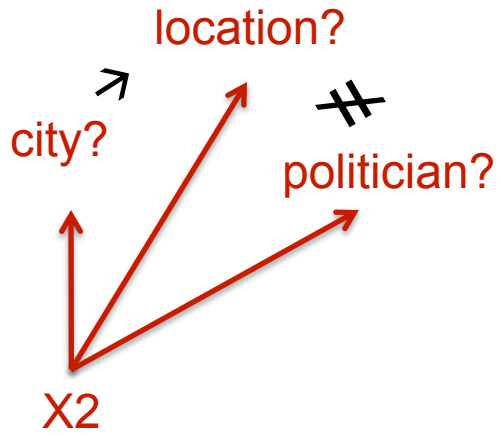
Constraining semi-supervised learning 2

Wish to learn f: X → Y

e.g., city: NounPhraseInSentence → {0,1}

Constraint type 2: couple training of multiple classes

Ontology provides coupling constraints



Luke is mayor of Pittsburgh.

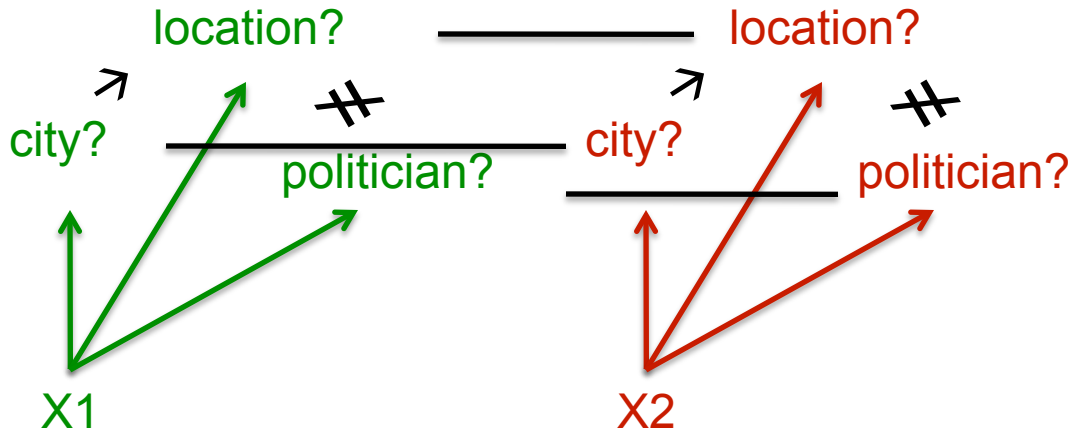
Constraining semi-supervised learning 2

Wish to learn f: X → Y

e.g., city: NounPhraseInSentence → {0,1}

Constraint type 2: couple training of multiple classes

Ontology provides coupling constraints



Luke is mayor of Pittsburgh.

Coupling unsupervised training of multiple functions

Cotraining: learn f: X→Y, by coupling training of

- f1: X1 → Y
- f2: X2 → Y

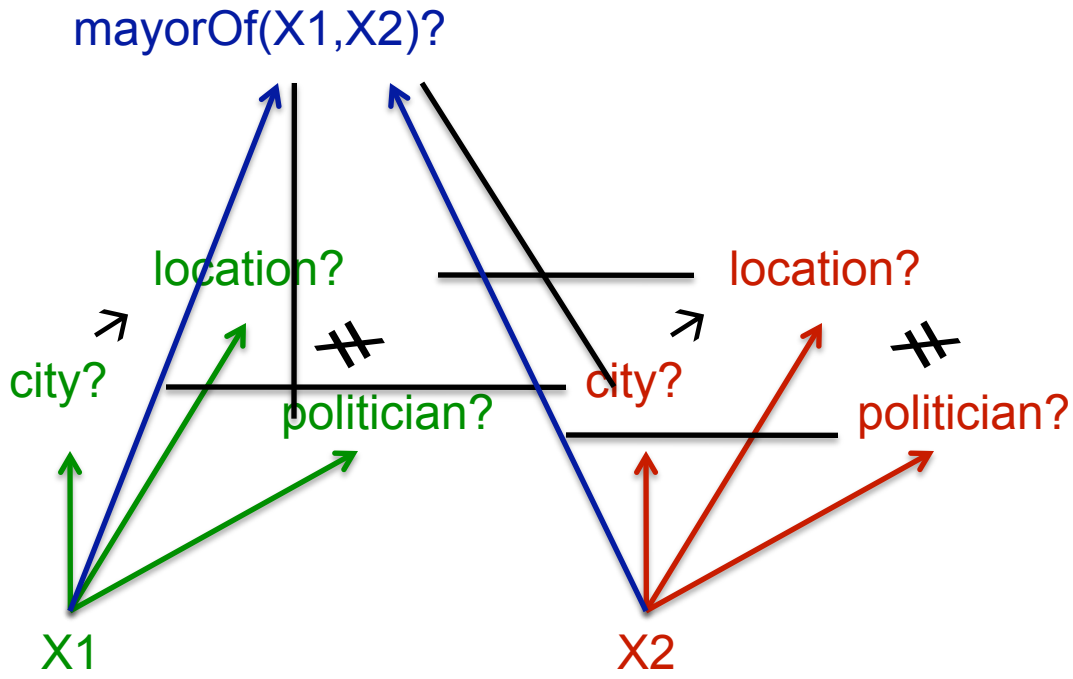
Coupled functions: learn f1: X → Y1, f2: X → Y2

where coupling is based on a constraint C(Y1,Y2)

- e.g., mutallyExclusive(Person, City)
- e.g., subset(Athlete, Person)

Constraining semi-supervised learning 3

Constraint type 3 (couple training of classes and relations)



Luke is mayor of Pittsburgh.

Coupled Bootstrap Learner algorithm

Algorithm 1: CBL Algorithm

Input: An ontology $\mathcal{O}$, and text corpus C
Output: Trusted instances/patterns for each predicate

SHARE initial instances/patterns among predicates;
for $i = 1, 2, \ldots, \infty$ **do**
  foreach *predicate* $p \in \mathcal{O}$ **do**
    EXTRACT new candidate instances/patterns;
    FILTER candidates;
    TRAIN instance/pattern classifiers;
    ASSESS candidates using trained classifiers;
    PROMOTE highest-confidence candidates;
  end
  SHARE promoted items among predicates;
end

In the **ontology**: categories, relations, seed instances and patterns, type information, mutual exclusion and subset relations

Sharing enforces mutual exclusion, subset relations, and type checking

Extraction (M45):
Arg1 HQ in Arg2 → (CBC || Toronto), (Adobe || San Jose), …

Filtering (M45):
CBC || Toronto → Not enough evidence
Micron || Boise → arg2 is headquarters for chipmaker arg1, arg1 of arg2, arg1 Corp headquarters in arg2
arg1 of arg2 → too general
arg2 is headquarters for chipmaker arg1 → too specific

Assessment (M45):
Classify candidate instances with a Naive Bayes classifier
Features related to strength of co-occurrence with each pattern

Promote top ranked instances and patterns. Use type-checking. Score patterns with estimate of precision

learned extraction patterns: Company

retailers_like__ such_clients_as__ an_operating_business_of__ being_acquired_by__
firms_such_as__ a_flight_attendant_for__ chains_such_as__ industry_leaders_such_as__
advertisers_like__ social_networking_sites_such_as__ a_senior_manager_at__
competitors_like__ stores_like__ __is_an_ebay_company discounters_like__
a_distribution_deal_with__ popular_sites_like__ a_company_such_as__ vendors_such_as__
rivals_such_as__ competitors_such_as__ has_been_quoted_in_the__ providers_such_as__
company_research_for__ providers_like__ giants_such_as__ a_social_network_like__
popular_websites_like__ multinationals_like__ social_networks_such_as__
the_former_ceo_of__ a_software_engineer_at__ a_store_like__ video_sites_like__
a_social_networking_site_like__ giants_like__ a_company_like__ premieres_on__
corporations_such_as__ corporations_like__ professional_profile_on__ outlets_like__
the_executives_at__ stores_such_as__ __is_the_only_carrier a_big_company_like__
social_media_sites_such_as__ __has_an_article_today manufacturers_such_as__
companies_like__ social_media_sites_like__ companies___including__ firms_like__
networking_websites_such_as__ networks_like__ carriers_like__
social_networking_websites_like__ an_executive_at__ insured_via__
__provides_dialup_access a_patent_infringement_lawsuit_against__
social_networking_sites_like__ social_network_sites_like__ carriers_such_as__
are_shipped_via__ social_sites_like__ a_licensing_deal_with__ portals_like__
vendors_like__ the_accounting_firm_of__ industry_leaders_like__ retailers_such_as__
chains_like__ prior_fiscal_years_for__ such_firms_as__ provided_free_by__
manufacturers_like__ airlines_like__ airlines_such_as__

learned extraction patterns: playsSport(arg1,arg2)

arg1_was_playing_arg2 arg2_megastar_arg1 arg2_icons_arg1 arg2_player_named_arg1
arg2_prodigy_arg1 arg1_is_the_tiger_woods_of_arg2 arg2_career_of_arg1
arg2_greats_as_arg1 arg1_plays_arg2 arg2_player_is_arg1 arg2_legends_arg1
arg1_announced_his_retirement_from_arg2 arg2_operations_chief_arg1
arg2_player_like_arg1 arg2_and_golfing_personalities_including_arg1 arg2_players_like_arg1
arg2_greats_like_arg1 arg2_players_are_steffi_graf_and_arg1 arg2_great_arg1
arg2_champ_arg1 arg2_greats_such_as_arg1 arg2_professionals_such_as_arg1
arg2_course_designed_by_arg1 arg2_hit_by_arg1 arg2_course_architects_including_arg1
arg2_greats_arg1 arg2_icon_arg1 arg2_stars_like_arg1 arg2_pros_like_arg1
arg1_retires_from_arg2 arg2_phenom_arg1 arg2_lesson_from_arg1
arg2_architects_robert_trent_jones_and_arg1 arg2_sensation_arg1 arg2_architects_like_arg1
arg2_pros_arg1 arg2_stars_venus_and_arg1 arg2_legends_arnold_palmer_and_arg1
arg2_hall_of_famer_arg1 arg2_racket_in_arg1 arg2_superstar_arg1 arg2_legend_arg1
arg2_legends_such_as_arg1 arg2_players_is_arg1 arg2_pro_arg1 arg2_player_was_arg1
arg2_god_arg1 arg2_idol_arg1 arg1_was_born_to_play_arg2 arg2_star_arg1
arg2_hero_arg1 arg2_course_architect_arg1 arg2_players_are_arg1
arg1_retired_from_professional_arg2 arg2_legends_as_arg1 arg2_autographed_by_arg1
arg2_related_quotations_spoken_by_arg1 arg2_courses_were_designed_by_arg1
arg2_player_since_arg1 arg2_match_between_arg1 arg2_course_was_designed_by_arg1
arg1_has_retired_from_arg2 arg2_player_arg1 arg1_can_hit_a_arg2
arg2_legends_including_arg1 arg2_player_than_arg1 arg2_legends_like_arg1
arg2_courses_designed_by_legends_arg1 arg2_player_of_all_time_is_arg1
arg2_fan_knows_arg1 arg1_learned_to_play_arg2 arg1_is_the_best_player_in_arg2
arg2_signed_by_arg1 arg2_champion_arg1

Experimental Evaluation

- 31 predicates
  - 15 relations, 16 categories
- Domains:
  - Companies
  - Sports
- Run for 15 iterations:
  - Full system
  - No Sharing of promoted items
  - No Relation/Category coupling
- Evaluated a sample of promoted items

Predicate	5 iterations			10 iterations			15 iterations		
	Full	NS	NCR	Full	NS	NCR	Full	NS	NCR
Actor	93	100	100	93	97	100	100	97	100
Athlete	100	100	100	100	93	100	100	73	100
Board Game	93	76	93	89	27	93	89	30	93
City	100	100	100	100	97	100	100	100	100
Coach	100	63	73	97	53	43	97	47	47
Company	100	100	100	97	90	97	100	90	100
Country	60	40	60	30	43	27	40	23	40
Economic Sector	77	63	73	57	67	67	50	63	40
Hobby	67	63	67	40	40	57	20	23	30
Person	97	97	90	97	93	97	93	97	93
Politician	93	93	97	73	53	90	90	53	87
Product	97	87	90	90	87	100	97	90	77
Product Type	93	93	90	70	73	97	77	80	67
Scientist	100	90	97	97	63	97	93	60	100
Sport	100	90	100	93	67	83	97	27	90
Sports Team	100	97	100	97	70	100	90	50	100
Category Average	92	84	89	82	70	84	83	63	79
Acquired(Company, Company)	77	77	80	67	80	47	70	63	47
CeoOf(Person, Company)	97	87	100	90	87	97	90	80	83
CoachesTeam(Coach, Sports Team)	100	100	100	100	100	97	100	100	90
CompetesIn(Company, Econ. Sector)	97	97	80	100	93	67	97	63	60
CompetesWith(Company, Company)	93	80	60	77	70	37	70	60	43
HasOfficesIn(Company, City)	97	93	40	93	90	27	93	57	30
HasOperationsIn(Company, Country)	100	95	50	100	97	40	90	83	13
HeadquarteredIn(Company, City)	77	90	20	70	77	27	70	60	7
LocatedIn(City, Country)	90	67	57	63	50	43	73	50	30
PlaysFor(Athlete, Sports Team)	100	100	0	100	97	7	100	43	0
PlaysSport(Athlete, Sport)	100	100	27	93	80	10	100	40	30
TeamPlaysSport(Sports Team, Sport)	100	100	77	100	97	80	93	83	67
Produces(Company, Product)	91	83	90	83	93	67	93	80	57
HasType(Product, Product Type)	73	63	17	33	67	33	40	57	27
Relation Average	92	88	57	84	84	48	84	66	42
All	92	86	74	83	76	68	84	64	62

Table 1: Precision (%) for each predicate. Results are presented after 5, 10, and 15 iterations, for the Full, No Sharing (NS), and No Category/Relation Coupling (NCR) configurations of CBL .

Extending Freebase

Category	Freebase Matches	CBL Instances	Est. Prec.	Est. New Instances
Actor	465	522	100	57
Athlete	54	117	100	63
Board Game	6	18	89	10
City	1665	1799	100	134
Company	995	1937	100	942
Econ. Sector	137	1541	50	634
Politician	74	962	90	792
Product	0	1259	97	1221
Sports Team	139	414	90	234
Sport	134	613	97	461

Table 3: Estimated numbers of 'New Instances', which are correct instances promoted by CBL in the Full 15 iteration run which do not have a match in Freebase, and the values used in calculating them (number of Freebase/CBL matches, number of CBL instances, and the estimated precision of CBL for the predicate).

Summary

For never-ending language learning, the key is achieving accurate semi-supervised training

→ Constrain learning by coupling the training of many types of knowledge (functions)

– sample complexity decreases as ontology size increases

→ Want an architecture in which current learning makes future learning even more accurate

-- learn symbolic rules which become new probabilistic constraints

→ Want architecture where self-consistency ≈ correctness