Machine Learning 10-601

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

October 18, 2017

Today:

- Graphical models
- Learning
  - Learning graph structure
  - Mixture models, clustering

Readings:

- Bishop chapter 8, 9-9.2
- Murphy chapter 11.4

Midterm Exam: Oct 25

- Pittsburgh: 6:30-9:00pm
- SV: 5:30-8:30pm

- No electronic anything allowed
- Bring one sheet of self-written notes (one side only)
- Will be graded on a curve
- Some easy questions, some hard

Potential new course

Conversational Machine Learning

***Natural language** as an interface to **machine learning** systems*

Machine Learning today is largely about finding patterns in large amounts of data.

However, as personal devices that interact with us in natural language become ubiquitous (e.g., Siri, Google Now), they open the possibility of ***letting users teach machines in natural language***, similar to how we teach each other.

- **Project-based course**
- **New datasets**
- **Open, research level projects**

Example challenges

1. Building a classifier from zero examples

2. Telling sequence to sequence models about their mistakes in natural language

3. Letting machine learning models ask questions and explain themselves via language

Learning of Bayes Nets

- Four categories of learning problems
  - Graph structure may be known/unknown
  - Variable values may be fully observed / partly unobserved

- Easy case: learn parameters when graph structure is *known*, and training data is *fully observed*

- Interesting case: graph *known*, data *partly observed*

- Gruesome case: graph structure *unknown*, data *partly unobserved*

Learning Bayes Net Structure

How can we learn Bayes Net graph structure?

In general case, open problem

- can require lots of data (else high risk of overfitting)
- can use Bayesian priors, or other kinds of prior assumptions about graph structure to constrain search

One key result:

- Chow-Liu algorithm: finds "best" tree-structured network
- What's best?
  - suppose P(**X**) is true distribution, T(**X**) is our tree-structured network, where **X** = <X_1, … X_n>
  - Chow-Liu minimizes Kullback-Leibler divergence:

$$KL(P(\mathbf{X}) \,||\, T(\mathbf{X})) \equiv \sum_k P(\mathbf{X} = k) \log \frac{P(\mathbf{X} = k)}{T(\mathbf{X} = k)}$$

Kullback-Leibler Divergence

- KL(P(X) || T(X)) is a measure of the difference between distribution P(X) and T(X)

$$KL(P(\mathbf{X}) \,||\, T(\mathbf{X})) \equiv \sum_k P(\mathbf{X}=k) \log \frac{P(\mathbf{X}=k)}{T(\mathbf{X}=k)}$$

- It is asymetric, always greater or equal to 0
- It is 0 iff P(X)=T(X)

$$KL(P(X)||T(X)) = \sum_k P(X=k) \log P(X=k) - \sum_k P(X=k) \log T(X=k)$$

$$= -H(P) + H(P,T)$$

where cross entropy $H(P,T) = \sum_k -P(X=k) \log T(X=k)$

Chow-Liu Algorithm

Key result: To minimize KL(P || T) over possible tree networks T approximating true P, it suffices to find the tree network T that maximizes the sum of mutual informations over its edges

Mutual information for an edge between variable A and B:

$$I(A,B) = \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

This works because for tree networks with nodes $\mathbf{X} \equiv \langle X_1 \ldots X_n \rangle$

$$\begin{aligned} KL(P(\mathbf{X}) \parallel T(\mathbf{X})) &\equiv \sum_k P(\mathbf{X}=k) \log \frac{P(\mathbf{X}=k)}{T(\mathbf{X}=k)} \\ &= -\sum_i I(X_i, Pa(X_i)) + \sum_i H(X_i) - H(X_1 \ldots X_n) \end{aligned}$$

Chow-Liu Algorithm

1. for each pair of variables A,B, use data to estimate P(A,B), P(A), and P(B)

2. for each pair A, B calculate mutual information

$$I(A,B) = \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

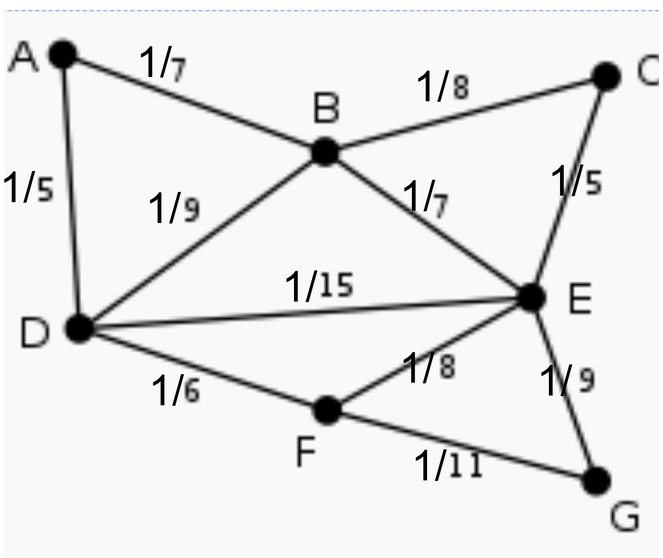
3. calculate the maximum spanning tree over the set of variables, using edge weights $I(A,B)$

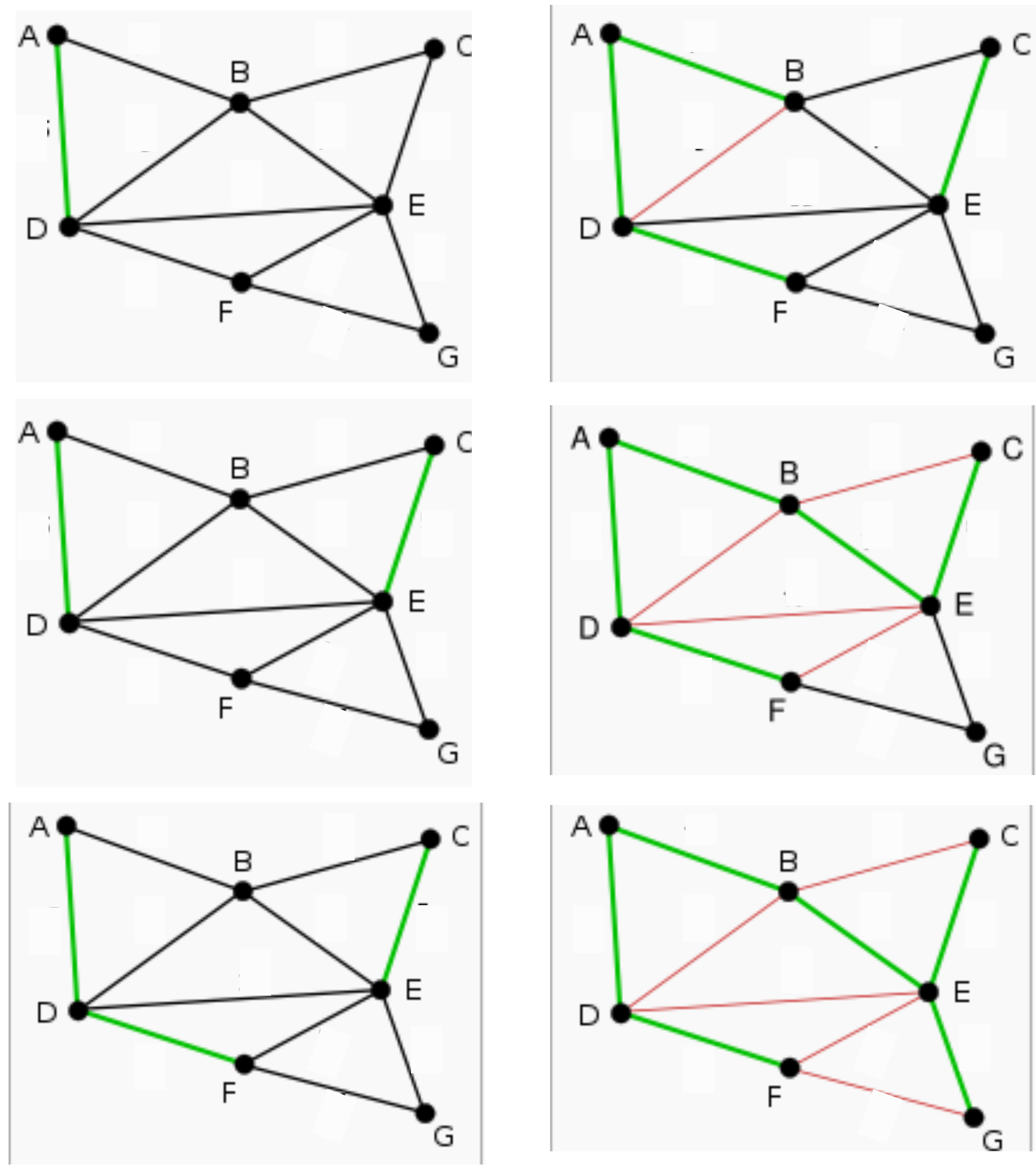
   (given N vars, this costs only $O(N^2)$ time)

4. add arrows to edges to form a directed-acyclic graph

5. learn the CPD's for this graph 

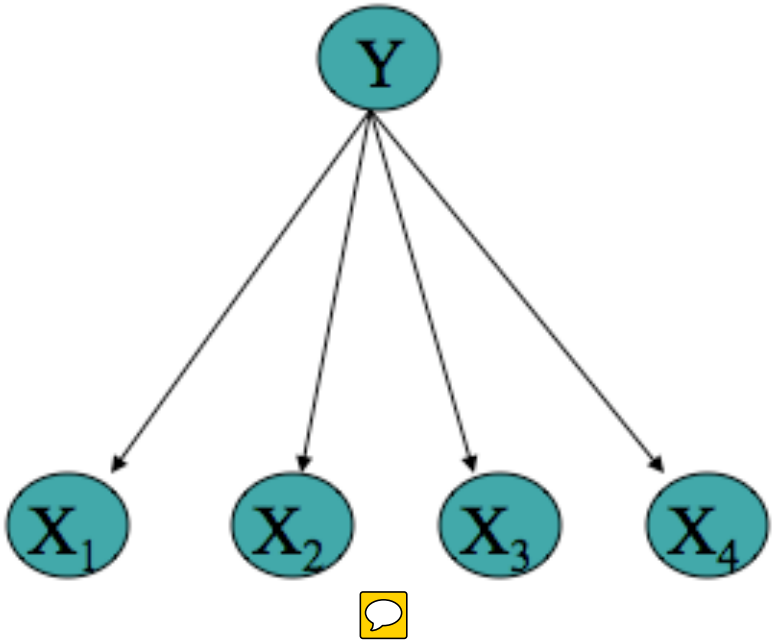
Chow-Liu algorithm example
Greedy Algorithm to find Max-Spanning Tree





[courtesy A. Singh, C. Guestrin]

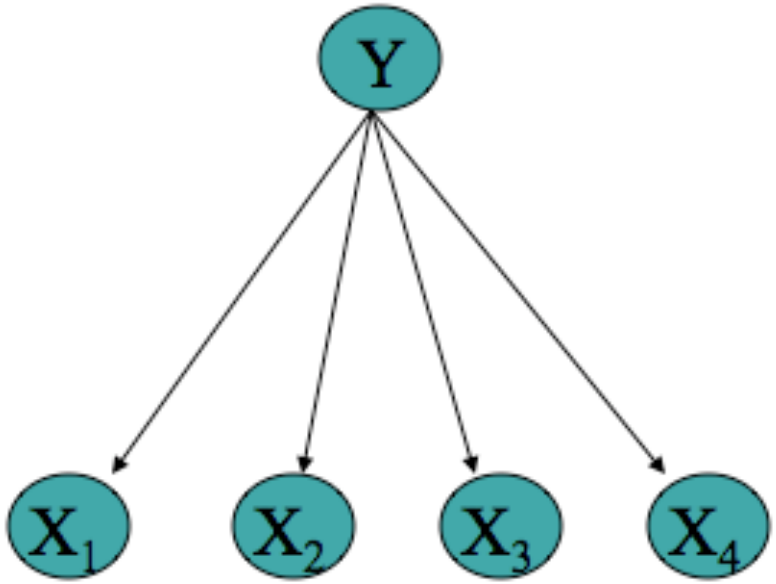
Can we use structure learning to modify Naïve Bayes cond. indep. assumptions?



[Nir Friedman et al., 1997]

Tree Augmented Naïve Bayes

[Nir Friedman et al., 1997]



EM and Unlabeled Data

EM Algorithm - Precisely

EM is a general procedure for learning from partly observed data

Given observed variables X, unobserved Z (X={F,A,H,N}, Z={S})

Define $Q(\theta'|\theta) = E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

current

M step new

Iterate until convergence:

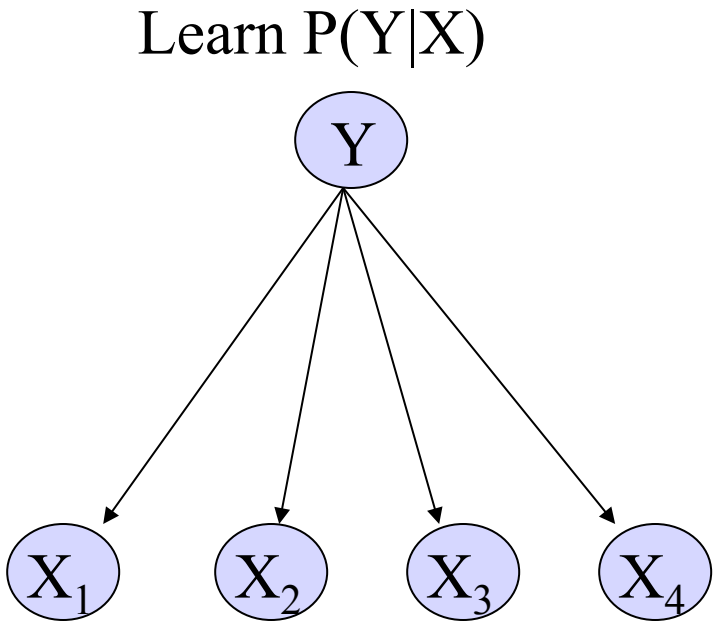
- E Step: Use X and current θ to calculate P(Z|X,θ)
- M Step: Replace current θ by

$$\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$$

Guaranteed to find local maximum.

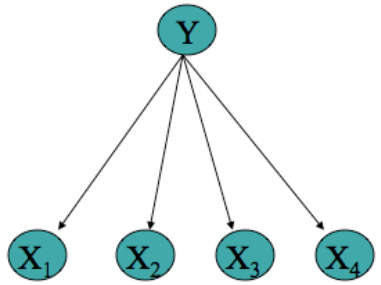
Each iteration increases $E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

Using Unlabeled Data to Help Train Naïve Bayes Classifier



Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
?	0	1	1	0
?	0	1	0	1

EM and estimating θ



Given observed set X, unobserved set Y of boolean values

E step: Calculate for each training example, k

the expected value of each unobserved variable Y

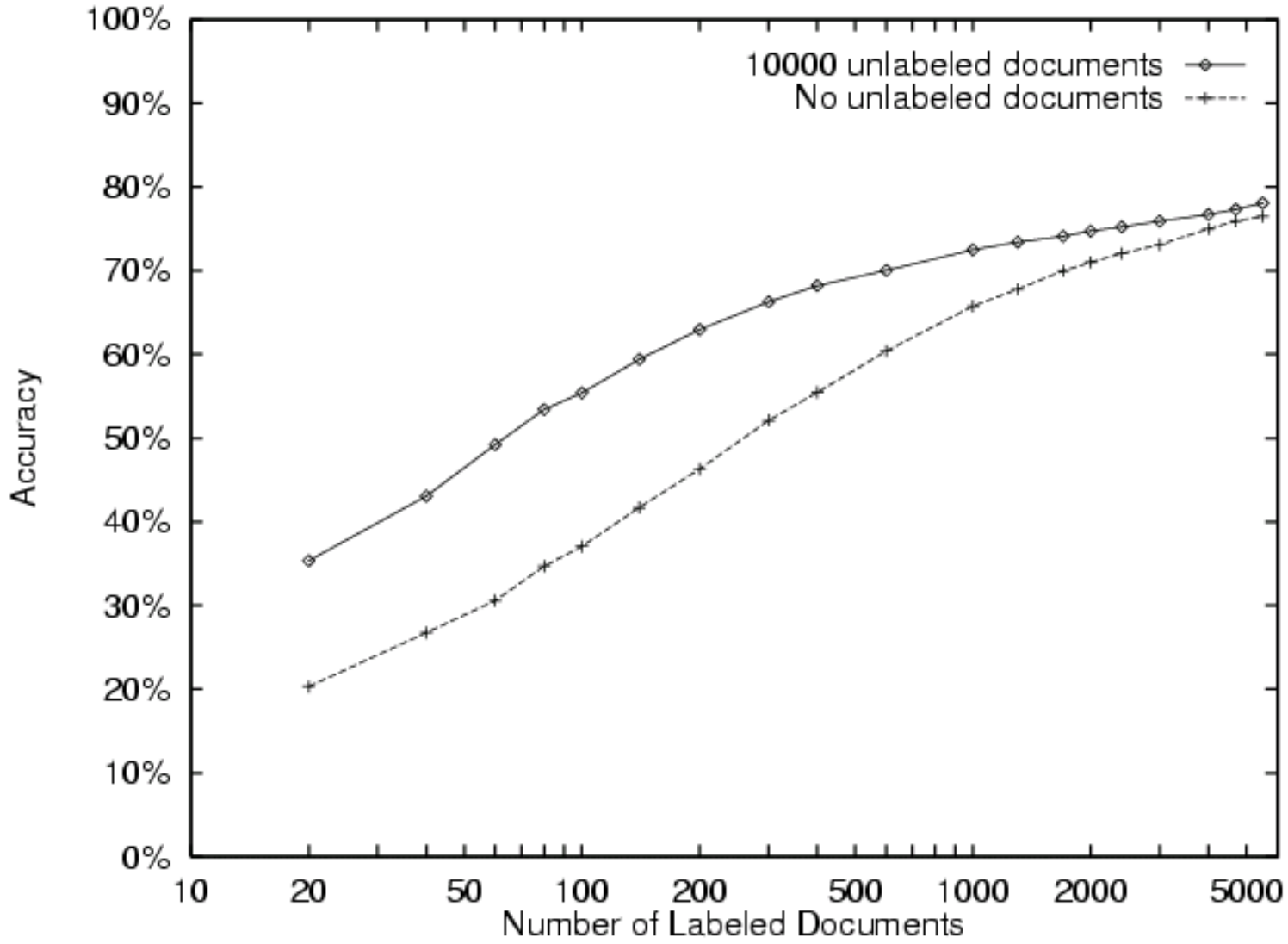
$$E_{P(Y|X_1 \ldots X_N)}[y(k)] = P(y(k)=1|x_1(k), \ldots x_N(k); \theta) = \frac{P(y(k)=1)\prod_i P(x_i(k)|y(k)=1)}{\sum_{j=0}^{1} P(y(k)=j)\prod_i P(x_i(k)|y(k)=j)}$$

M step: Calculate estimates similar to MLE, but replacing each count by its expected count

$$\theta_{ij|m} = \hat{P}(X_i = j|Y = m) = \frac{\sum_k P(y(k)=m|x_1(k) \ldots x_N(k))\ \delta(x_i(k)=j)}{\sum_k P(y(k)=m|x_1(k) \ldots x_N(k))}$$

MLE would be: $\hat{P}(X_i = j|Y = m) = \frac{\sum_k \delta((y(k)=m) \wedge (x_i(k)=j))}{\sum_k \delta(y(k)=m)}$

20 Newsgroups



What you should know about EM

- For learning from partly unobserved data
- MLE of θ = $\arg\max_\theta \log P(data|\theta)$
- EM estimate: θ = $\arg\max_\theta E_{Z|X,\theta}[\log P(X, Z|\theta)]$

  Where X is observed part of data, Z is unobserved

- EM for training Bayes networks
- Can also develop MAP version of EM
- Can also derive your own EM algorithm for your own problem
  - write out expression for $E_{Z|X,\theta}[\log P(X, Z|\theta)]$
  - E step: for each training example X^k, calculate $P(Z^k \mid X^k, \theta)$
  - M step: chose new θ to maximize $E_{Z|X,\theta}[\log P(X, Z|\theta)]$

Usupervised clustering

Just extreme case for EM with zero labeled examples…

Clustering

- Given set of data points, group them
- Unsupervised learning
- Which patients are similar? (or which earthquakes, customers, faces, web pages, …)

Mixture Distributions

Model joint $P(X_1 \ldots X_n)$ as mixture of multiple distributions.

Use discrete-valued random var Z to indicate which distribution is being use for each random draw

So

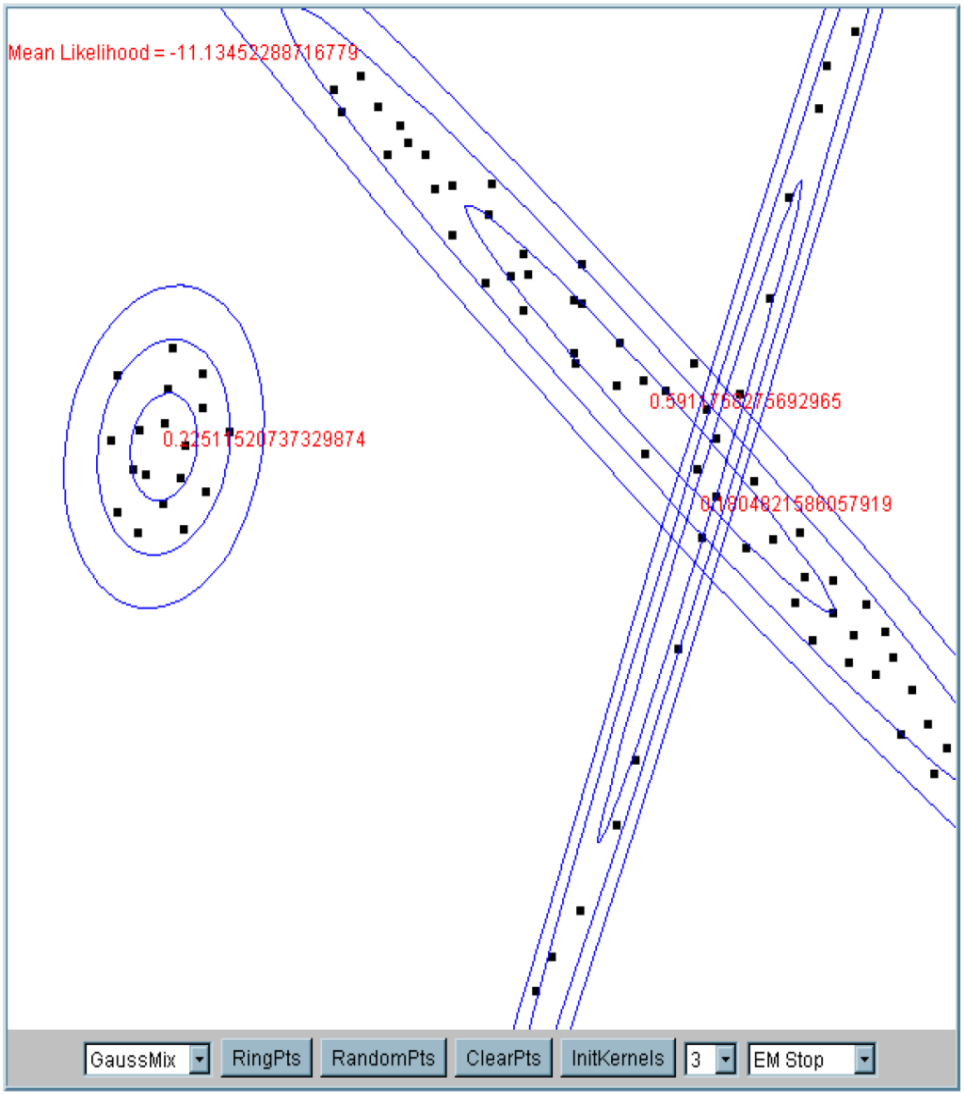
$$P(X_1 \ldots X_n) = \sum_i P(Z = i) \;\; P(X_1 \ldots X_n | Z)$$

Mixture of *Gaussians*:

- Assume each data point X=<X1, … Xn> is generated by one of several Gaussians, as follows:

1. randomly choose Gaussian i, according to P(Z=i)
2. randomly generate a data point <x1,x2 .. xn> according to $N(\mu_i, \Sigma_i)$

Mixture of Gaussians



EM for Mixture of Gaussian Clustering

Let's simplify to make this easier:

1. assume $X=<X_1 \ldots X_n>$, and the X_i are conditionally independent given Z.

$$P(X|Z=j) = \prod_i N(X_i|\mu_{ji}, \sigma_{ji})$$

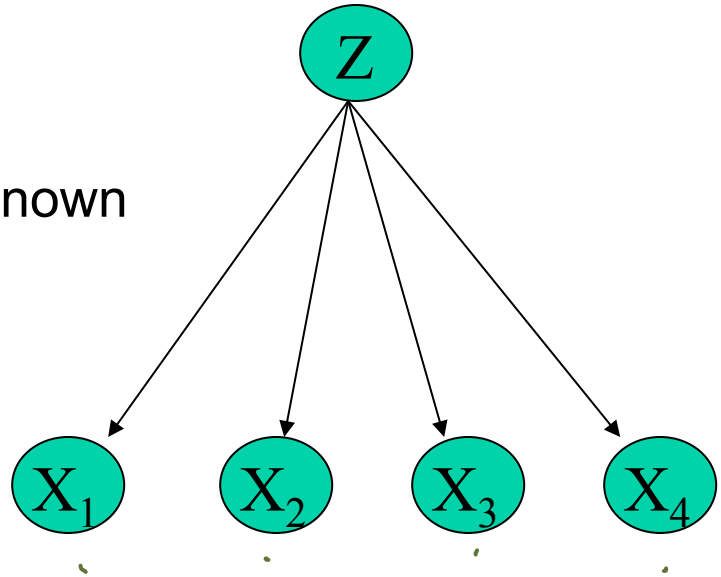
2. assume only 2 clusters (values of Z), and $\forall i, j, \sigma_{ji} = \sigma$

$$P(\mathbf{X}) = \sum_{j=1}^{2} P(Z=j|\pi) \prod_i N(x_i|\mu_{ji}, \sigma)$$

3. Assume σ known, $\pi_1 \ldots \pi_K, \mu_{1i} \ldots \mu_{Ki}$ unknown

Observed: $X=<X_1 \ldots X_n>$

Unobserved: Z

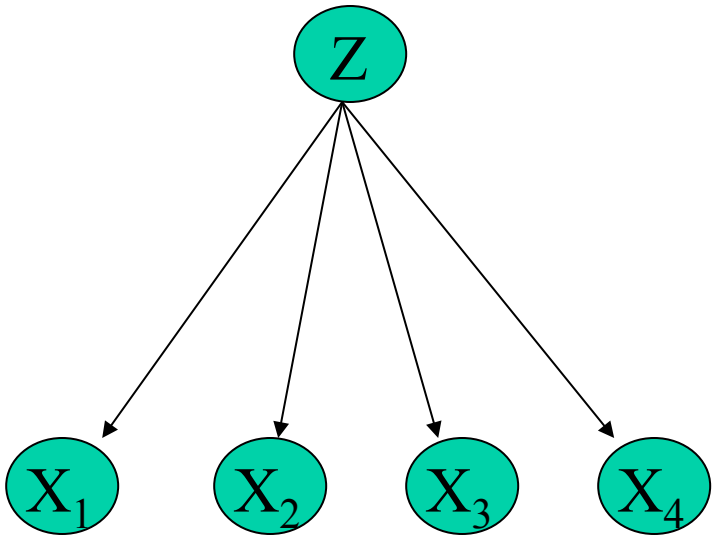


EM

Given observed variables X, unobserved Z

Define $Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X, Z|\theta')]$

where $\theta = \langle \pi, \mu_{ji} \rangle$



Iterate until convergence:

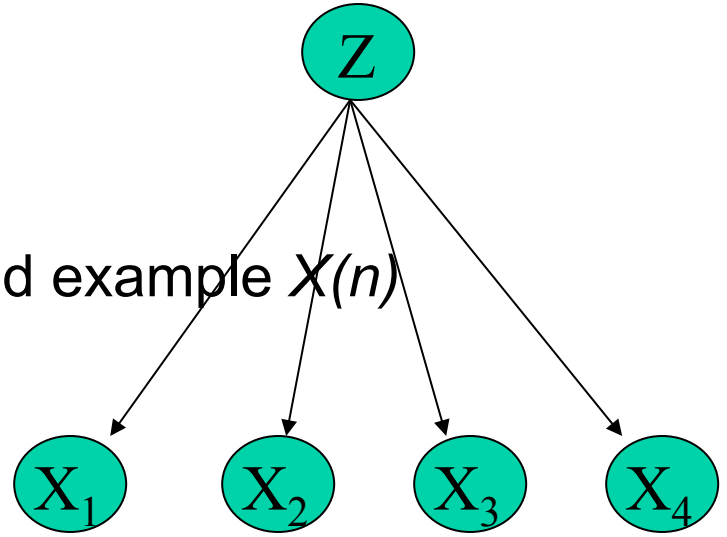
- E Step: Calculate $P(Z(n)|X(n), \theta)$ for each example $X(n)$. Use this to construct $Q(\theta'|\theta)$
- M Step: Replace current θ by

$$\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$$

EM – E Step

Calculate $P(Z(n)|X(n),\theta)$ for each observed example $X(n)$

$X(n)=<x_1(n), x_2(n), \ldots x_T(n)>$.



$$P(z(n)=k|x(n),\theta)=\frac{P(x(n)|z(n)=k,\theta)\quad P(z(n)=k|\theta)}{\sum_{j=0}^{1}p(x(n)|z(n)=j,\theta)\;\;P(z(n)=j|\theta)}$$

$$P(z(n)=k|x(n),\theta)=\frac{\prod_i P(x_i(n)|z(n)=k,\theta)]\quad P(z(n)=k|\theta)}{\sum_{j=0}^{1}\prod_i P(x_i(n)|z(n)=j,\theta)\;\;P(z(n)=j|\theta)}$$

$$P(z(n)=k|x(n),\theta)=\frac{\prod_i N(x_i(n)|\mu_{k,i},\sigma)]\quad (\pi^k(1-\pi)^{(1-k)})}{\sum_{j=0}^{1}[\prod_i N(x_i(n)|\mu_{j,i},\sigma)]\quad (\pi^j(1-\pi)^{(1-j)})}$$

EM – M Step

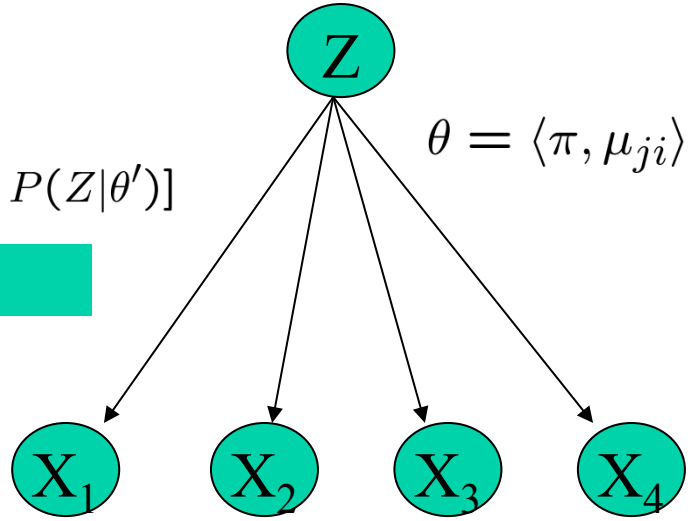
First consider update for π

$$Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X, Z|\theta')] = E[\log P(X|Z, \theta') + \log P(Z|\theta')]$$

π' has no influence

$$\pi \leftarrow \arg\max_{\pi'} E_{Z|X,\theta}[\log P(Z|\pi')]$$

z=1 for nth example



$$E_{Z|X,\theta}\left[\log P(Z|\pi')\right] = E_{Z|X,\theta}\left[\log\left(\pi'^{\sum_n z(n)}(1-\pi')^{\sum_n (1-z(n))}\right)\right]$$

$$= E_{Z|X,\theta}\left[\left(\sum_n z(n)\right)\log\pi' + \left(\sum_n (1-z(n))\right)\log(1-\pi')\right]$$

$$= \left(\sum_n E_{Z|X,\theta}[z(n)]\right)\log\pi' + \left(\sum_n E_{Z|X,\theta}[(1-z(n)])\right)\log(1-\pi')$$

$$\frac{\partial E_{Z|X,\theta}[\log P(Z|\pi')]}{\partial \pi'} = \left(\sum_n E_{Z|X,\theta}[z(n)]\right)\frac{1}{\pi'} + \left(\sum_n E_{Z|X,\theta}[(1-z(n)])\right)\frac{(-1)}{1-\pi'}$$

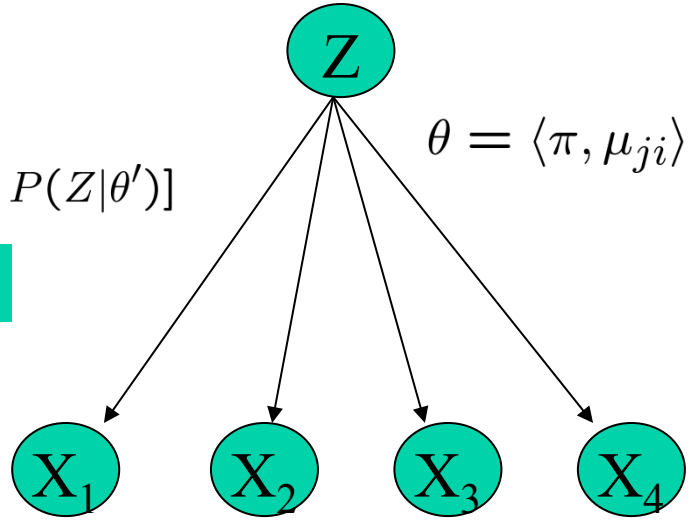
$$\pi \leftarrow \frac{\sum_{n=1}^N E[z(n)]}{\left(\sum_{n=1}^N E[z(n)]\right) + \left(\sum_{n=1}^N (1-E[z(n)])\right)} = \frac{1}{N}\sum_{n=1}^N E[z(n)]$$

EM – M Step

Now consider update for μ_{ji}

$$Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X, Z|\theta')] = E[\log P(X|Z, \theta') + \log P(Z|\theta')]$$

μ_{ji}' has no influence



$$\mu_{ji} \leftarrow \arg\max_{\mu'_{ji}} E_{Z|X,\theta}[\log P(X|Z, \theta')]$$

…
…
…

$$\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta) \quad x_i(n)}{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta)}$$

Compare above to MLE if Z were observable:

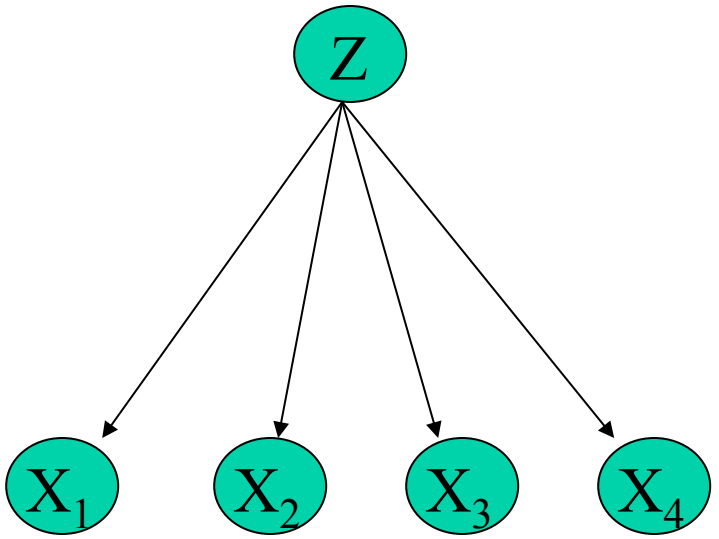
$$\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} \delta(z(n) = j) \quad x_i(n)}{\sum_{n=1}^{N} \delta(z(n) = j)}$$

EM – putting it together

Given observed variables X, unobserved Z

Define $Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X, Z|\theta')]$

where $\theta = \langle \pi, \mu_{ji} \rangle$



Iterate until convergence:

- E Step: For each observed example X(n), calculate $P(Z(n)|X(n), \theta)$

$$P(z(n) = k \mid x(n), \theta) = \frac{[\prod_i N(x_i(n)|\mu_{k,i}, \sigma)] \quad (\pi^k (1-\pi)^{(1-k)})}{\sum_{j=0}^{1} [\prod_i N(x_i(n)|\mu_{j,i}, \sigma)] \quad (\pi^j (1-\pi)^{(1-j)})}$$

- M Step: Update $\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$

$$\pi \leftarrow \frac{1}{N} \sum_{n=1}^{N} E[z(n)]$$

$$\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta) \quad x_i(n)}{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta)}$$

Mixture of Gaussians applet

Go to:

http://www.socr.ucla.edu/htmls/SOCR_Charts.html

then go to Go to “Line Charts” → SOCR EM Mixture Chart

- try it with 2 Gaussian mixture components (“kernels”)
- try it with 4

What you should know about EM

- For learning from partly unobserved data
- MLE of θ = $\arg\max_\theta \log P(data|\theta)$
- EM estimate: θ = $\arg\max_\theta E_{Z|X,\theta}[\log P(X, Z|\theta)]$

  Where X is observed part of data, Z is unobserved
- Nice case is Bayes net of boolean vars:
  - M step is like MLE, with with unobserved values replaced by their expected values, given the other observed values
- EM for training Bayes networks
- Can also develop MAP version of EM
- Can also derive your own EM algorithm for your own problem
  - write out expression for $E_{Z|X,\theta}[\log P(X, Z|\theta)]$
  - E step: for each training example X^k, calculate $P(Z^k \mid X^k, \theta)$
  - M step: chose new θ to maximize

Never Ending Language Learning

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

Thesis:

We will never really understand learning until we build machines that

- learn many different things,
- from years of diverse experience,
- in a staged, curricular fashion,
- and become better learners over time.

NELL: Never-Ending Language Learner

The task:

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

Inputs:

- initial ontology (categories and relations)
- dozen examples of each ontology predicate
- the web
- occasional interaction with human trainers

Research questions

How can we architect system so that acquiring one skill improves ability to learn others?

What parts of agent should be fixed, vs. plastic?

How to learn from mostly unsupervised training?

How to avoid “learning plateaus”?

What self-reflection and self-modification?

What theoretical guarantees?

NELL today

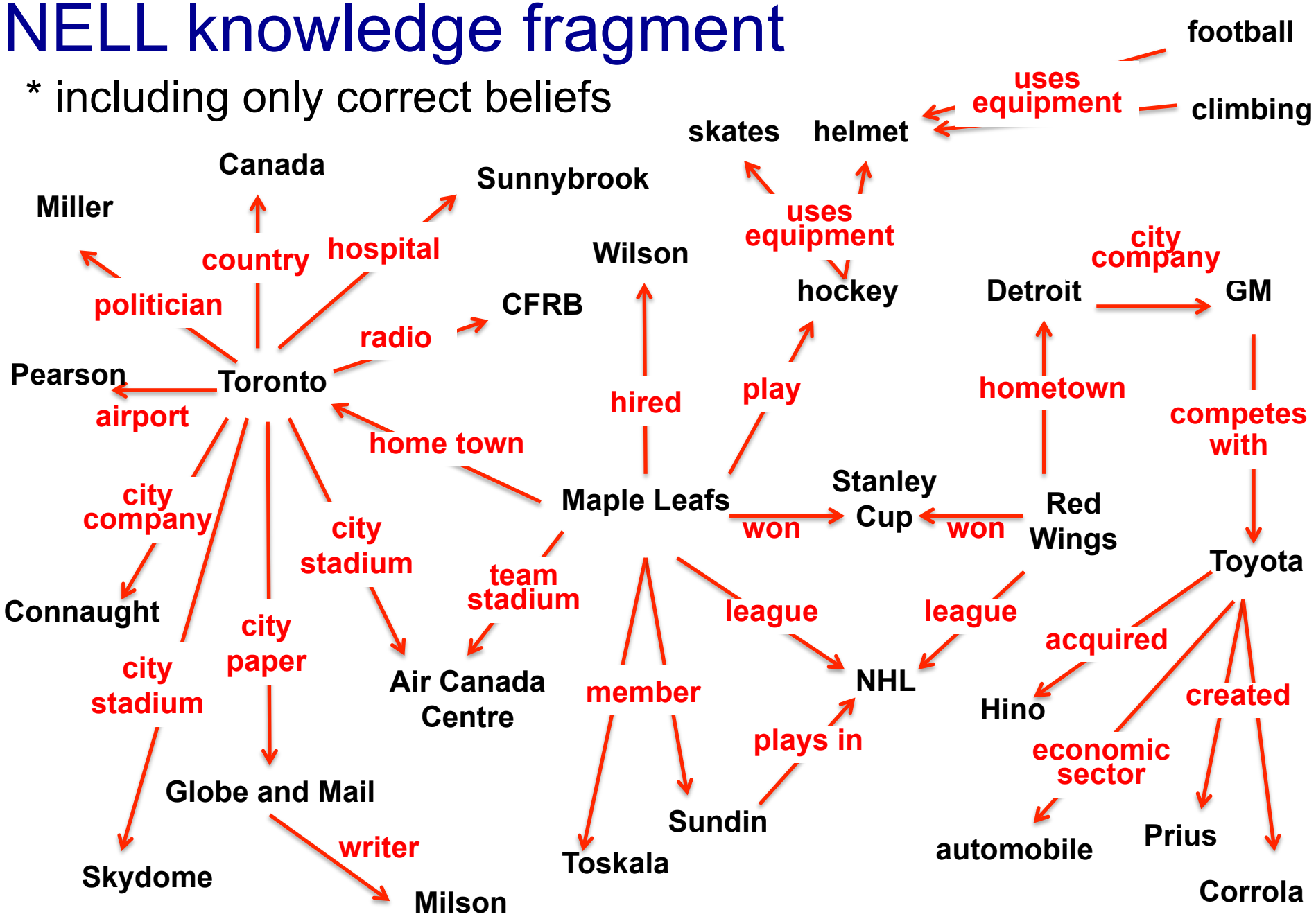
Running 24x7, since January, 12, 2010

Result:

- knowledge base with ~100 million confidence-weighted beliefs
- learning to read
- learning to reason
- extending ontology

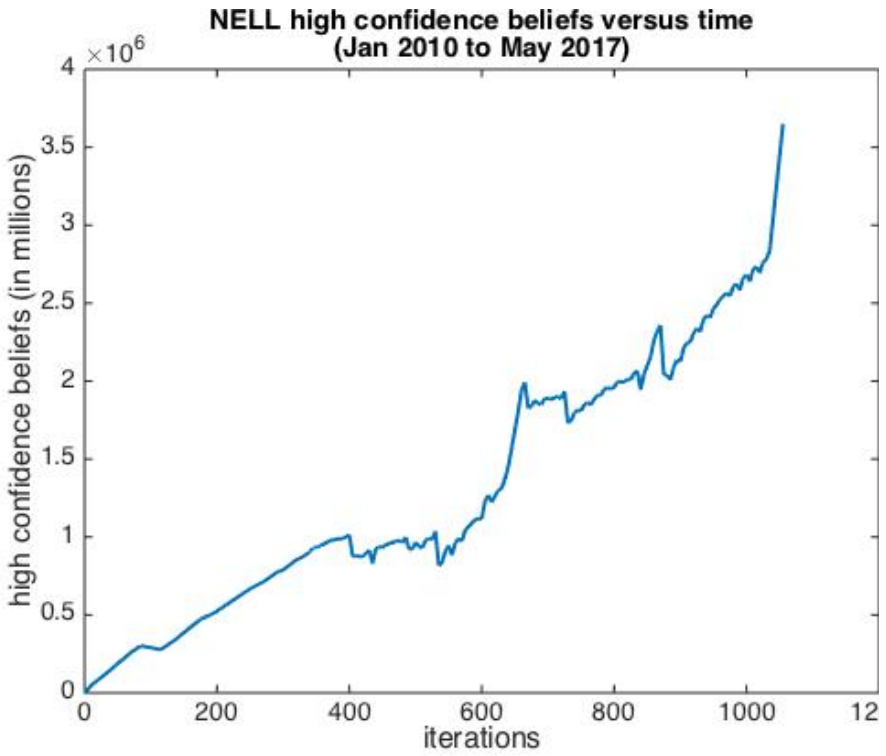
NELL knowledge fragment

* including only correct beliefs

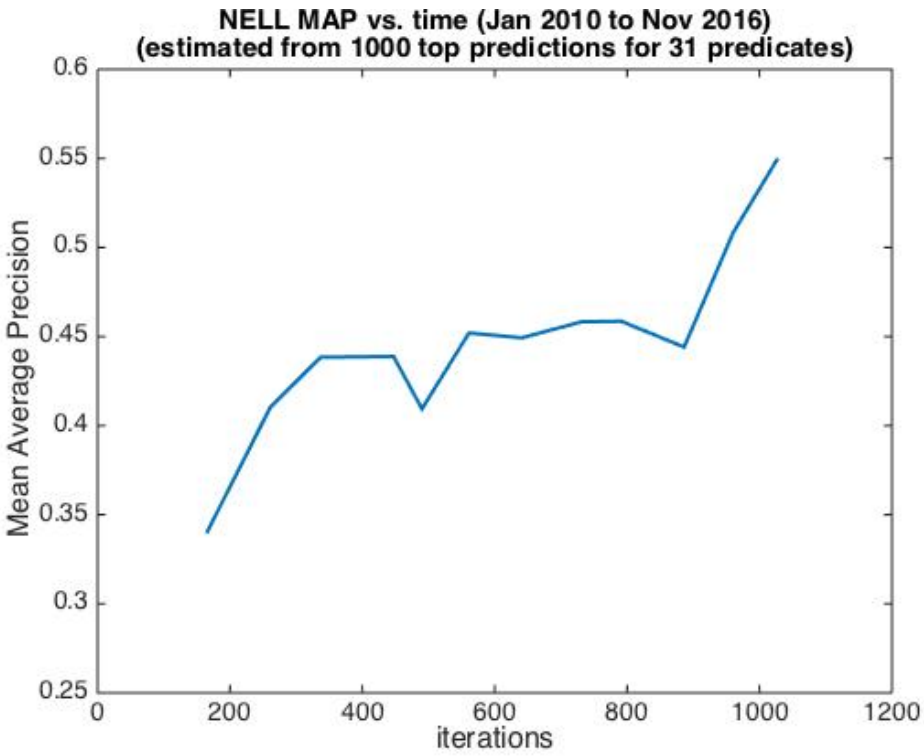


NELL Is Improving Over Time (Jan 2010 to now)

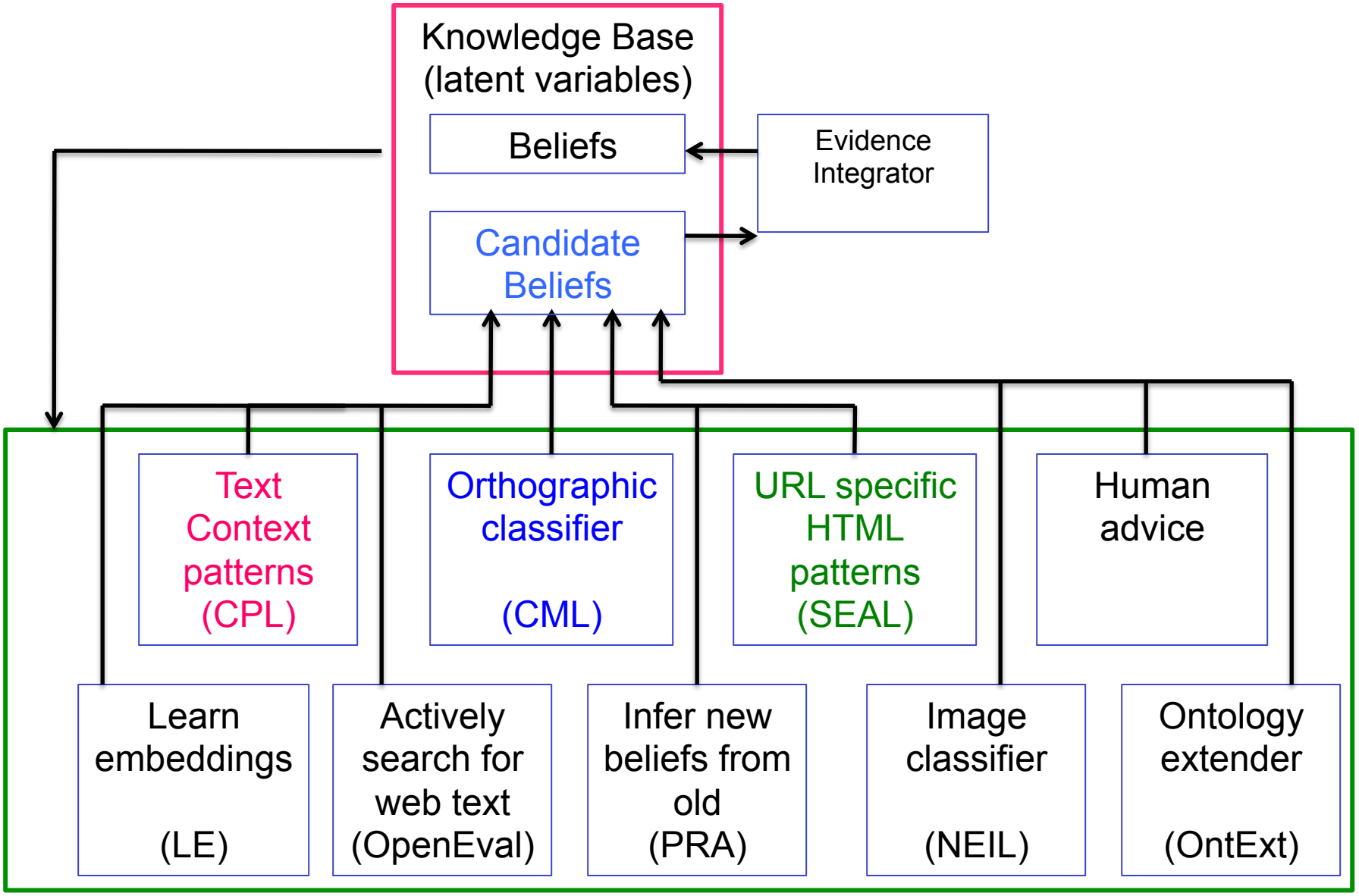
Quantity



Accuracy



NELL Architecture



NELL: Learned reading strategies

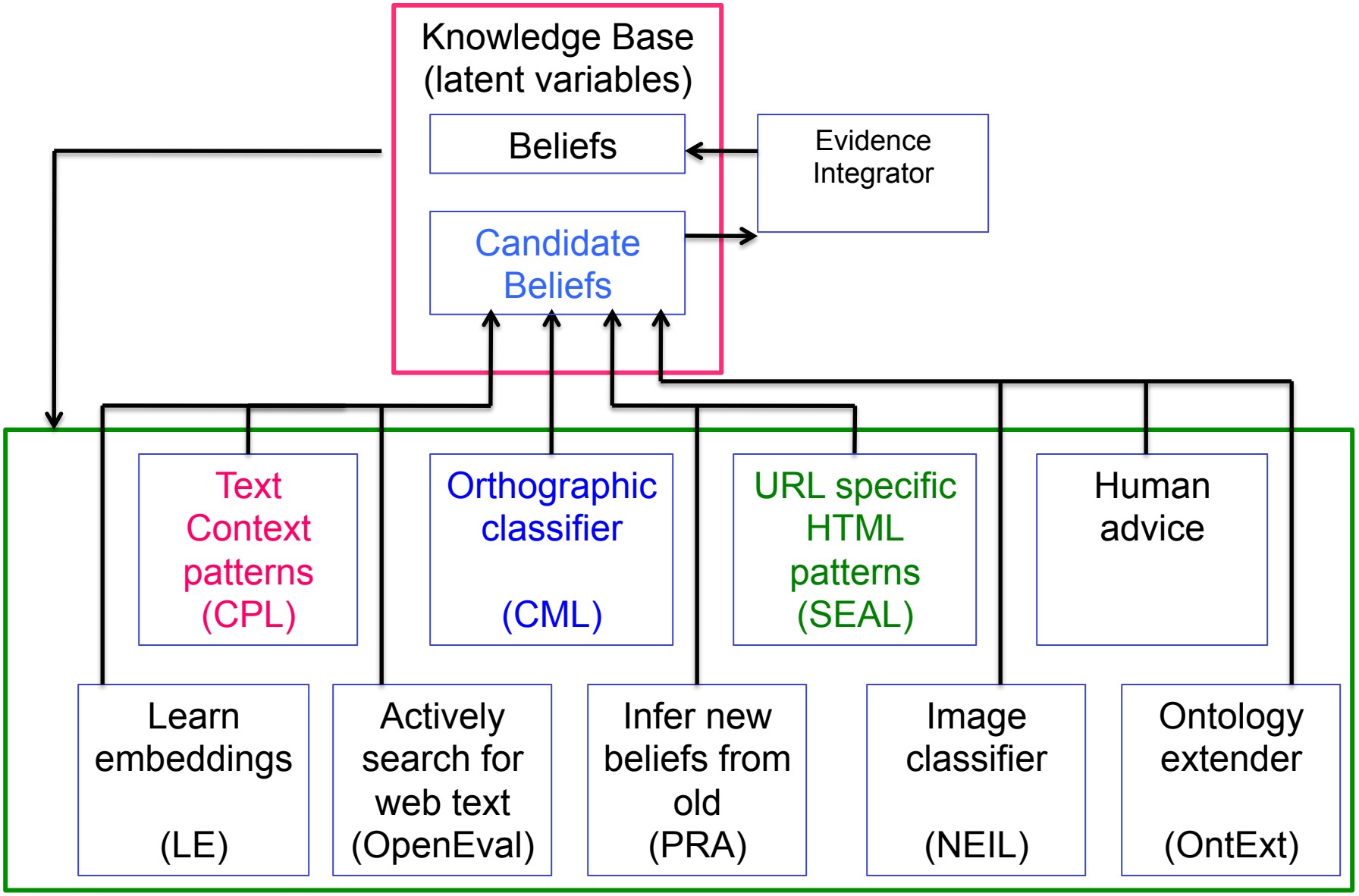
Mountain:

"volcanic crater of _" "We 've climbed
atop _" "week hiking in _" "_ 's Base
_" "west to beyond _" "white ledge in
area surrounding _" "wilderness area
of _" "winter ascents of _" "winter exp
"world famous view of _" "world famo
"you 've just climbed _" "you just clim
' eruption" "_ 's drug guide" "_ 's east
Face" "_ 's North Peak" "_ 's North R
southeast ridge" "_ 's summit caldera
's west ridge" "_ (D,DDD ft" " "_ climb
consult el diablo" "_ cooking planks"
the western skyline" "_ dominating th
"volcanic peak of _" "volcanic region
_" "volcano is called _" "volcano kno
named _" "volcanoes , including _" "
_" "volcanoes include _" "volcanoes

Predicate	Feature	Weight
mountain	LAST=peak	1.791
mountain	LAST=mountain	1.093
mountain	FIRST=mountain	-0.875
musicArtist	LAST=band	1.853
musicArtist	POS=DT_NNS	1.412
musicArtist	POS=DT_JJ_NN	-0.807
newspaper	LAST=sun	1.330
newspaper	LAST=university	-0.318
newspaper	POS=NN_NNS	-0.798
university	LAST=college	2.076
university	PREFIX=uc	1.999
university	LAST=state	1.992
university	LAST=university	1.745
university	FIRST=college	-1.381
visualArtMovement	SUFFIX=ism	1.282
visualArtMovement	PREFIX=journ	-0.234

Predicate	Web URL	Extraction Template
academicField	http://scholendow.ais.msu.edu/student/ScholSearch.Asp	` [X] -`
athlete	http://www.quotes-search.com/d_occupation.aspx?o=+athlete	`<a href='d_author.aspx?a=[X]'>-`
bird	http://www.michaelforsberg.com/stock.html	`<option>[X]</option>`
bookAuthor	http://lifebehindthecurve.com/	`</li> <li>[X] by [Y] &#8211;`

NELL Architecture



Cumulative, Staged Learning in NELL

Learning X improves ability to learn Y

1. Classify noun phrases (NP's) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances
4. Learn which NP's (co)refer to which latent concepts
5. Discover new relations to extend ontology
6. Learn to infer relation instances via targeted random walks
7. Learn to microread single sentences, paragraphs
8. Vision: connect NELL and NEIL
9. Learn in multiple languages
10. Goal-driven reading: predict, then read to corroborate/correct
11. Make NELL a conversational agent on Twitter
12. Add a robot body to NELL

NELL is here

Bayes Nets – What You Should Know

- Representation
  - Bayes nets represent joint distribution as a DAG + Conditional Distributions
  - D-separation lets us decode conditional independence assumptions
- Inference
  - NP-hard in general
  - For some graphs, closed form inference is feasible
  - Approximate methods too, e.g., Monte Carlo methods, …
- Learning
  - Easy for known graph, fully observed data (MLE's, MAP est.)
  - EM for partly observed data, known graph
  - Learning graph structure: Chow-Liu for tree-structured networks
  - Hardest when graph unknown, data incompletely observed