

On-Line Learning of Predictive Compositional Hierarchies

Karl Pfleger

Computer Science Department

Stanford University

`kpfleger@cs.stanford.edu`

`www-cs-students.stanford.edu/~kpfleger`

Abstract

- a *compositional hierarchy* (CH) is a part–whole hierarchy
- *predictive* CHs are sensitive to statistical properties of the environment and can, among other things, predict unseen data as a result
- this talk focuses on *learning* such structures
- the learning is *unsupervised*, *on–line* (seeing a little data at a time), *data–driven*, and *bottom–up*

- This type of learning is particularly important for creating agents with significant levels of autonomy, flexibility, and intelligence.

The Goal

scale automatically from low-level data to higher-level representations

- Agents need, and people benefit from, many high-level representations in their heads. High-level representations include things such as:
 - the word 'automatically'
 - the sequence of actions involved in tying shoelaces
 - the melody of Twinkle, Twinkle, Little Star
 - the layout of their homes
- Such high-level representations can be learned automatically through repeated exposure to the patterns in the environment.

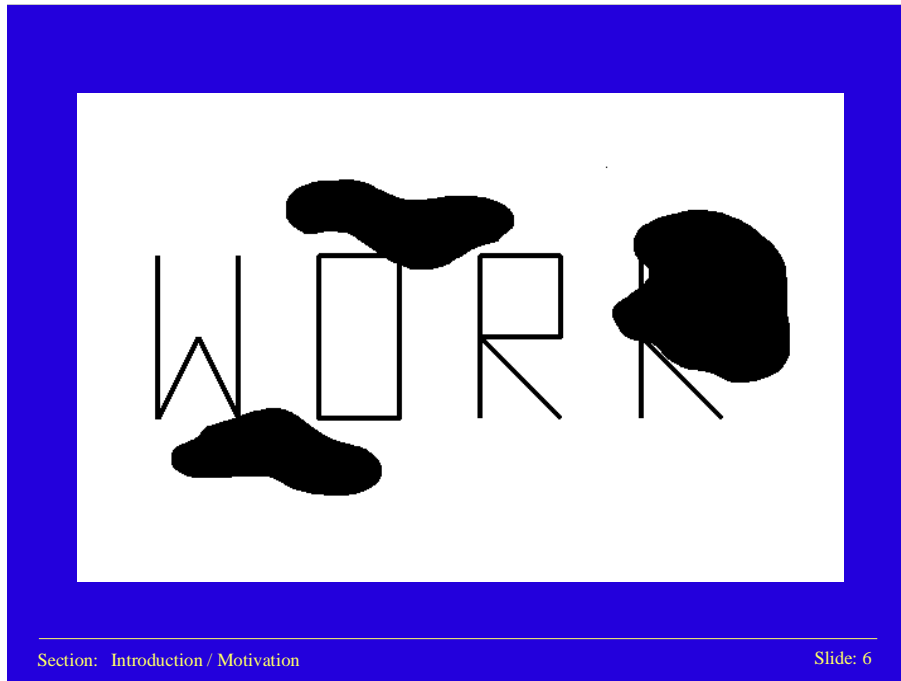
The Approach

identify repeated, frequent
patterns in data, enabling the
future (hierarchical) discovery
of even larger patterns

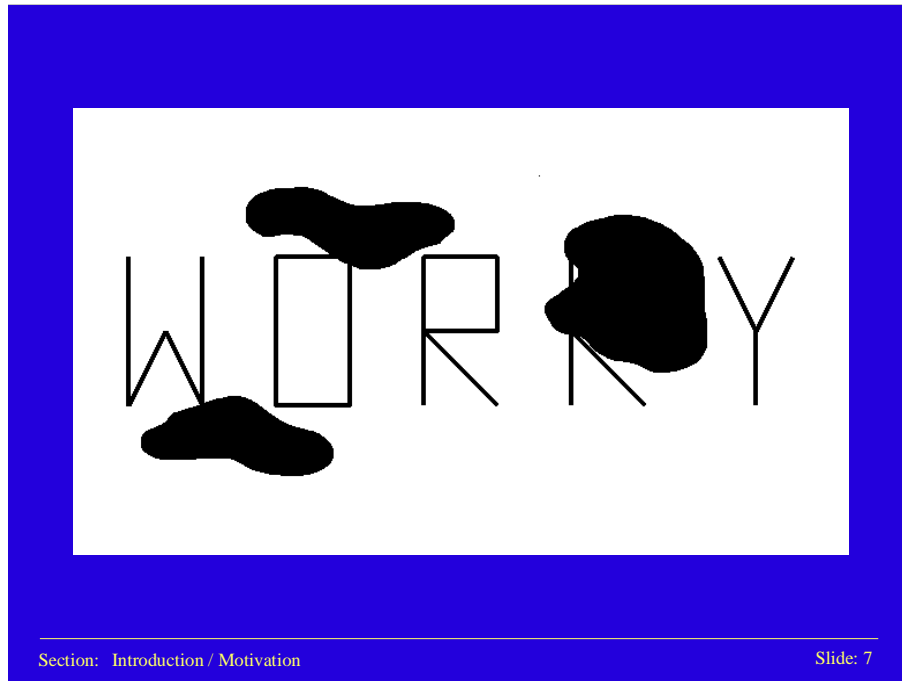
Outline

- Introduction
 - background, motivation, learning framework
- Compositional probabilistic networks
 - probabilistic neural network graphical models
- Hierarchical sparse n -grams
 - simple sparse counting models
- Conclusion
 - related work, the big picture, conclusion

- I will discuss two learning systems, briefly covering the first and covering the second in more detail.



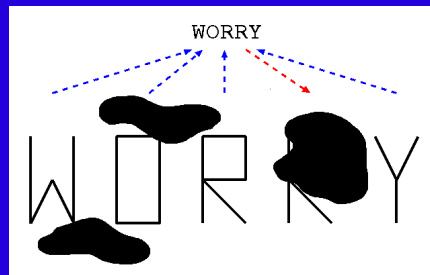
- What is the 4th letter?
- How do we know?



- What is the 2nd to last letter?
- In this case we used context on both sides.

Context and Knowledge

- context allows lateral inference (fill in the blank, resolve the ambiguity, predict the future/past)
- this requires knowledge of common patterns



Section: Introduction / Motivation

Slide: 8

- There are successful existing systems in both AI and psychology that can do this type of inference using prespecified compositional hierarchies, smoothly integrating the bottom-up and top-down influences. This talk discusses techniques for learning models capable of doing this at multiple levels of granularity simultaneously.

Learning Knowledge Units

- AI systems need many knowledge units (especially highly flexible, autonomous agents)
- the effectiveness of AI systems is too dependent on human specification of representational units (states, operators, features, rules, schemata, scripts)
- some knowledge units should come from observed regularities of the environment
- *learning is needed*

- Learning is needed for several reasons:
 - It is too large an endeavor to manually program sufficient knowledge for broad and deep competence.
 - Much of the important knowledge is not conscious or easily spelled out in a usable form.
 - We want to be able to create systems that can develop competence in environments where we do not have all the relevant knowledge to begin with.

Hierarchy in Learning

- key idea: substantial intelligence, flexibility, and autonomy requires that agents must learn *representations that can embody significant complexity*
- bootstrapped learning: an agent should learn and then use what it has learned to learn more—an idea often described but seldom operationalized

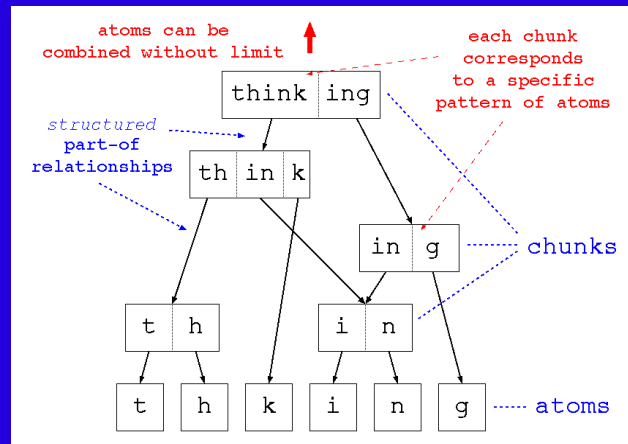
- Tuning a few parameters is not sufficient!

Types of Hierarchies

- two types of relationship are fundamental for KR: taxonomic (is-a) and compositional (part-of)
- ample work on learning taxonomic hierarchies
- almost no work on learning CHs despite their ubiquity
- taxonomy learning in existing work is bounded for discrete data, but my CH learning can scale to *arbitrary levels of complexity* with discrete data

- Given the success of existing systems that use prespecified CHs to make predictions from context (as with the inkblot examples), given the pervasiveness of the idea of learning and then using that to learn more, and given the obvious importance of both types of hierarchy to AI, it is remarkable that there has been such a discrepancy in learning research with so little work on bottom-up CH learning.

An Example Compositional Hierarchy

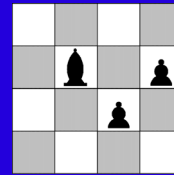


- multiple edges can link the same parent and child

- Throughout this talk the structural relationships will be simple orderings, but more complicated structural relationships are possible, such as geometric relationships.
- Many types of data can be interpreted in terms of compositional hierarchies. Letters are used here and throughout the talk since common patterns are easily recognized. However, note that the research is not about language specifically, but about general techniques applicable to language as well as other types of data, such as music, event chronologies, action sequences, or spatial configurations.
- Evidence shows that people use hierarchical compositional representations in a variety of settings.

Benefits of CHs of Common Patterns

- hierarchies of common patterns occur in: language, music, event chronologies, action sequences, spatial configurations, ...
- there are many ways to apply knowledge of common patterns besides prediction
 - memory
 - communication
 - associative thought
 - feature construction for statistical learning



Section: Introduction

Slide: 13

- Both humans and machines have a limited capacity in their fastest, most accessible memory. The effective capacity can be increased by storing only pointers to common patterns themselves stored in larger memories. This allows chess masters to perceive and remember chess boards from actual games more effectively than novices (Chase & Simon) and allows people to remember long sequences of digits (Miller).
- Pavlov's dogs would have had a much harder time learning to associate the complex perceptual patterns of food and bells if they had not first developed representations for each individually.
- Given the importance of knowledge of common patterns as identified by luminaries like Miller and Simon, and the ample evidence for the use of CHs by people, it is amazing there are not dozens of competing methods for creating such representations automatically.

Unbounded Learning Framework

- need a natural unbounded analog of unsupervised IID prediction; strive for simplicity and generality
- positionally discrete data of discrete symbols (e.g., sequences)
- instance: X_1, X_2, \dots, X_k , where $X_i \in$ finite alphabet Σ
- on-line/incremental data presentation
- unbounded, unsegmented data (snippets not strings)

- On-line learning is necessary for long-term learning in agents faced with perceptual overload from a complex environment.

Data Assumptions

- no assumption of simple, finite parameter world
- infinitely wide stationary joint distribution (Markovian limit)

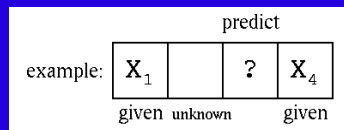
$$p \left(\overline{\dots \ x_{-2} \ x_{-1} \ x_0 \ x_1 \ x_2 \ \dots} \right)$$

- influence tends to fall off with distance
- a learner must approximate ever-widening marginal distributions
- can never fully characterize the environment, nor even make use of all relevant predictor variables!

- Though this makes the learning task quite a bit harder in an important sense than most learning frameworks, this characterization is apt for many aspects of the learning and inference that agents must do in the real world or other complex environments.

Performance Task and Evaluation

- predict any specified missing symbols given whatever symbols are present (the context)

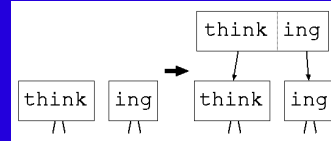


- measures of predictive performance:
 - accuracy (% correct guesses, or 0/1 loss)
 - cross entropy (average surprise, or log of perplexity)

- The task is more general than forward-only next-symbol prediction. Agents in complex environments are often faced with predicting using information from both sides, and there is evidence that human perceptual systems do not simply predict forward. Also, arbitrary prediction generalizes more naturally to higher-dimensional data.

General Strategy

- repeatedly identify frequently occurring patterns of primitives or previously identified patterns, bottom-up



- two problems to solve:
 - how to embody a CH in a predictive representation
 - how to incrementally grow the CH with new data

- This hierarchical composition process is a structural learning problem. On-line structure learning is hard. Structure learning with batch training often works by optimizing a global metric such as a Bayesian posterior or MDL score of the training set, but this cannot be done in on-line learning unless the model encodes sufficient statistics to summarize the training set with respect to the metric in question. Growing models studied here cannot contain sufficient statistics of past training data for newly grown structure. Nonetheless, building CHs of *frequent* patterns allows for this simple bottom-up strategy. The CH structure serves as a domain independent inductive bias for the structure learning.

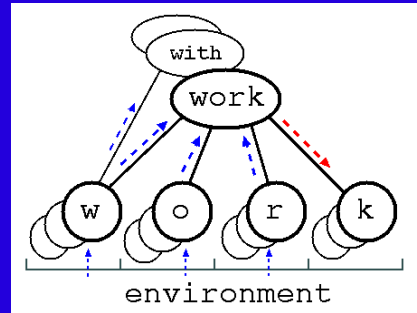
Outline

- ✓ Introduction
- Compositional probabilistic networks
 - Boltzmann machines with generalizations
 - Hebbian Chunking
- Hierarchical sparse n -grams
- Conclusion

- That's the problem: to build compositional hierarchies bottom-up, on-line, from unsegmented data alone, in such a way that they are sensitive to statistical properties and can be used to make arbitrary predictions. This includes both a qualitative goal to identify hierarchies of frequent patterns and a quantitative goal to do well at the predictions. I've described a framework in which to evaluate the predictions and a general strategy for building the hierarchies. To demonstrate that the strategy is both feasible and general, I will tell you about two systems that instantiate the strategy using representations from two different classes of traditional learning models.

Neural Nets and Graphical Models

- symmetric recurrent neural networks and graphical models can make general predictions as required
- CHs can be encoded into network structure
- activation flows along part-whole links
- *prior work did not learn weights or structure*

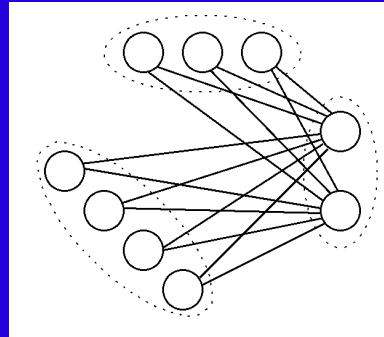


Probabilistic Nets: Boltzmann Machines

- Boltzmann machines (BMs) are weight-tunable, probabilistic nets in the symmetric recurrent neural network class and also undirected graphical models
- many nice properties (e.g., local computation)
- on-line learning rule implements max. likelihood; net converges to probability dist. of environment
- no prior work using CHs within BMs

Categorical Boltzmann Machines

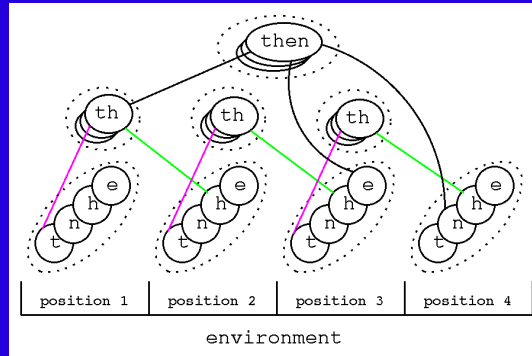
- need to generalize binary variables to categorical
- groups (or pools) of nodes represent a variable, with one node per value (instead of one/variable)
- connect two pools by connecting all nodes pairwise



- The generalization to categorical variables is well-recognized, but a clear exposition (and usually even mention of the generalization) is absent from Boltzmann machine literature. My specific formulation using nodes as values and maintaining individual real-valued weights is particularly important for enabling the chunking rule described shortly.

Weight Sharing

- we need chunks of different sizes (a hierarchy)

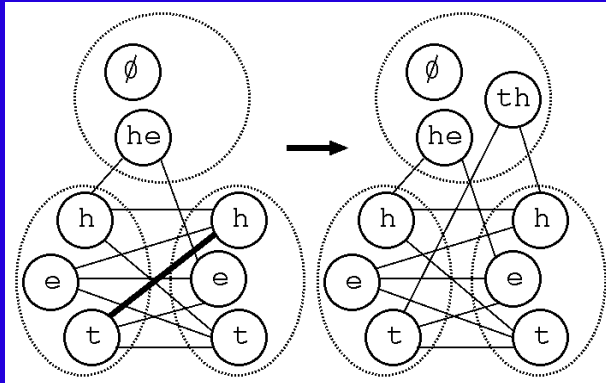


- duplicated structure with shared weights insures ability to recognize small patterns at any position

- In the field of neural networks, weights sharing is usually applied only within feedforward networks. Its use here is similar to the unrolling of HMMs or dynamic belief networks.

Hebb–Rule Based Chunking

- add new chunks by growing new hidden nodes
- trigger by correlations denoted by large Hebbian weights
- promote correlations to 1st-class entities
- 1st use of Hebbian dynamics for chunking



Section: Compositional Probabilistic Networks / Hebbian Chunking

Slide: 23

- This is the heart of the model and the first chunking rule based on Hebbian weight dynamics.
- The half-century old Hebb rule says that the weight between two neurons should increase whenever both are active together. This causes weights to converge to something like correlations. Weights between frequent pairs will grow large. By promoting strong correlations to first-class entities, the network can eventually represent higher-order correlations.
- This is also the only use of which I am aware of dynamic (i.e. on-line) hidden node creation in Boltzmann machines or any other kind of symmetric recurrent neural network or undirected graphical model.

Example Learned Chunks

- successfully learns a hierarchy of frequent chunks on–line, bottom–up, etc.
- 2–chunks: he, th, in, nd, an, er, ng, of, ed, hi, ou, ve, st, ly, on, re, as, wa, ll, ha, be, it, co, wi, ur, sh, ow, me, gh, ma, om, wh, by, ut, ch, is, to, ck, fo, ak, ul, at, ac, av, ab, yo, pr, li, br, up, po, im, or, ex, us, ic, ev, un
- 3–chunks: the, ing, and, was, her, ver, you, his, hat, wit, for, man, com, oul, hav, ugh, oug, ved, abr, con, red, all, she, eve, vin, uld, ery, hic, ich
- improves predictions with more data

Section: Compositional Probabilistic Networks / Generalized BMs

Slide: 24

- This network model satisfies the requirements set forth: bottom–up, on–line CH learning with statistical sensitivity for arbitrary prediction. It is the first system to do so. It generalizes the prior work with similar networks by adding learning abilities. It does so in a way that is locally computable and thus parallelizable or biologically realizable.

Outline

- ✓ Introduction
- ✓ Compositional probabilistic networks
- Hierarchical sparse n -grams
 - Sparse n -grams
 - On-line learning
 - Hierarchical n -grams
- Conclusion

- I will now present a second system for several reasons:
 - To demonstrate the generality of CH building across learning representations.
 - Because the prior system is inefficient in many ways. There is years of work on improving the efficiency of Boltzmann machines and many of these improvements could be folded together with CH learning, but I have not tried to contribute to this field.
 - The essence of the CH learning in the network is to decide which patterns to create nodes for at each level (width), and this selection process can be more directly studied in the context of simpler probabilistic models.
- In order to describe the next system, I must first describe the primary subcomponent of the system.

n -grams

- very simple statistical model using empirical frequencies (observed counts) as probabilities
- state-of-the-art prediction
- # parameters exponential in width

$$p_i = c_i/T$$

a	a	a	1
a	a	b	3
a	a	c	11
a	a	d	2
a	a	e	0
a	a	f	6
a	a	g	5
⋮			

Section: Hierarchical Sparse n -grams / Sparse n -grams

Slide: 26

- The same set of counts can be organized as an undirected joint distribution (shown) or as a conditional distribution of the last symbol given the previous $n-1$, which is more common. I use the undirected formulation (1) for simplicity, (2) because the undirected version more naturally generalizes to higher-dimensional data, (3) to maintain the ability to predict in arbitrary directions equally well, and (4) to maintain similarity to previous work with symmetric recurrent neural networks (my own model and prior work). Commitment to the undirected version is not an inherent aspect of CH-learning research, and it may be possible to apply similar ideas to directed versions and variants, such prediction suffix trees, or to directed networks or graphical models. Note that Moore's ADTrees allow undirected n -grams to be accessed as efficiently as directed versions, with more generality.

Sparse n -grams

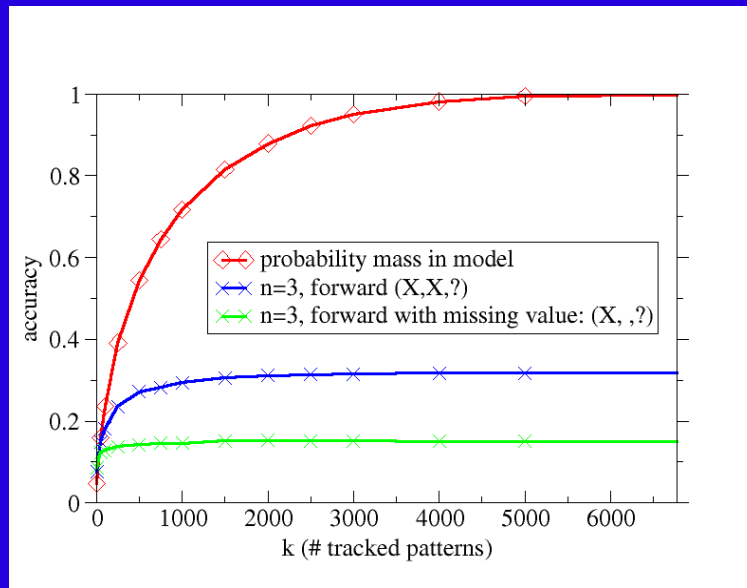
- sparse n -grams represent joints using counts for only some (frequent) patterns
- divide remaining prob. mass equally
- degrades prediction, but not much
- allows wider models; can predict better
- sparseness is inherent in KR and intimately related to compositionality (cf. Bienenstock, Geman, & Potter)

t	h	e	10899
a	n	d	4842
i	n	g	4699
h	e	r	3883
s	h	e	2697
t	h	a	2569
e	r	e	2390

Section: Hierarchical Sparse n -grams / Sparse n -grams

Slide: 27

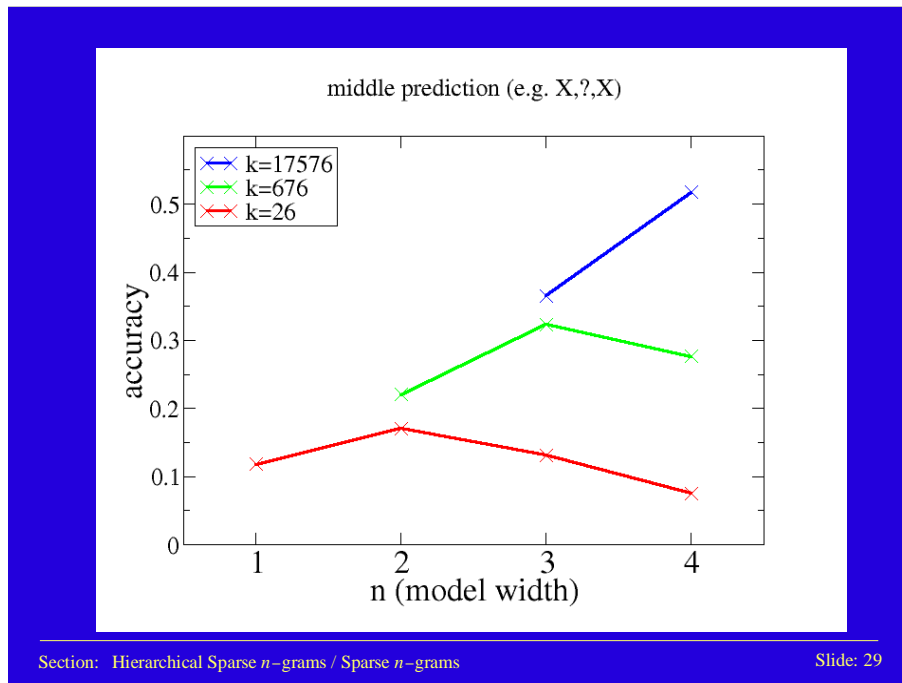
- I independently formulated sparse n -grams on the intuition that low-frequency counts should not help accuracy scores much since these do not depend on fine distinctions between the likelihoods of extremely unlikely events. Pruning low-frequency patterns from batch-trained n -grams after training using hard count cutoff thresholds has been studied before (Goodman & Gao '00) and is common in practical applications but is not as flexible and gives no insight into how to train such models on-line. Also, many of my observations and analyses about memory use and predictive performance under accuracy vs. entropy are novel.



Section: Hierarchical Sparse n -grams / Sparse n -grams

Slide: 28

- The thesis discusses in detail why sparseness has such a benign effect on accuracy and why it is significantly more benign than the effect on cross entropy.



- Two forces are at odds here. Sparseness degrades prediction for fixed n , but use of additional predictor variables increases predictive power. This shows that the value of the extra predictor variables can outweigh the detriment of sparseness.

On-Line Learning

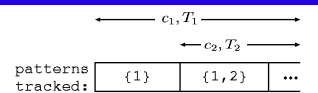
- on-line learning is hard; the model does not know *a priori* which patterns are frequent
- perform a stochastic search over pattern subsets:
 - add patterns randomly (only on occurrence)
 - throw out the infrequent and add more
- the chance of misjudging a frequent pattern to be infrequent becomes vanishingly small as it is tracked longer

- The key is that while a pattern is in the model, we can tell how frequent it is. While a truly frequent pattern may be removed soon after being added due to an anomalous stretch of data in which it is uncharacteristically infrequent, it will be added again. As it stays in the model longer and longer, its empirical frequency converges to its actual probability and the chance of a long enough drought of occurrences to remove it becomes vanishingly small.

Predicting with Unreliable Estimates

- since patterns were added at different times, not all probability estimates are equally reliable

- newer estimates will fluctuate more; model distribution should not be jarred (e.g., by new addition)



$$p_1 = c_1/T_1$$

$$p_2 = \frac{1}{T_1} \left[c_2 + (T_1 - T_2)(1 - p_1) \frac{1}{|A|^n - 1} \right]$$

$$p_i = \frac{1}{T_1} \left[C_i + \sum_{j=1}^{i-1} (T_j - T_{j+1})(1 - \sum_{j=1}^i p_j) \frac{1}{|A|^n - j} \right]$$

- simple approach interpolates empirical frequency and untracked average

Bayesian/EM Approaches

- Bayesian solution to slight variant of problem
- EM approach: treat identities of untracked patterns as hidden information, estimate using model dist.

$$p_i = \frac{1}{T_1} \left[c_i + \sum_{j=1}^{i-1} c_{\{j+1,k\}}^j \frac{p_i}{1 - \sum_{l=1}^j p_l} \right]$$

- solvable analytically; equivalent to Bayesian solution

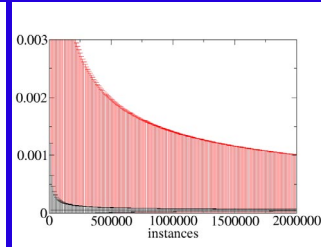
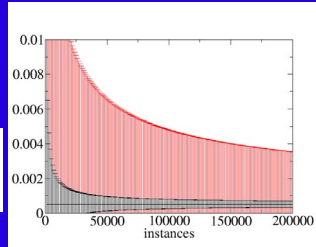
- There are reasons to suspect that this solution may converge to empirical frequencies too aggressively, but we'll see how to improve upon this formula shortly anyway.

Improved Hoeffding Races

- need to decide when to remove a low frequency pattern and which one
- adapt and improve Moore's use of Hoeffding races, based on bounding empirical frequency error

$$\epsilon \leq \sqrt{\frac{\ln(2/\delta)}{2T}}$$

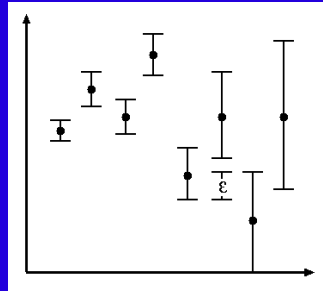
$$\epsilon_+ \leq \sqrt{\frac{3\ln(2/\delta)}{T}} \sqrt{p}$$



- When considering a pattern to drop, this allows a smooth weighting of how converged the empirical estimate is and how much more frequent the other patterns are.
- The bound improves the closer p is to 0, and thus should be better for larger n .

Stochastic Search

- remove a pattern when reasonably confident it is within ϵ of being least frequent
- with small probability, add a pattern that occurs during training
- converges to batch performance, but slowly for large widths



- Specifically, the model removes the pat with lowest upper bound once its upper bound is within ϵ of the (next-) lowest lower bound.

Hierarchical Sparse n -grams

- multi-width combinations of sparse n -grams
- implicit compositional structure
- improves:
 - probability estimation
 - pattern selection

a	47897	t	h	18038	t	h	e	10899	t	h	e	r	1888
b	10407	h	e	17888	a	n	d	4842	t	h	a	t	1639
c	12967	i	n	11175	i	n	g	4699	n	t	h	e	1637
d	26472	e	r	10568	h	e	r	3883					
e	71680	a	n	9582	s	h	e	2697					
f	12465	r	e	7519									
g	12631	e	d	6972									
h	37889												

Improved Probability Estimation

- use lower model's ($n-1$) distribution to estimate which untracked patterns occurred

$$p_{D_n}(2) = \frac{1}{T_{max}} [C_2 + (T_{max} - T_1)p_{D_{n-1}}(2) + (T_1 - T_2)(1 - p_{D_n}(1)) p_{D_{n-1}}(2 | \neg 1)]$$

- widening $n-1$ distribution by one is straightforward
- factorization of quadratic number of now-different terms maintains linear computation of all p 's
- fall back directly on renormalized lower dist. for untracked patterns

Section: Hierarchical Sparse n -grams / Hierarchical n -grams

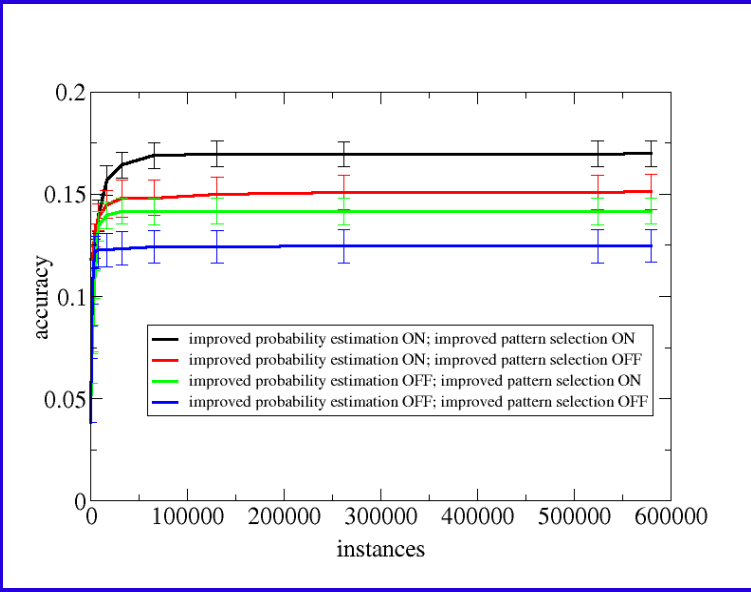
Slide: 36

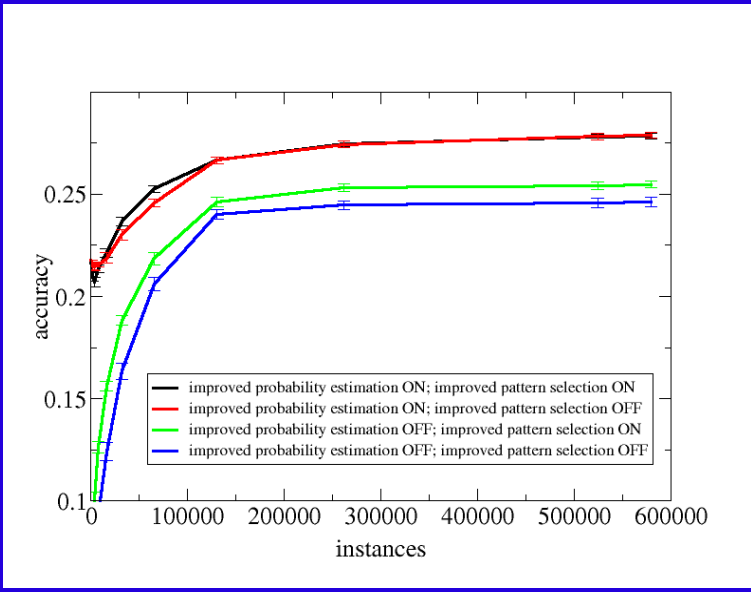
- This is one of the two main ideas of this section.
- The lower order distribution converges exponentially faster than the distribution at the current n . Thus, it is much better than the Bayesian/EM formula using order- n information only. This uses distribution information that is poor exactly where it is needed, early in training before everything has converged to accurate empirical frequencies.
- This is a novel method for falling back on a lower order model when combining n -grams of different widths. Nice properties: Adding a new widest width will not jar the distribution of the model as a whole. Estimates both within and across widths are smoothly weighted according to their reliabilities, and authority smoothly transitions to greater widths with more training data.

Improved Pattern Selection

- searching for frequent patterns randomly is intractable
- bias selections toward frequent patterns based on the lower-order model
- effectively a refinement of stochastic composition of smaller existing patterns
- on-line stochastic analog of pruning in association rule mining

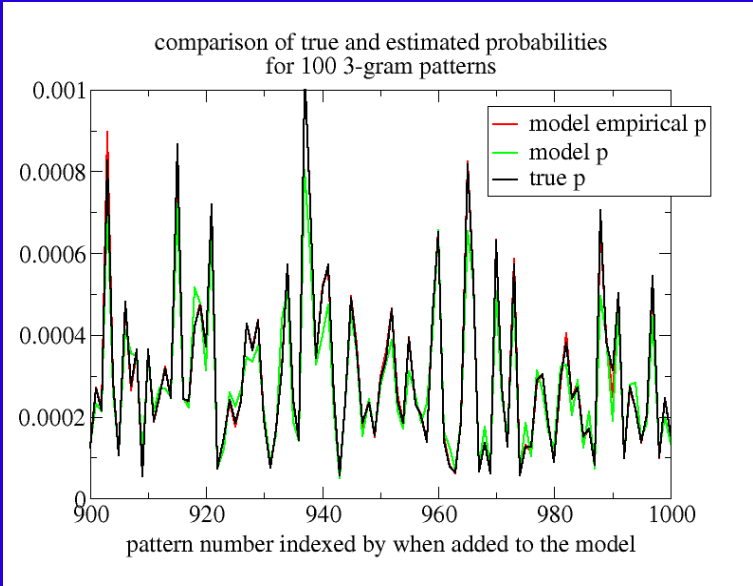
- This is the other key idea of this model and the key to its composition (chunking) rule.
- Hierarchy of refinement:
 - Randomly combine two existing patterns.
 - Take into account the number of parses of the new pattern into existing patterns, including overlaps.
 - Factor in the probabilities of all component patterns.
 - Simply use the $n-1$ dist. probability for the candidate pattern, which smoothly integrates all these factors.
- This is analogous to a stochastic version of the pruning used in association rule mining to reduce the number of candidate combinations to consider based on smaller existing market baskets. The on-line nature of the algorithm here, which prevents the sets of patterns at any level from being exactly the frequent ones, prevents the exact same pruning from being done here.





n	2	3	4	5	6	7	8	9
k	367	1540	1131	433	142	38	11	7
unique	670	12556	121799	626465				
top100	100	100	99	75				
top1000	367	910	560	238				
	th	the	tion	ation	nation	saidthe	official	president
	he	ing	said	ofthe	saidth	esident	resident	ternation
	in	and	nthe	inthe	ingthe	residen	presiden	residento
	er	ion	ther	tions	ations	nationa	thenatio	theminst
	an	ent	dthe	ingth	aidthe	preside	tsaidthe	andforthe
	re	ter	atio	edthe	sident	thatthe	roundthe	isingthat
	on	tio	fthe	there	ationa	officia	essaidth	isionthea
	es	for	that	saidt	esiden	fficial	oftheall	
	ed	ere	ofth	tiona	reuter	withthe	ancountr	
	st	int	ingt	ngthe	reside	eration	ngtheser	
	en	nth	ment	aidth	andthe	ssaidth	hadahear	
	nt	her	tthe	ingto	forthe	inister		
	at	ati	thes	tothe	offici	tingthe		
	to	tha	inth	natio	hesaid	ections		
	or	ers	ethe	atthe	ration	ringthe		
	te	aid	with	orth	minist	ministr		
	ti	est	rthe	esaid	ficial	rationa		
	ea	dth	othe	ction	thath	ionofth		
	ar	sai	ions	onthe	tofthe	onatio		
	nd	ate	thec	state	sinthe	thegove		
	al	res	sthe	siden	ection	nationo		
	sa	ort	sand	eside	ctions	ationso		
	it	ons	edth	their	romthe	rsaidth		
	ng	ted	here	idthe	iththe	aidthec		
	as	edt	ithe	ndthe	illion	thesout		
	ha	eth	over	andth	terthe	iations		

top 30 chunks
of each size
ordered by
model
probability



Outline

- ✓ Introduction
- ✓ Compositional probabilistic networks
- ✓ Hierarchical sparse n -grams
- Conclusion
 - Related work
 - Broader picture
 - Summary, contributions, and future

Work Related to CHs

- prespecified CHs: HEARSAY-II, IAM (McClelland & Rumelhart '81)
- non-predictive: SEQUITUR (Nevill-Manning '96), MK10 (Wolff '75)
- SCFG induction: Stolcke & Omohundro '94, Langley & Stromsten '00
- hierarchical RL: Ring '95, Drescher '93, Andre '98, Sun & Sessions '00
- compression: Lempel-Ziv
- segmentation: Olivier '68, Redlich '93, Saffran et al '96, Brent '99, Venkataraman '01, Cohen '01
- hierarchical HMMs: Fine et al '98, Murphy '01
- layered reps. in parameterized nets: Lewicki & Sejnowski '97, Hinton
- speedup learning: Soar, EBL
- misc: de Marken, Geman & Potter

- There is a large but diverse collection of existing work involving compositional structures, much of which shares overlapping motivations or research themes. An unfortunate feature of this literature is its rather low density of inter-citation given the broad similarities.
- Despite this, no prior system learns predictive compositional hierarchies in an on-line/incremental fashion purely from unsegmented data.

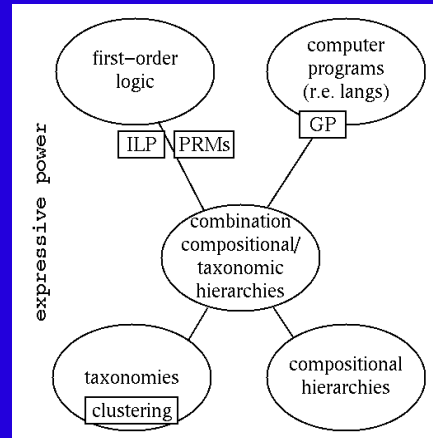
Work Related to Learning Framework

- SPARC (Dietterich & Michalski '86)
- n -grams: lots of work
- PPM* (Cleary & Teehan '97), TDAG (Laird & Saul '94)
- **pruned n -grams (Goodman & Gao '00)**
- **PSTs: Ron et al. '96, Pereira et al. '95, Willems et al. '95**
- HMMs, DBNs

- This work does not specifically use compositional representations, but could be applied within the stated learning framework.

Expressive Power

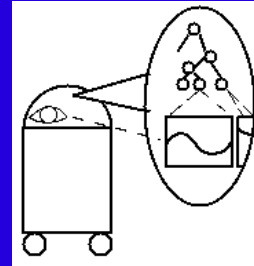
- CHs like those discussed are the simplest representation capable of embodying unbounded complexity
- HHMMs may be the first example of a model class in the middle combination oval



- CHs or combination compositional/taxonomic hierarchies are representationally less powerful than logic or recursively enumerable languages, but this can be a blessing for learning.

Symbol Grounding

- how do symbols derive meanings separate from associations in the heads of external observers?
- grounding is correspondence to environment signals
- bottom-up hierarchy learning allows grounding of representations of increasingly expansive signals, creating reps. that are grounded *and* sensitive to statistical properties of the environment all the way up



Section: Conclusion / Broader Picture

Slide: 46

- Many people think grounding is a critically important problem. There has been work, especially recent interest, in learning to ground low-level symbols in signals, but these techniques would not work well trying to ground high-level symbols directly. This is the first coherent vision of how to scale to grounded higher-level representations in a way that is sensitive to the statistical properties of the environment.

Contributions: Detailed

- novel learning framework for unbounded data
- constructive node creation rule for Boltzmann machines that is on-line and locally computable
- new undirected sparse n -gram model and explanation of effect of sparseness on prediction
- on-line training method for sparse n -grams
- probability estimation techniques for such training
- two new methods for utilizing lower-order n -gram models when combining different orders

Contributions: General

- method for incrementally building representations of common patterns from data
- extension of early non-learning CH work
- refinement of statistically insensitive chunking
- complement to ML taxonomy work
- bridge across the granularity gap, low to high level

The Future

- combine compositional and taxonomic learning
- generalize to 2D data, non-discrete data, etc.
- generalize to other sequential part-whole relations (e.g., chunks that skip positions)
- combine with connectionist symbol processing
- fold in goal/importance sensitivity
- other uses (transfer, semi-supervised learning), integration into autonomous agent architectures
- graft CH learning into other learning models

- End

Extras

- Motivation summary
- Related work summary table
- Grafting CH learning into other systems (e.g., PSTs, Venkataraman, Woods and Sanner)
- Moore's ADTrees for efficient n -gram inference

Motivation Summary

- successful prediction from prespecified CH systems
- idea of learning and using that to learn more
- importance of compositional representations in AI
- scarcity of learning CHs compared with taxonomies
- evidence for CH representations in people
- importance of knowledge of common patterns for many activities of significance to autonomous agents

Acknowledgements

- Barbara Hayes–Roth, Richard Fikes, Nils Nilsson
- Chris Manning, Joshua Tenebaum
- Pat Langley, David Rumelhart, Craig Nevill–Manning
- Quail, MLC++, MLRG, Learning Seminar, TSG, Signals–to–Symbols Symp, CMSS, Trailer folks, Patrick, Urszula, Eyal, Pedrito, Ronny, George J., Scott B., Ofer, Illah, Mehran, Uri, Simon, Lise, Scott R., Marko, Sean, Jeff, Gary, Craig S., Eugene, Karel, Luis, Dzin, Sam, Yogo
- 14H, 450, 12/13/14'hood, Thur crew, Tue crew, Kent, Laura D., Mark, Ben, Steve H., Paul, Randall, Uli, Steve F., Mike, Russ, Charles, Rick, Tallis, Tad, Misa, Pam, Cheryl, Saskia, Laura C., Emily O., Emily P., Dave, Jason, Alex, Tom, Teresa, Amy, Doug, Mary, George M., Jae, Kortney
- Mom, Dad, Gram, Cathie