Recap: Inference in Probabilistic Graphical Models

R. Möller
Institute of Information Systems
University of Luebeck

A Simple Example

$$P(A,B,C) = P(A)P(B,C \mid A)$$
$$= P(A) \quad P(B|A) \quad P(C|B,A)$$

$$= P(A) \quad P(B|A) \quad P(C|B)$$

C is conditionally independent of A given B

Graphical Representation ???

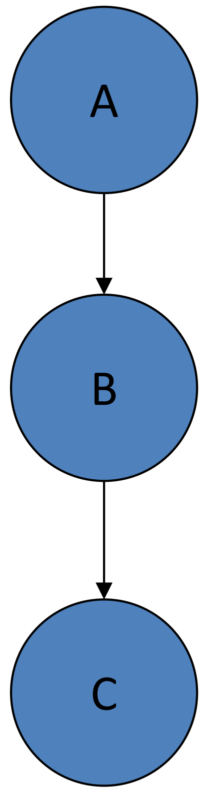
Bayesian Network

Directed Graphical Model

$U = (V_1, \ldots, V_n)$

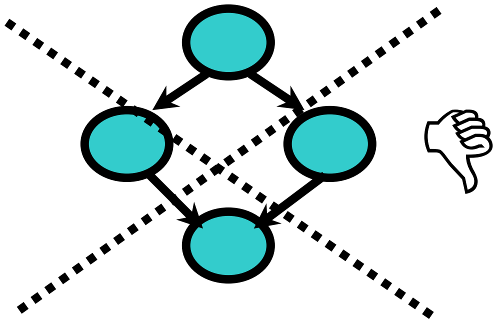
$$P(U) = \prod P(V_i \mid Pa(V_i))$$

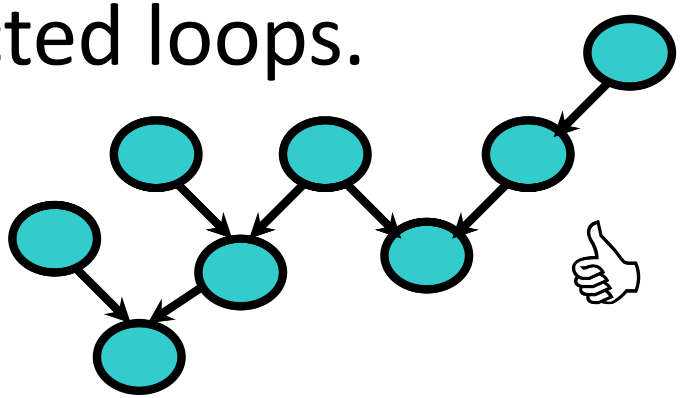
$$P(A,B,C) = P(A)\, P(B \mid A)\, P(C \mid B)$$



Digression: Polytrees

- A network is *singly connected* (a *polytree*) if it contains no undirected loops.





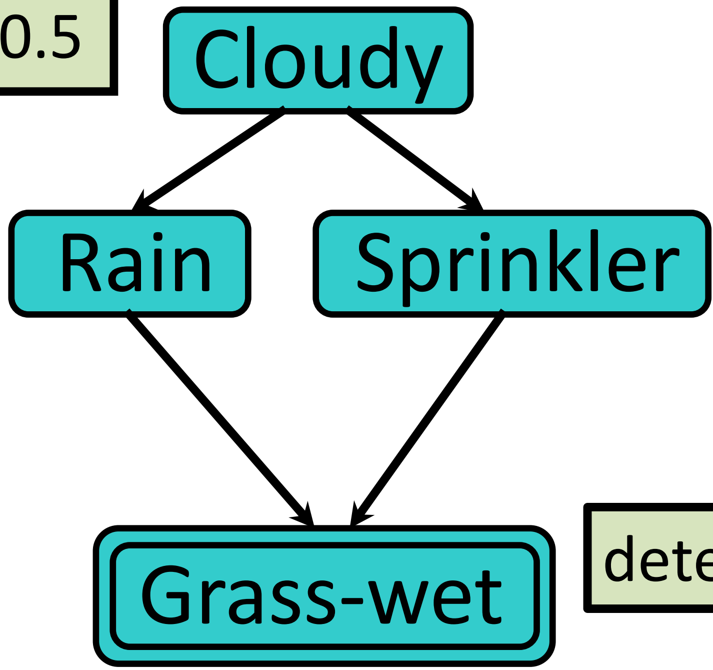
Theorem: Inference in a singly connected network can be done in linear time*.

Main idea: in variable elimination, need only maintain distributions over single nodes.

* in network size including table sizes.

The problem with loops

P(c)	0.5



	c	$\bar{c}$
P(r)	0.98	0.01

	c	$\bar{c}$
P(s)	0.02	0.99

deterministic or

The grass is dry only if no rain and no sprinklers.

$$P(\bar{g}) = P(\bar{r}, \bar{s}) \sim 0$$

The problem with loops contd.

$$P(\bar{g}) = \overbrace{P(\bar{g} \mid r, s)}^{0} P(r, s) + \overbrace{P(\bar{g} \mid r, \bar{s})}^{0} P(r, \bar{s})$$

$$+ \underbrace{P(\bar{g} \mid \bar{r}, s)}_{0} P(\bar{r}, s) + \underbrace{P(\bar{g} \mid \bar{r}, \bar{s})}_{1} P(\bar{r}, \bar{s})$$

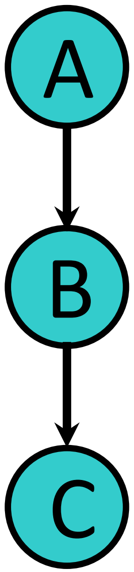
$$= P(\bar{r}, \bar{s}) \quad \sim 0$$

Propagation

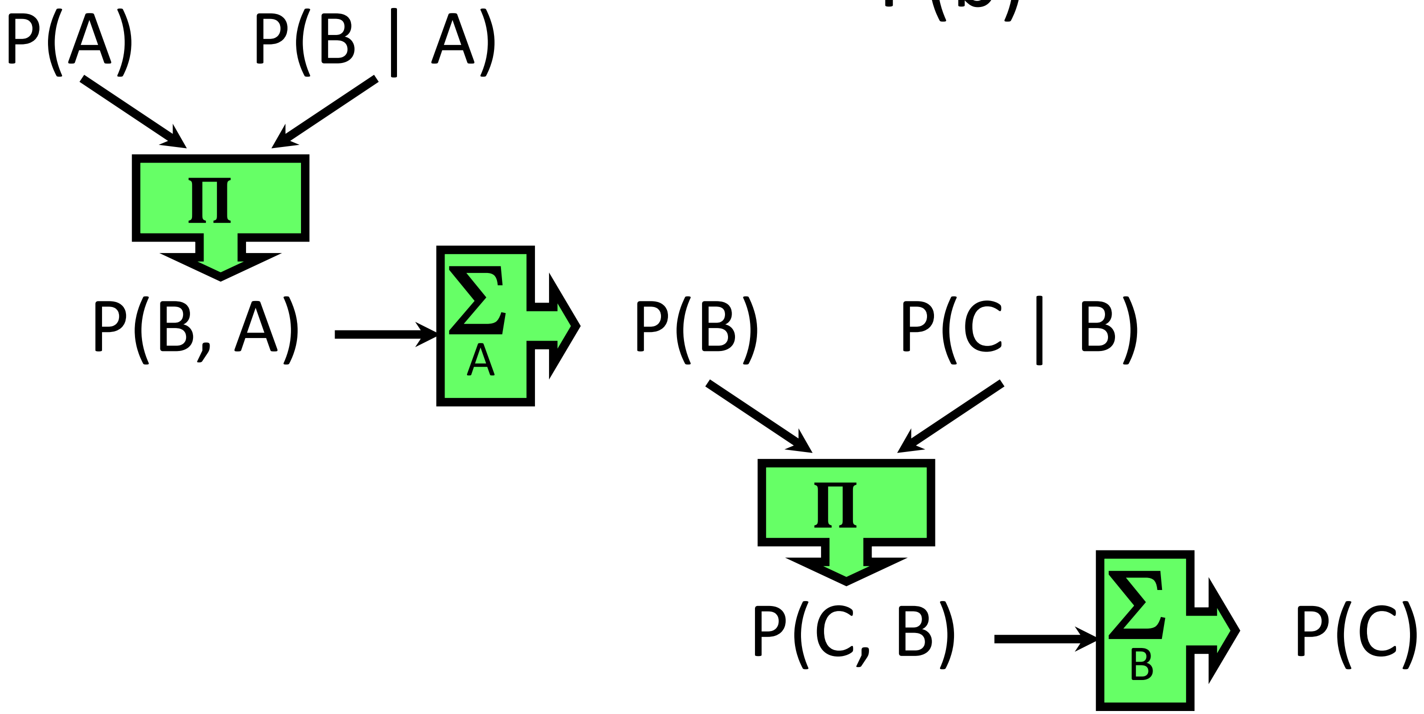
$$\neq P(\bar{r})\, P(\bar{s}) \sim 0.5 \cdot 0.5 = 0.25$$

problem

Variable elimination



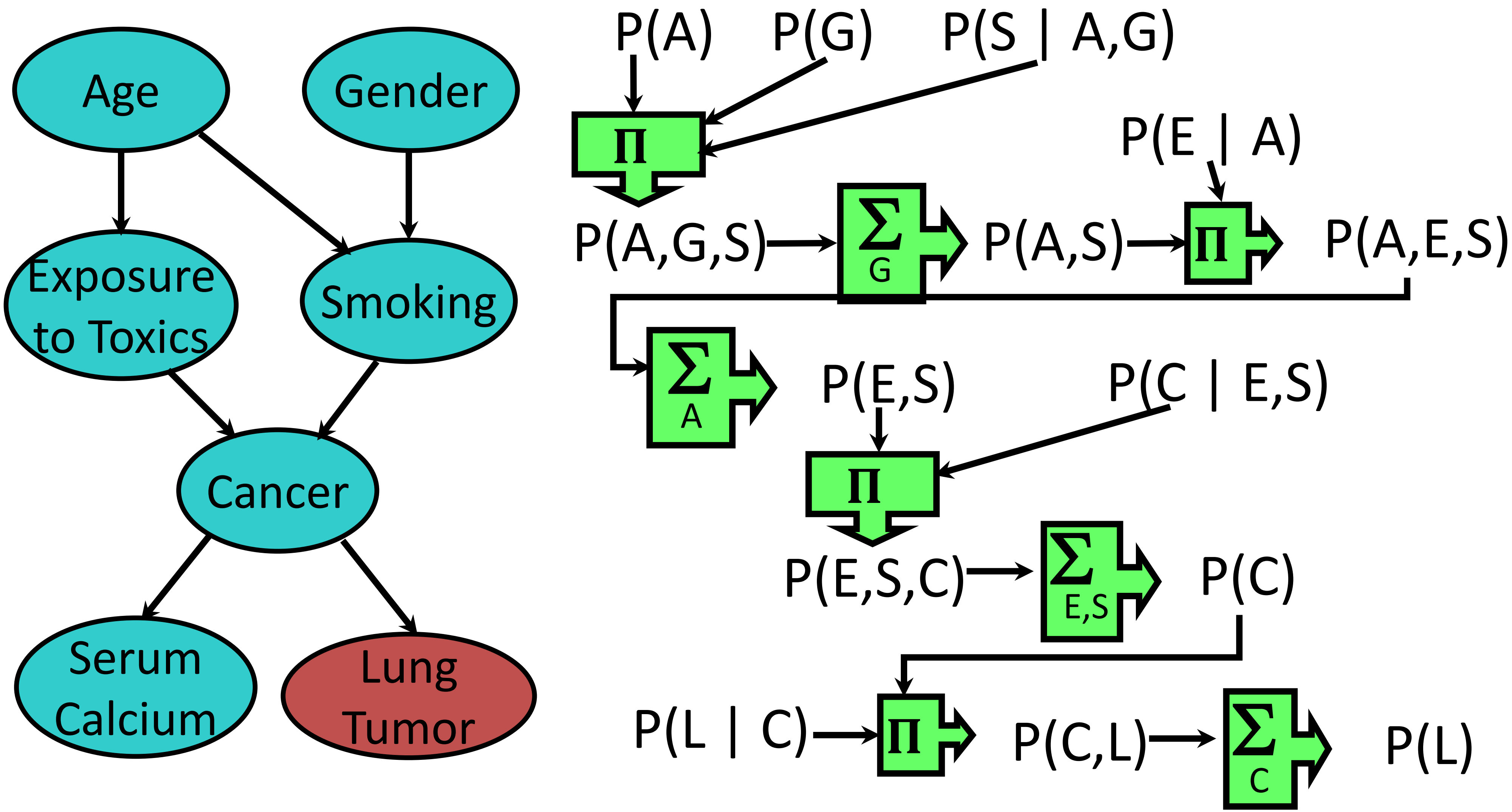
$$P(c) = \sum_b P(c \mid b) \underbrace{\sum_a P(b \mid a)\, P(a)}_{P(b)}$$



Inference as variable elimination

- A **factor** over ***X*** is a function from *val(**X**)* to numbers in [0,1]:
  - A CPT is a factor
  - A joint distribution is also a factor
- BN inference:
  - factors are multiplied to give new ones
  - variables in factors summed out
- A variable can be summed out as soon as all factors mentioning it have been multiplied.

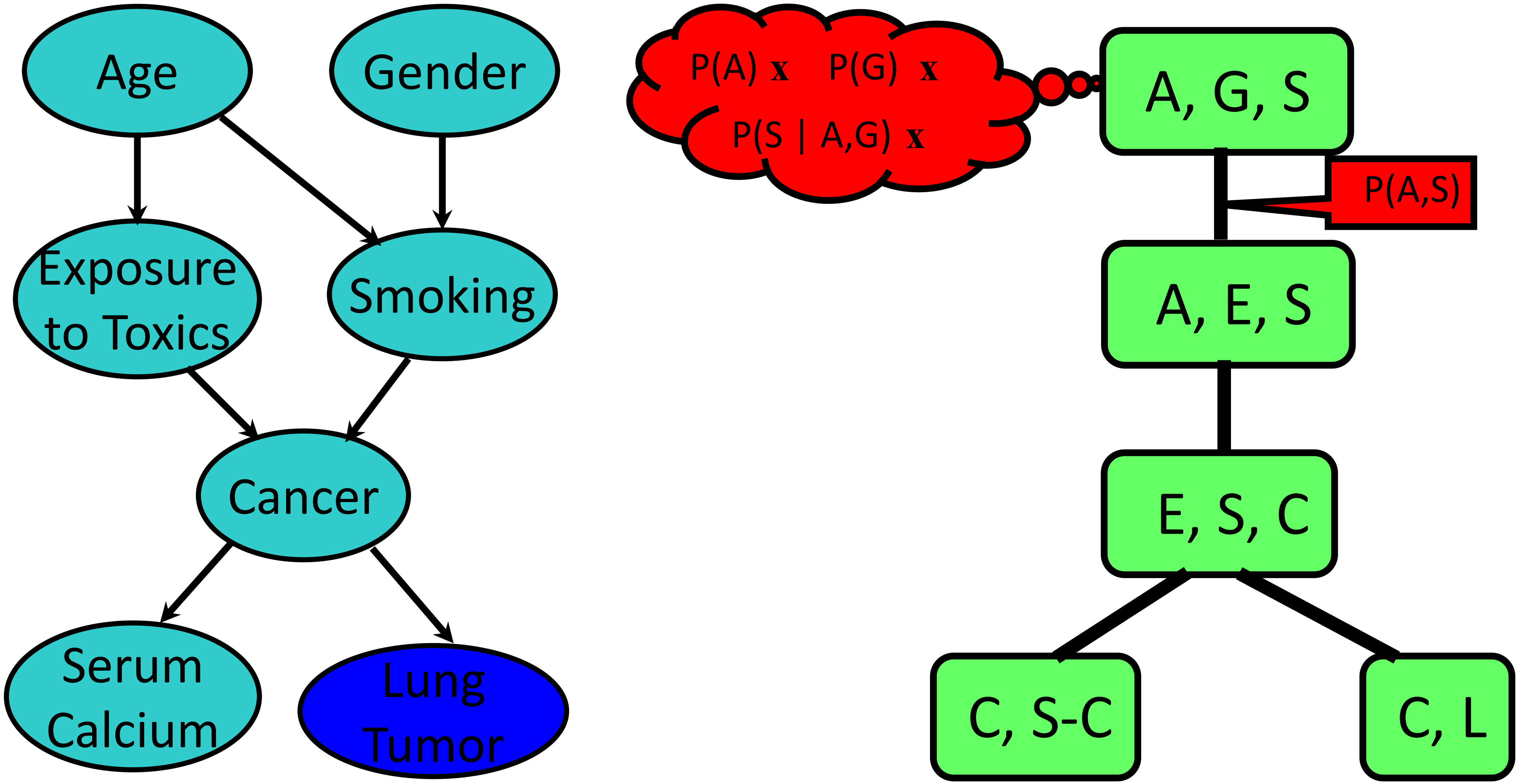
Variable Elimination with loops



Complexity is exponential in the size of the factors

Join trees*

A join tree is a partially precompiled factorization

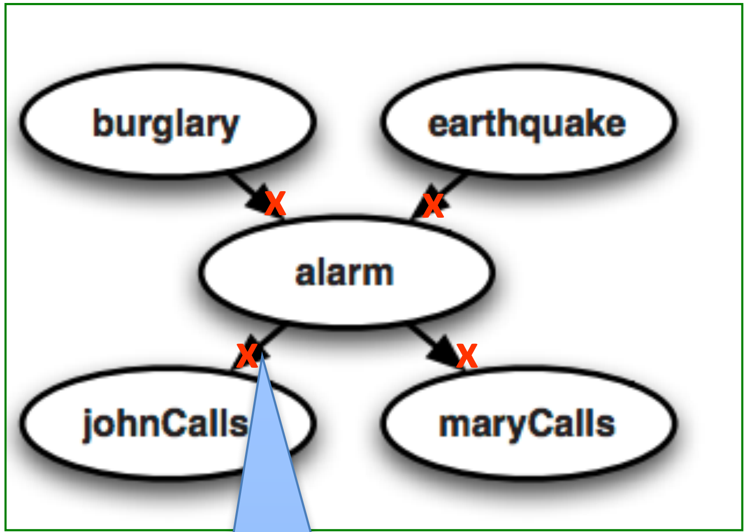


* aka Junction Tree, Lauritzen-Spiegelhalter, or Hugin algorithm, …

Background: Markov networks

- Random variable: B,E,A,J,M
- Joint distribution: Pr(B,E,A,J,M)

- Undirected graphical models give another way of defining a compact model of the joint distribution…via potential functions.
- φ(A=a,J=j) is a scalar measuring the "compatibility" of A=a J=j



A	J	φ(a,j)
F	F	20
F	T	1
T	F	0.1
T	T	0.4

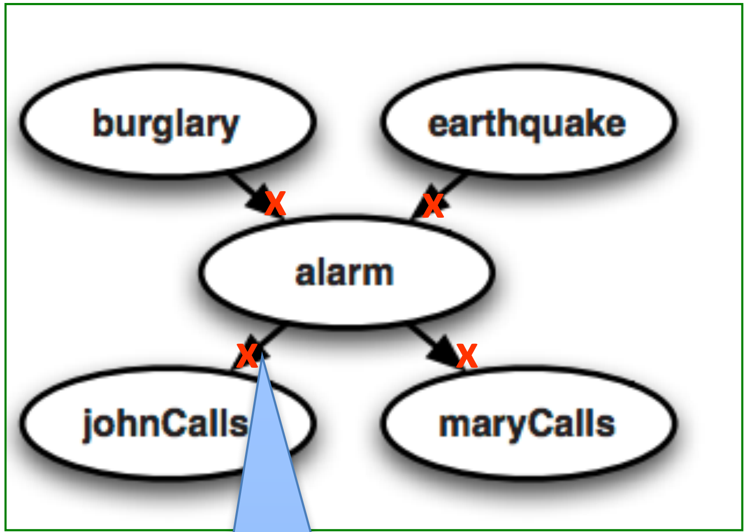
Background

$$\Pr(B=b, E=e, A=a, j, m)$$
$$= \frac{1}{Z}\phi_{JA}(a,j)\phi_{MA}(a,m)\phi_{AB}(a,b)\phi_{AE}(a,e)\phi_{E}(e)\phi_{B}(b)$$

clique potential

…

…



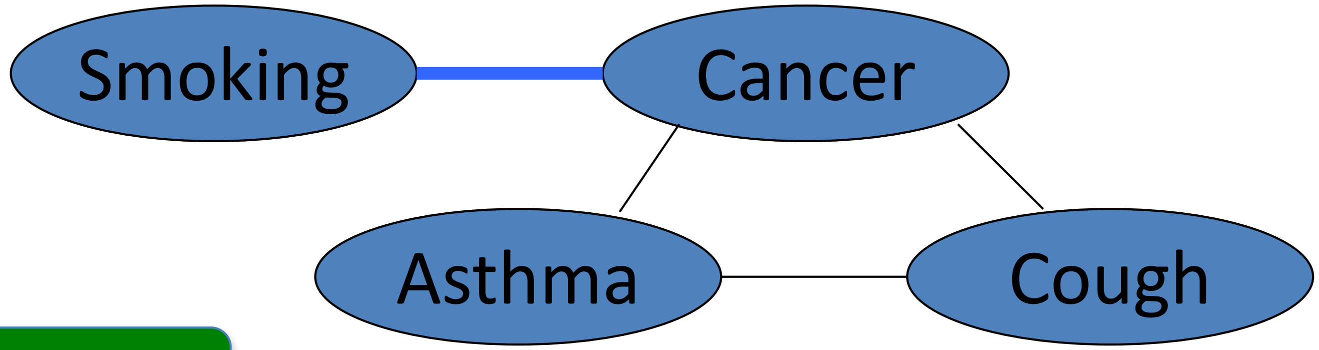
A	J	φ(a,j)
F	F	20
F	T	1
T	F	0.1
T	T	0.4

- φ(A=a,J=j) is a scalar measuring the "compatibility" of A=a J=j

Another example

- **Undirected** graphical models

[h/t Pedro Domingos]



x = vector

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

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

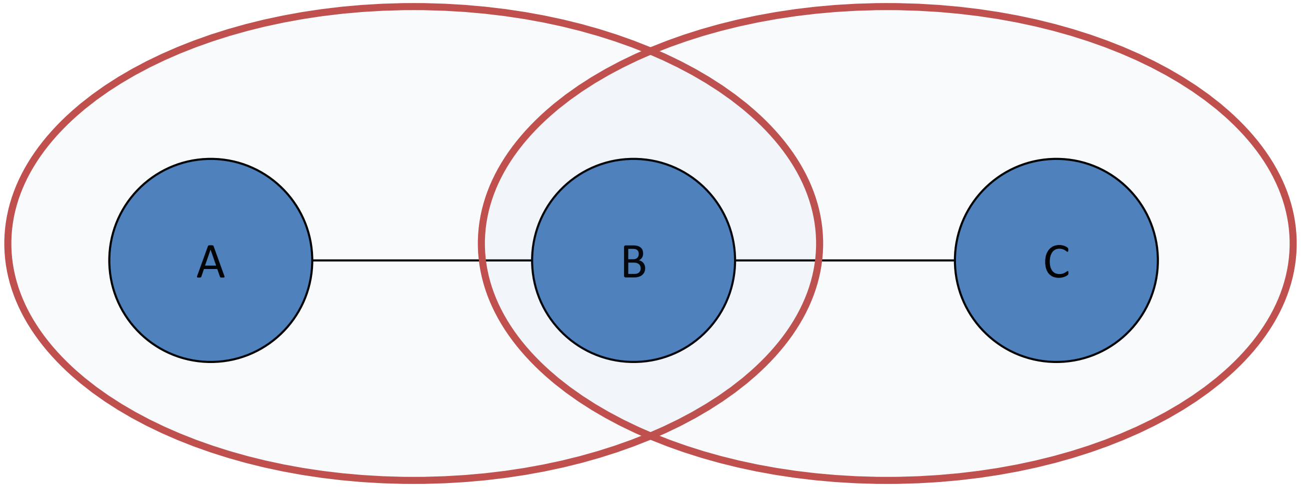
x_c = short vector

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

H/T: Pedro Domingos

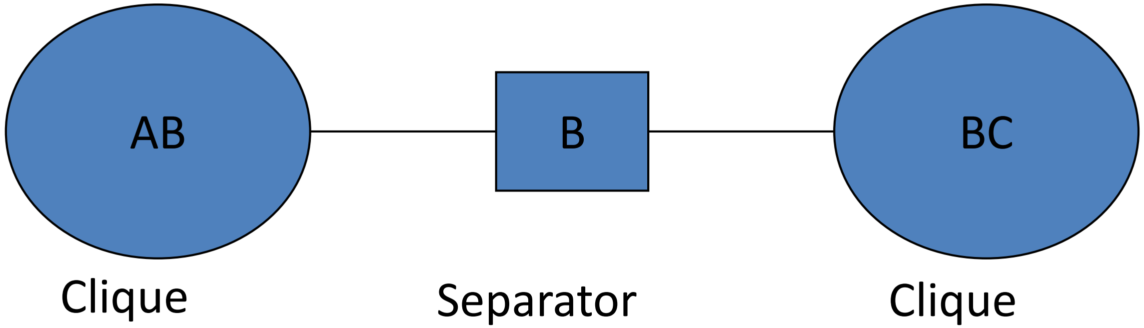
Markov Networks = Markov Random Fields

Undirected Graphical Model



Markov Random Fields

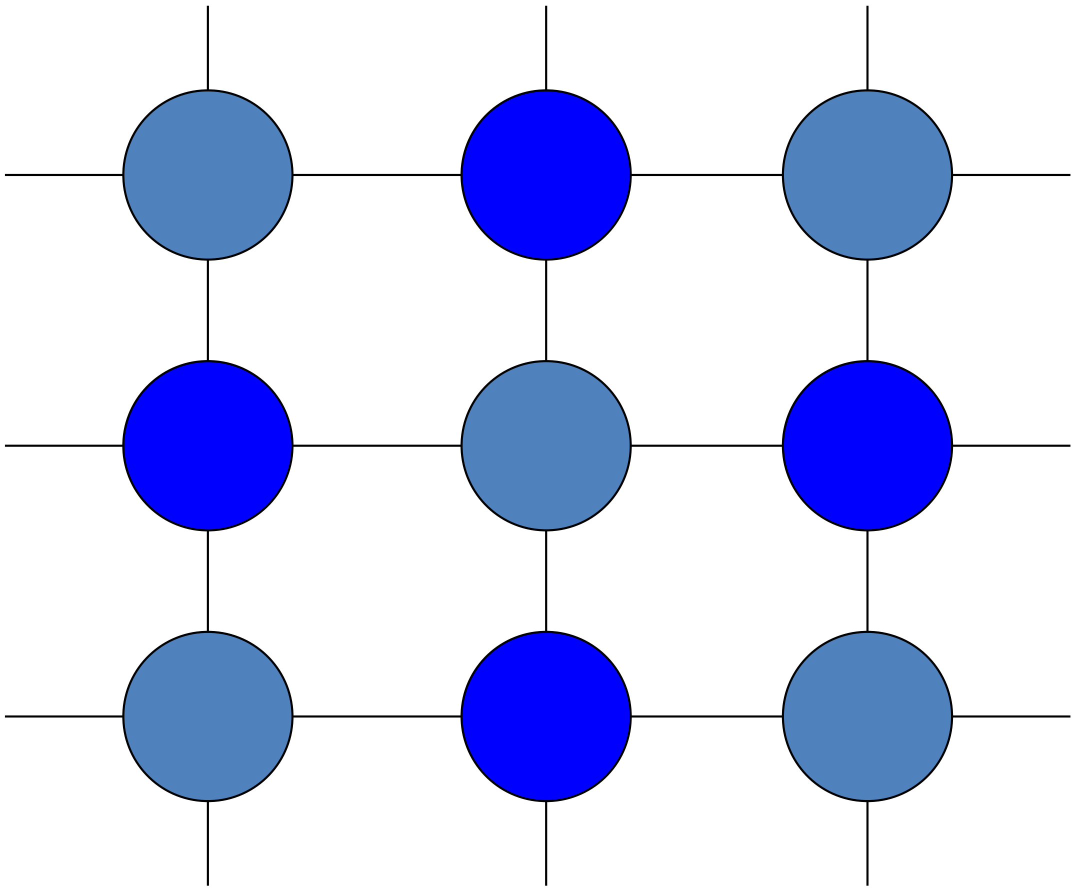
Undirected Graphical Model



$$P(U) = \prod P(\text{Clique}) \;/\; \prod P(\text{Separator})$$

$$P(A,B,C) = P(A,B)\,P(B,C)\;/\;P(B)$$

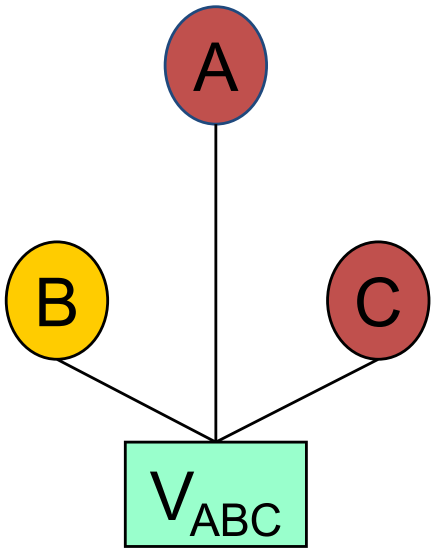
Markov Random Fields

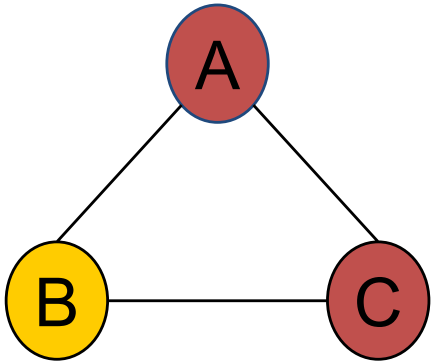


A node is conditionally independent of all others given its neighbours.

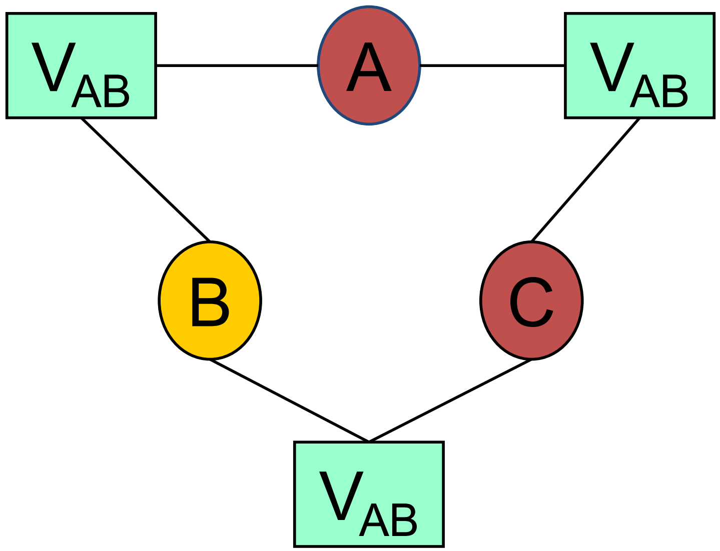
Factor Graphs

- Example
  - Exponential (joint) parameterization
  - Pairwise parameterization





Markov network

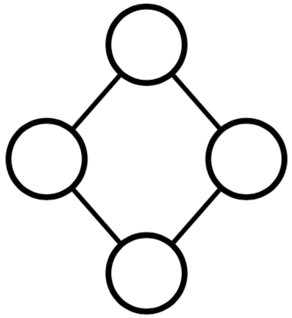


Factor graph for
joint parameterization

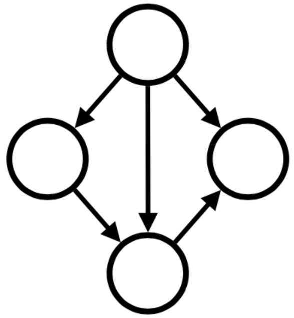
Factor graph for
pairwise parameterization

Transforming MRFs into BNs and back

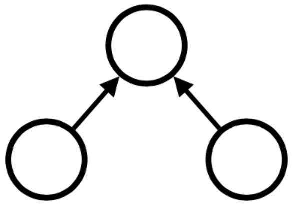
Because MRF and BN are incomparable, some independence structure is lost in conversion



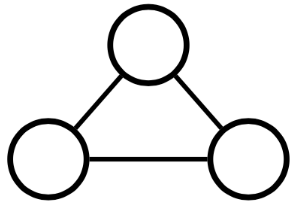
$\mu(x) = \psi(x_1, x_2)\psi(x_1, x_3)\psi(x_2, x_4)\psi(x_3, x_4)$
$x_1 \perp x_4 | (x_2, x_3)$
$x_2 \perp x_3 | (x_1, x_4)$



$x_2 \perp x_3 | (x_1, x_4)$



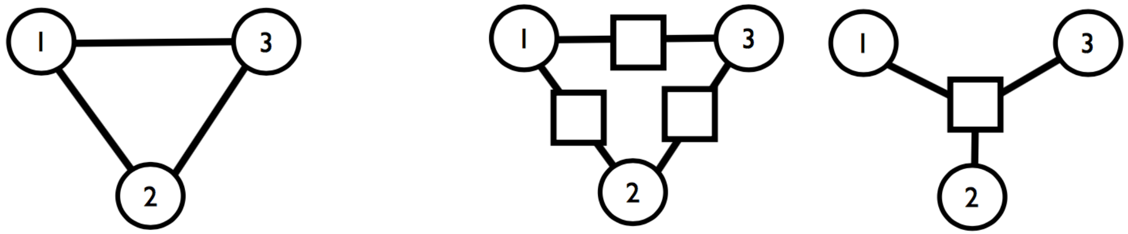
$\mu(x) = \mu(x_2)\mu(x_3)\mu(x_1 | x_2, x_3)$
$x_2 \perp x_3$



no independence

Factor Graphs vs. MRFs

Factor graphs are more 'fine grained' than undirected graphical models



$\psi(x_1, x_2, x_3)$ $\quad$ $\psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)\psi_{31}(x_3, x_1)$ $\quad$ $\psi_{123}(x_1, x_2, x_3)$

all three encodes same independencies, but different factorizations (in particular the degrees of freedom in the compatibility functions are $3|\mathcal{X}|^2$ vs. $|\mathcal{X}|^3$)

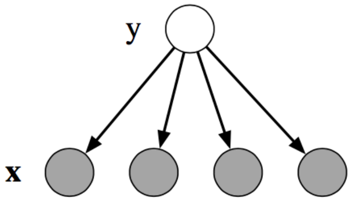
- set of independencies represented by MRF is the same as FG
- but FG can represent a larger set of factorizations

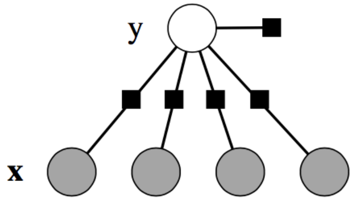
BNs – MRFs – FGs

- undirected graphical models can be represented by factor graphs
    - we can go from MRF to FG without losing any information on the independencies implies by the model
- Bayesian networks are not compatible with undirected graphical models or factor graphs
    - if we go from one model to the other, and then back to the original model, then we will not, in general, get back the same model as we started out with
    - we lose any information on the independencies implies by the model, when switching from one model to the other

Generative vs. Discriminative

Generative ML or MAP Learning: *Naïve Bayes*

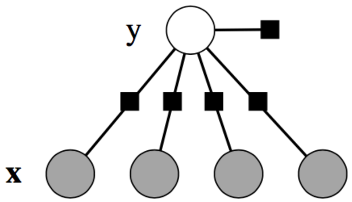




$$p(y, x) = p(y) \prod_{m=1}^{M} p(x_m \mid y)$$

- Class-specific distributions for each of *M* features

Discriminative ML or MAP Learning: *Logistic regression*



$$p(y = k \mid x, \theta) = \frac{1}{Z(x, \theta)} \prod_{m=1}^{M} \exp\left\{\theta_k^T \phi(x_m)\right\}$$

$$Z(x, \theta) = \sum_{k=1}^{K} \prod_{m=1}^{M} \exp\left\{\theta_k^T \phi(x_m)\right\}$$

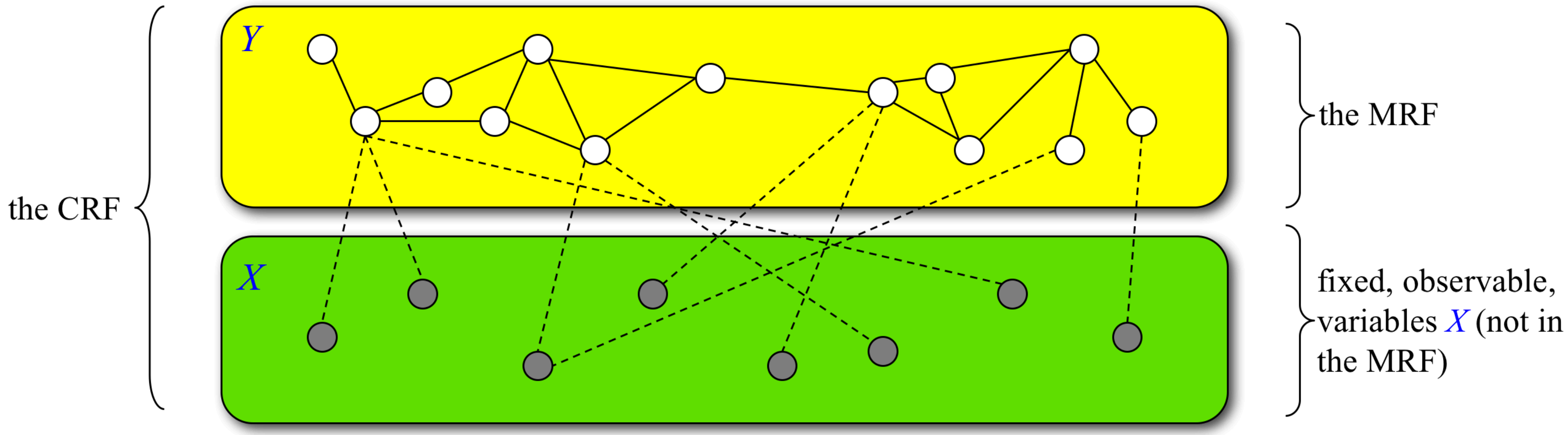
- Exponential family distribution (maximum entropy classifier)
- Different distribution, and normalization constant, for each x

Conditional Random Field

- A Conditional random field (CRF) is a Markov random field of unobservables which are globally conditioned on a set of observables (Lafferty et al., 2001)

Lafferty, J., McCallum, A., Pereira, F. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann. pp. 282–289. **2001**

A Conditional random field is effectively an MRF plus a set of "external" variables X, where the "internal" variables Y of the MRF are the unobservables (○) and the "external" variables X are the observables (●):



Thus, we could denote a CRF informally as:

$$C=(M, X) \qquad P(\mathbf{Y} \mid \mathbf{X})$$

for MRF M and external variables X, with the understanding that the graph $G_{X \cup Y}$ of the CRF is simply the graph G_Y of the underlying MRF M plus the vertices X and any edges connecting these to the elements of G_Y.

Note that in a CRF *we do not explicitly model any direct relationships between the observables (i.e., among the X)* (Lafferty *et al.*, 2001).

Duke UNIVERSITY


KLAR
SOWEIT ?

Augmenting Probabilistic Graphical Models with Ontology Information: Object Classification

R. Möller
Institute of Information Systems
University of Luebeck

Based on ECCV14 paper:

Large-Scale Object Recognition using Label Relation Graphs

Jia Deng[1,2], Nan Ding[2], Yangqing Jia[2], Andrea Frome[2], Kevin Murphy[2], Samy Bengio[2], Yuan Li[2], Hartmut Neven[2], Hartwig Adam[2]

University of Michigan[1], Google[2]

Object Classification

- Assign semantic labels to objects

Dog ✔
Corgi ✔
Puppy ✔
Cat ✖

Object Classification

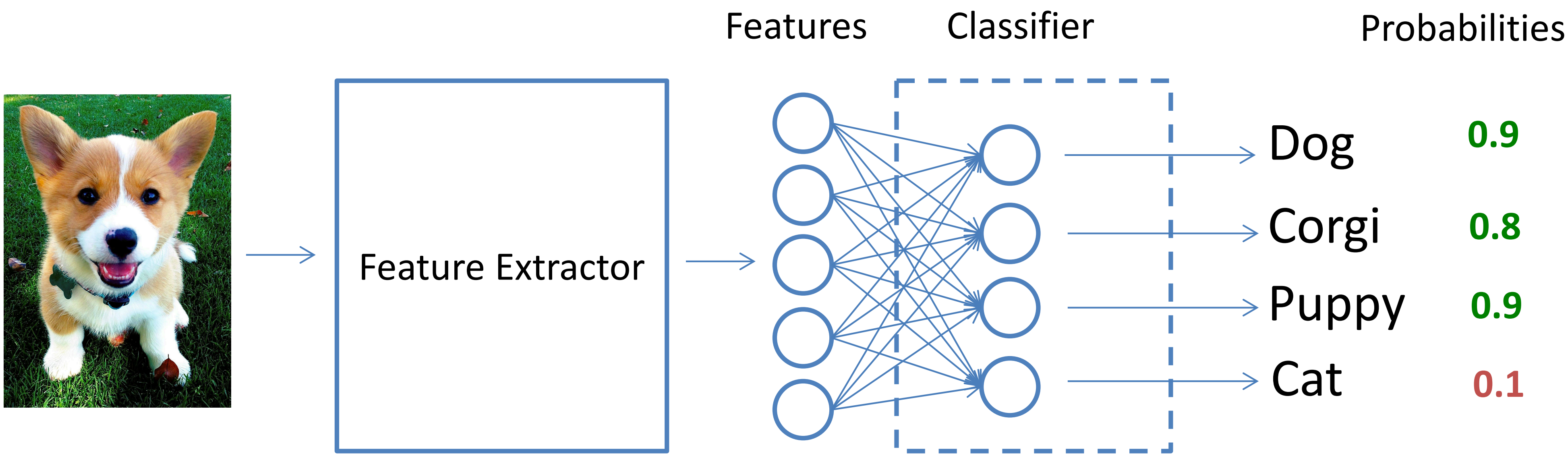
- Assign semantic labels to objects

Probabilities

Dog	0.9
Corgi	0.8
Puppy	0.9
Cat	0.1

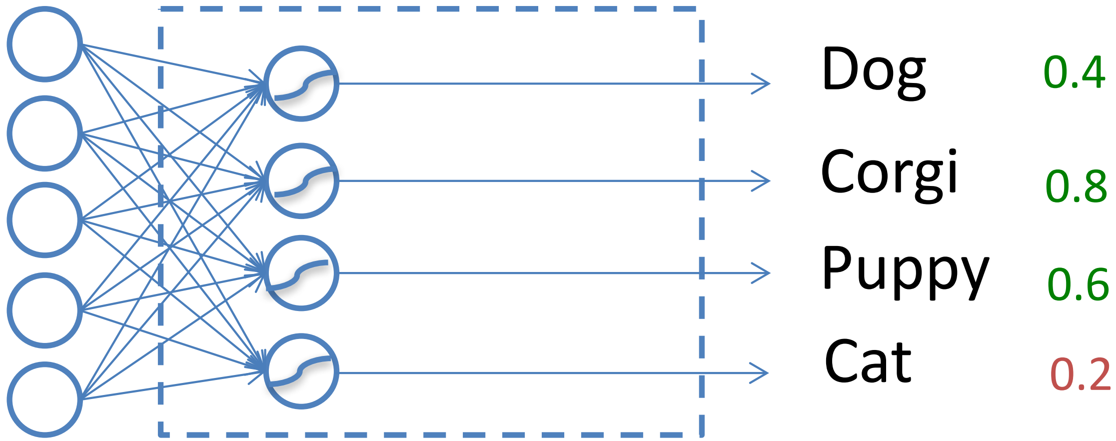
Object Classification

- Assign semantic labels to objects



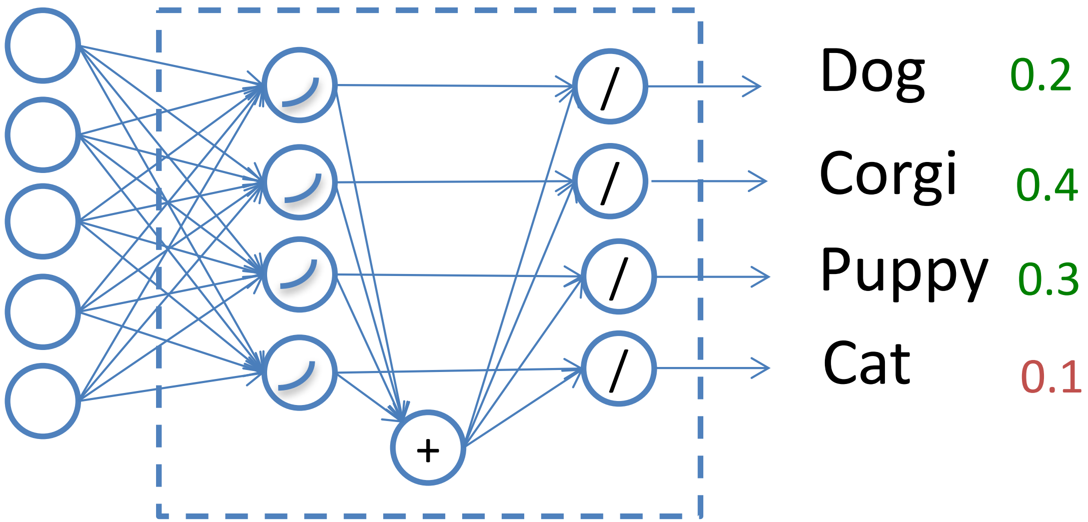
Object Classification

- Independent binary classifiers: Logistic Regression



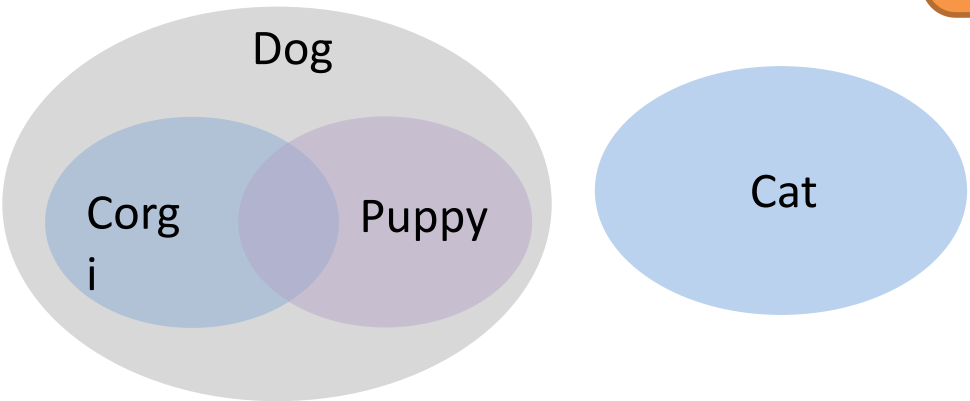
No assumptions about relations.

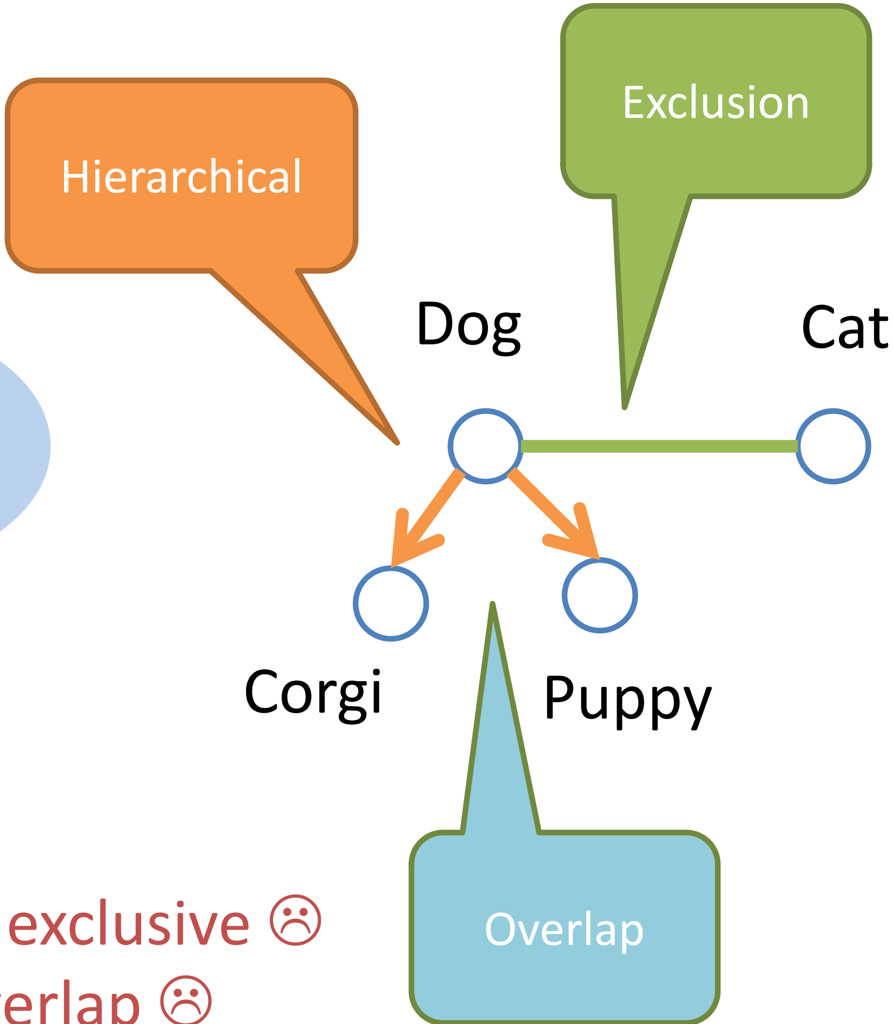
- Multiclass classifier: Softmax



Assumes mutual exclusive labels.

Object labels have rich relations

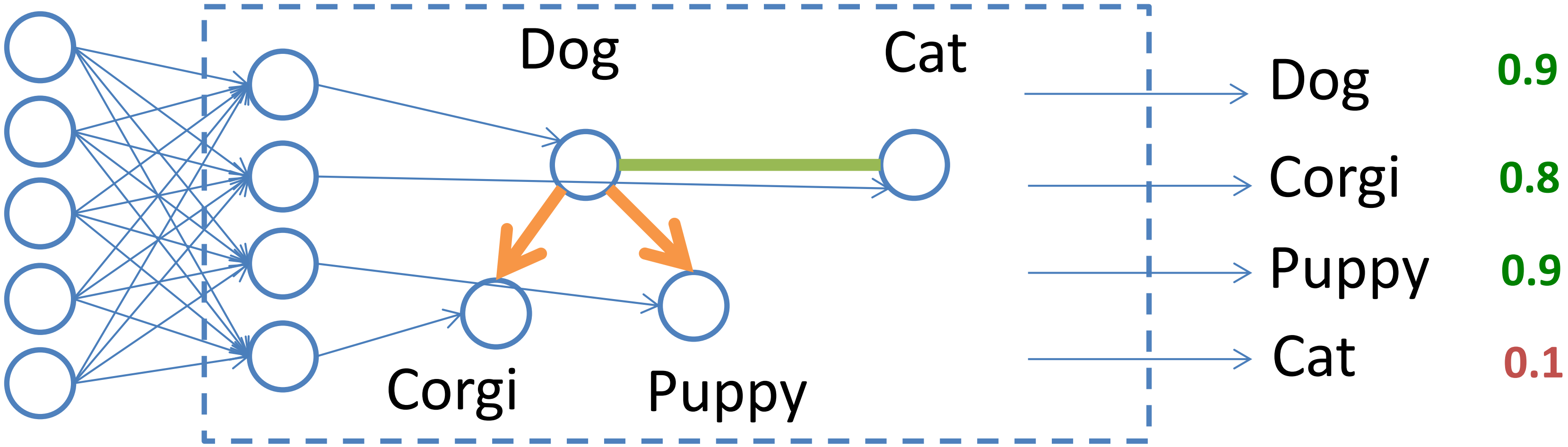




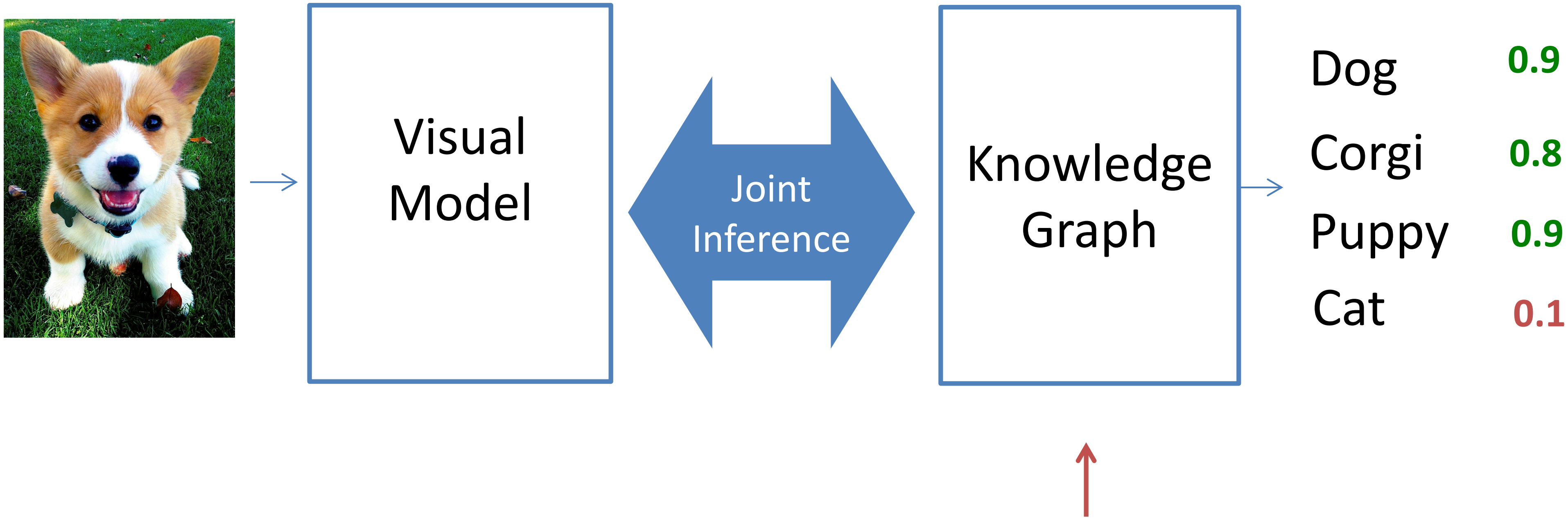
Softmax: all labels are mutually exclusive ☹
Logistic Regression: all labels overlap ☹

Goal: A new classification model

Respects real world label relations



Visual Model + Knowledge Graph



Assumption in this work:
Knowledge graph is given and fixed.

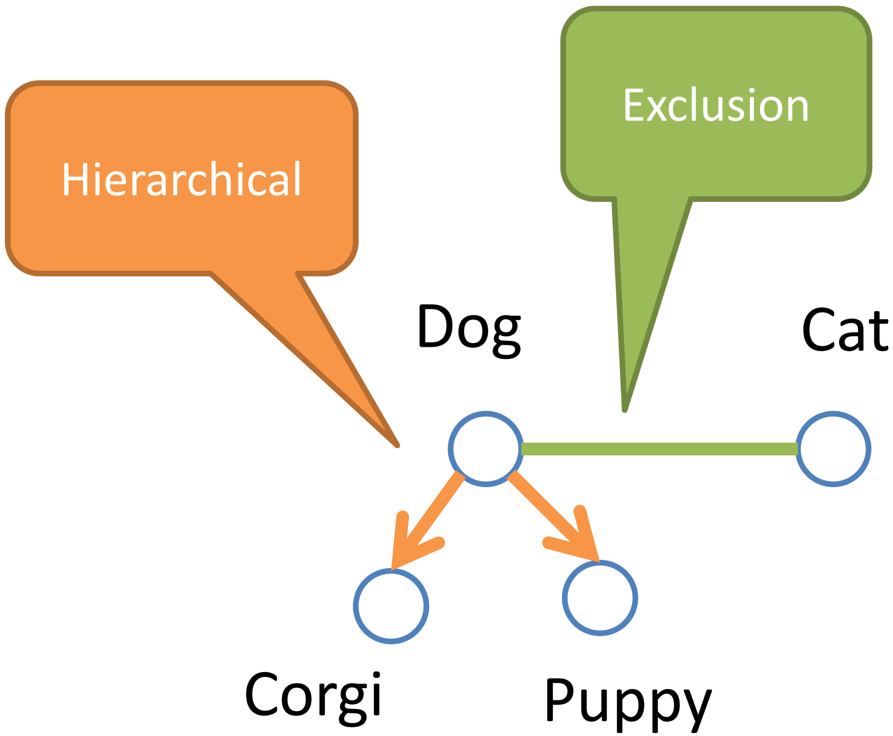
Agenda

- Encoding prior knowledge (HEX graph)
- Classification model
- Efficient Exact Inference

Agenda

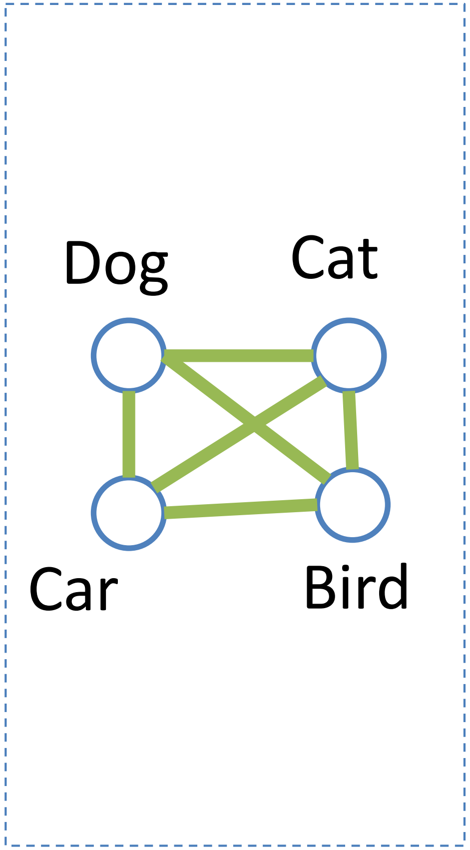
- Encoding prior knowledge (HEX graph)
- Classification model
- Efficient Exact Inference

Hierarchy and Exclusion (HEX) Graph

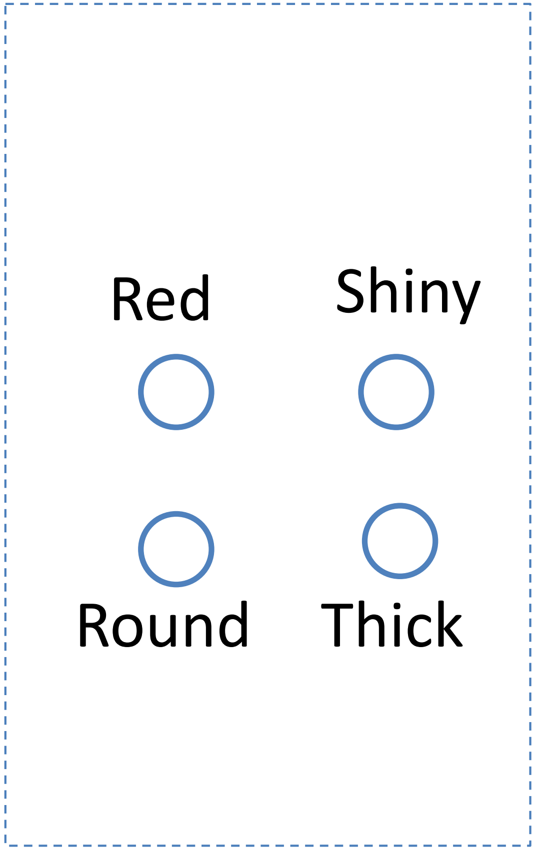


- Hierarchical edges (directed)
- Exclusion edges (undirected)

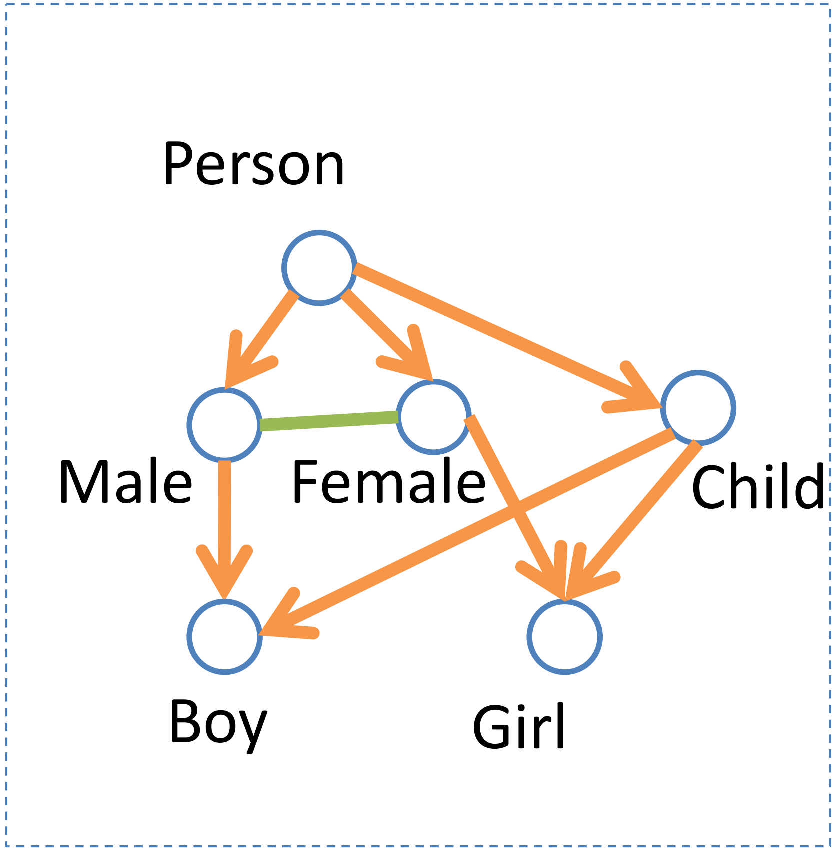
Examples of HEX graphs



Mutually exclusive



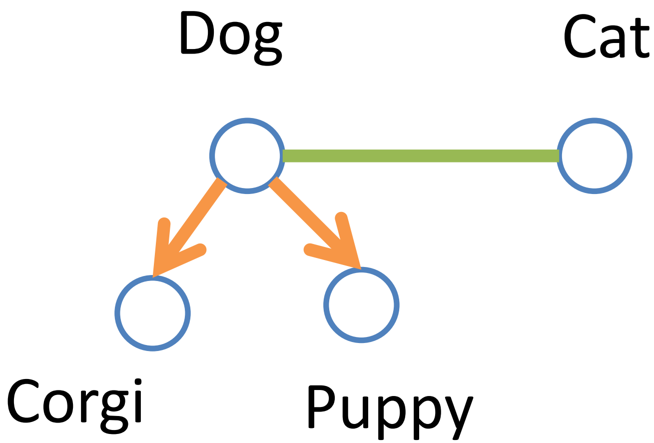
All overlapping



Combination

State Space: Legal label configurations

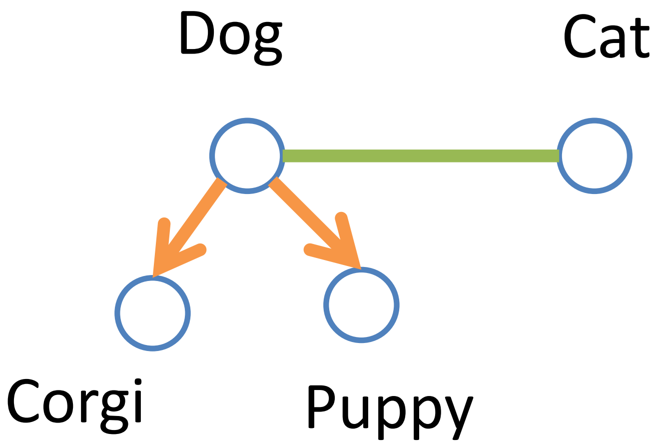
Each edge defines a constraint.



Dog	Cat	Corgi	Puppy
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
1	0	0	0
…			
1	1	0	0
1	1	0	1
…			

State Space: Legal label configurations

Each edge defines a constraint.

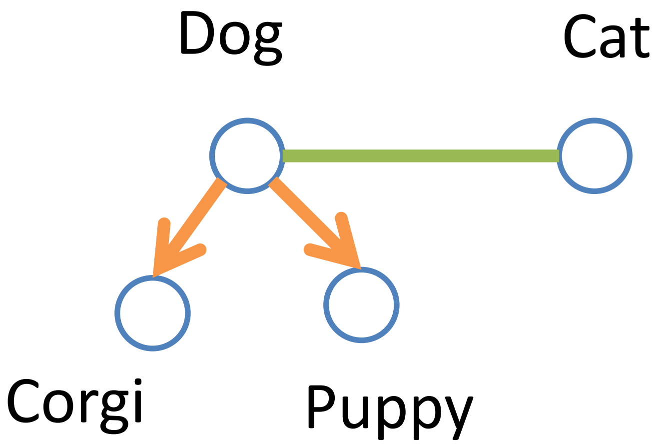


Hierarchy: (dog, corgi) can’t be (0,1)

Dog	Cat	Corgi	Puppy
0	0	0	0
~~0~~	~~0~~	~~0~~	~~1~~
~~0~~	~~0~~	~~1~~	~~0~~
~~0~~	~~0~~	~~1~~	~~1~~
1	0	0	0
…			
1	1	0	0
1	1	0	1
…			

State Space: Legal label configurations

Each edge defines a constraint.



Hierarchy: (dog, corgi) can’t be (0,1)

Exclusion: (dog, cat) can’t be (1,1)

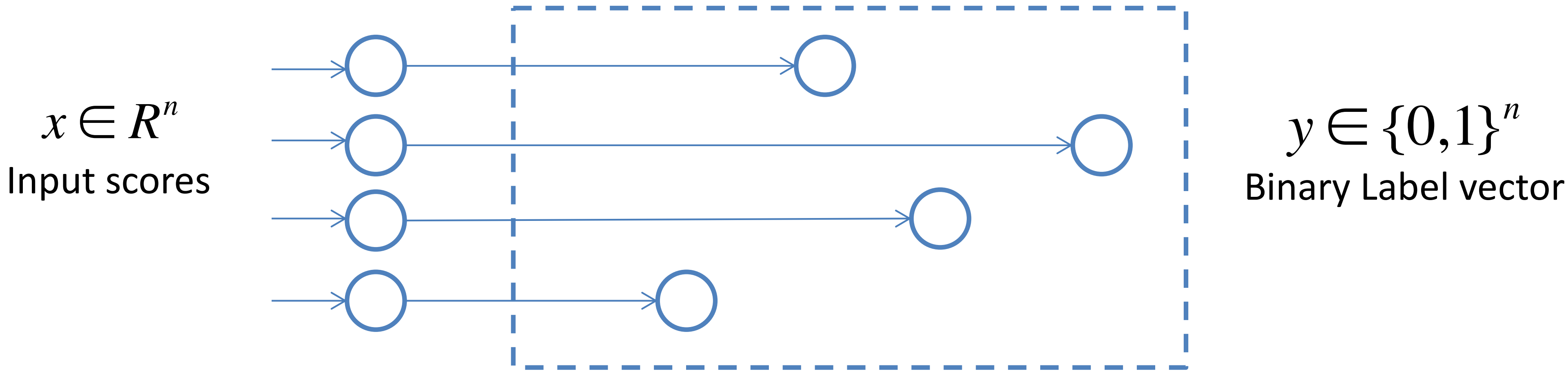
Dog	Cat	Corgi	Puppy
0	0	0	0
~~0~~	~~0~~	~~0~~	~~1~~
~~0~~	~~0~~	~~1~~	~~0~~
~~0~~	~~0~~	~~1~~	~~1~~
1	0	0	0
	…		
~~1~~	~~1~~	~~0~~	~~0~~
~~1~~	~~1~~	~~0~~	~~1~~
	…		

Agenda

- Encoding prior knowledge (HEX graph)
- Classification model
- Efficient Exact Inference

HEX Classification Model

- Pairwise Conditional Random Field (CRF)



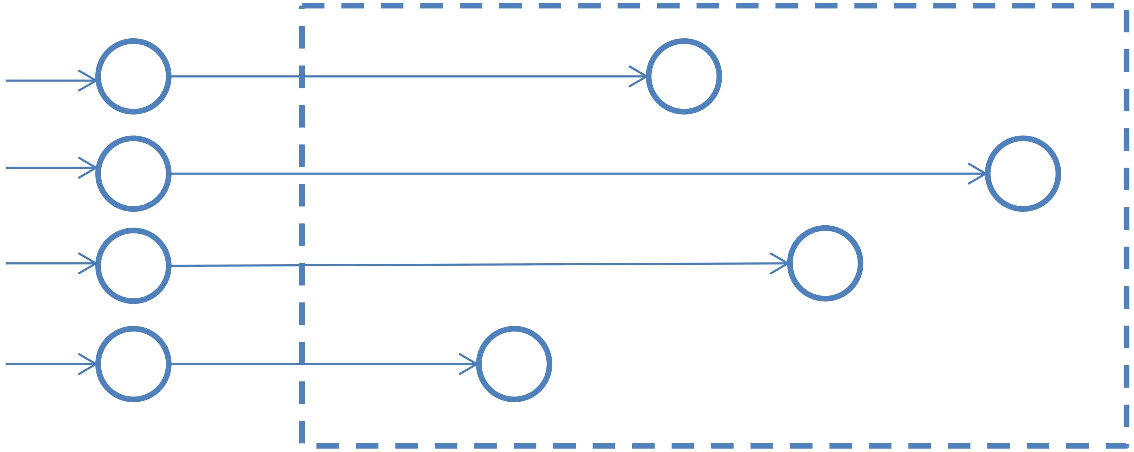
$$\Pr(y \mid x) = \frac{1}{Z(x)} \prod_i \phi_i(x_i, y_i) \prod_{i,j} \psi_{i,j}(y_i, y_j)$$

HEX Classification Model

- Pairwise Conditional Random Field (CRF)

$x \in R^n$

Input scores



$y \in \{0,1\}^n$

Binary Label vector

$$\Pr(y \mid x) = \frac{1}{Z(x)} \prod_i \phi_i(x_i, y_i) \prod_{i,j} \psi_{i,j}(y_i, y_j)$$

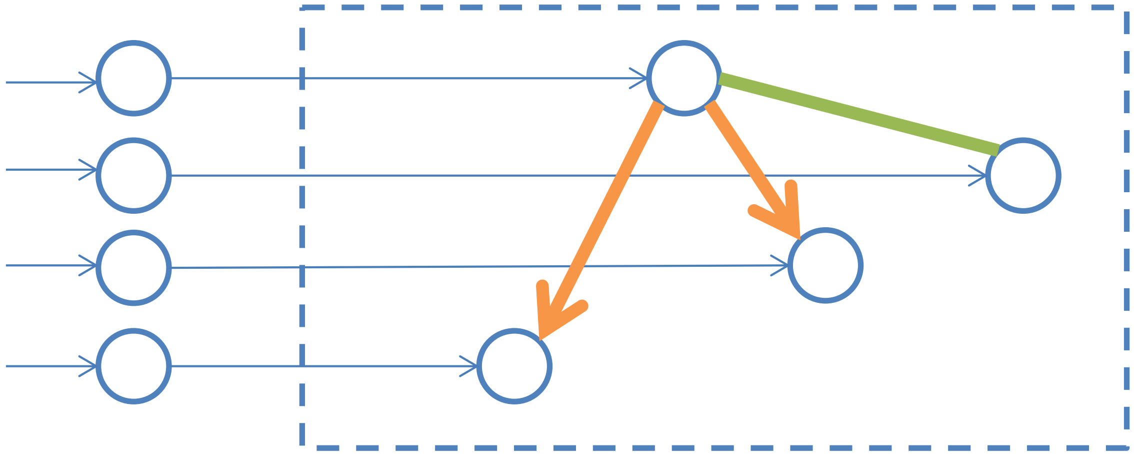
$$\phi_i(x_i, y_i) = \begin{cases} sigmoid(x_i) & if\ y_i = 1 \\ 1 - sigmoid(x_i) & if\ y_i = 0 \end{cases}$$

Unary: same as logistic regression

HEX Classification Model

- Pairwise Conditional Random Field (CRF)

$x \in R^n$
Input scores



$y \in \{0,1\}^n$
Binary Label vector

$$\Pr(y \mid x) = \frac{1}{Z(x)} \prod_i \phi_i(x_i, y_i) \prod_{i,j} \psi_{i,j}(y_i, y_j)$$

$$\phi_i(x_i, y_i) = \begin{cases} sigmoid(x_i) & if\ y_i = 1 \\ 1 - sigmoid(x_i) & if\ y_i = 0 \end{cases}$$

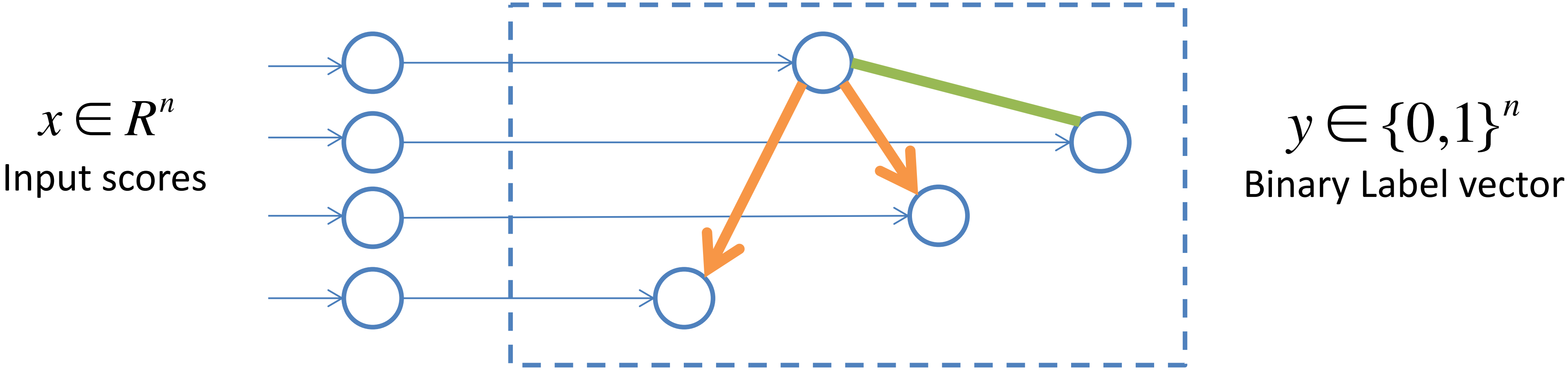
Unary: same as logistic regression

$$\psi_{i,j}(y_i, y_j) = \begin{cases} 0 & \textbf{If violates constraints} \\ 1 & \text{Otherwise} \end{cases}$$

Pairwise: set illegal configuration to zero

HEX Classification Model

- Pairwise Conditional Random Field (CRF)



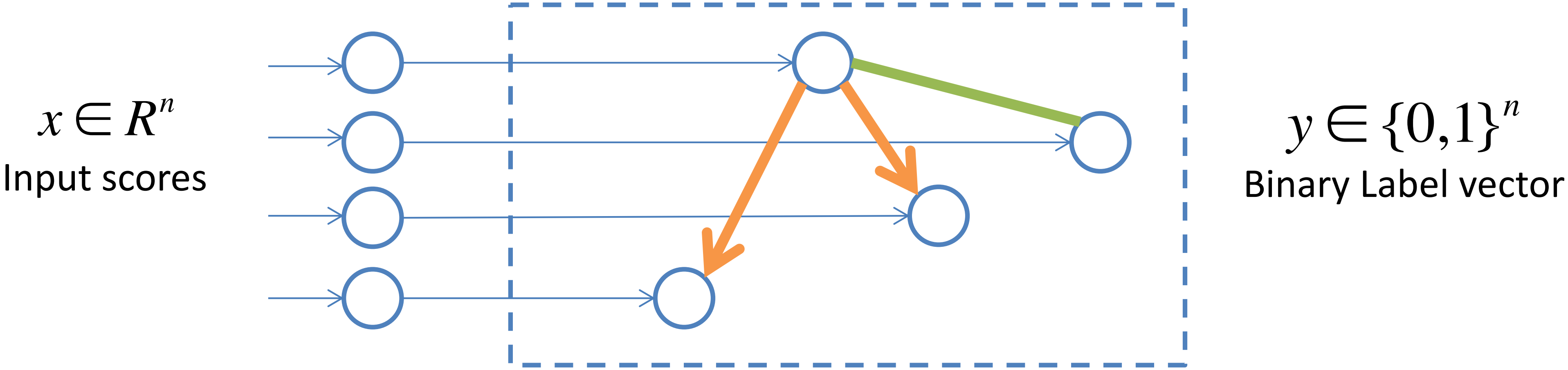
$$\Pr(y \mid x) = \frac{1}{Z(x)} \prod_i \phi_i(x_i, y_i) \prod_{i,j} \psi_{i,j}(y_i, y_j)$$

$$Z(x) = \sum_{\bar{y} \in \{0,1\}^n} \prod_i \phi_i(x_i, \bar{y}_i) \prod_{i,j} \psi_{i,j}(\bar{y}_i, \bar{y}_j)$$

Partition function: Sum over all (legal) configurations

HEX Classification Model

- Pairwise Conditional Random Field (CRF)



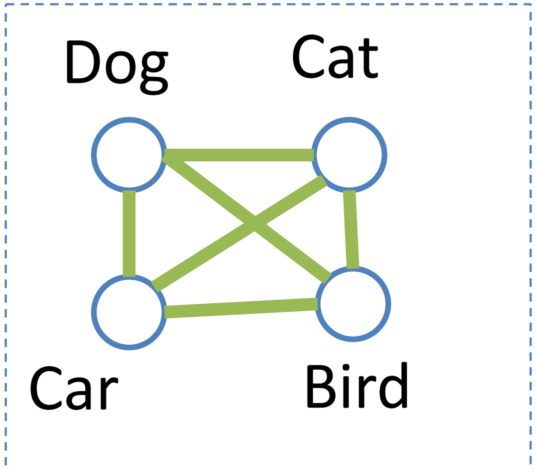
$$\Pr(y \mid x) = \frac{1}{Z(x)} \prod_i \phi_i(x_i, y_i) \prod_{i,j} \psi_{i,j}(y_i, y_j)$$

Probability of a single label: marginalize all other labels.

$$\Pr(y_i = 1 \mid x) = \frac{1}{Z(x)} \sum_{\bar{y}:\bar{y}_i=1} \prod_i \phi_i(x_i, \bar{y}_i) \prod_{i,j} \psi_{i,j}(\bar{y}_i, \bar{y}_j)$$

Special Case of HEX Model

- Softmax



Mutually exclusive

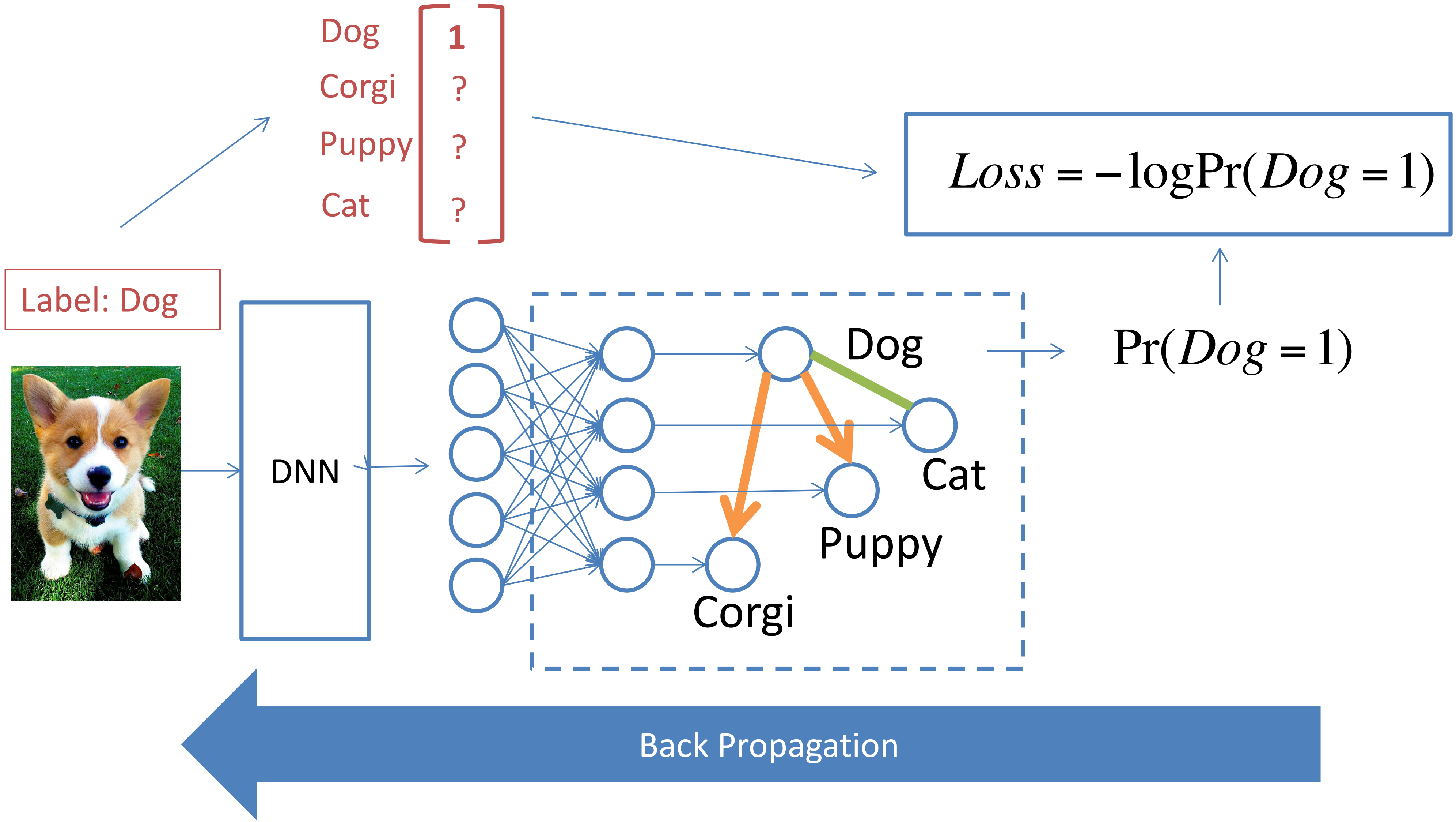
$$\Pr(y_i = 1 \mid x) = \frac{\exp(x_i)}{1 + \sum_j \exp(x_j)}$$

- Logistic Regressions

All overlapping

$$\Pr(y_i = 1 \mid x) = \frac{1}{1 + \exp(-x_i)}$$

Learning



Maximize marginal probability of observed labels

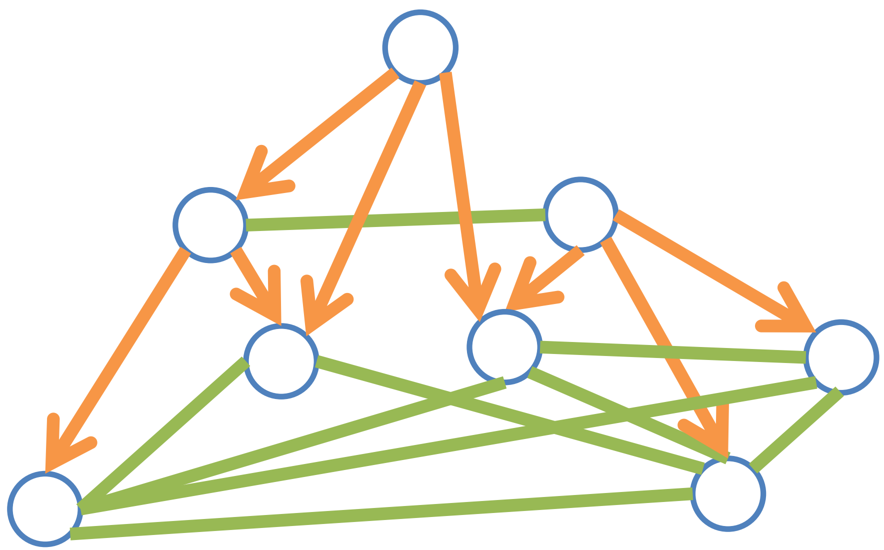
DNN = Deep Neural Network

Agenda

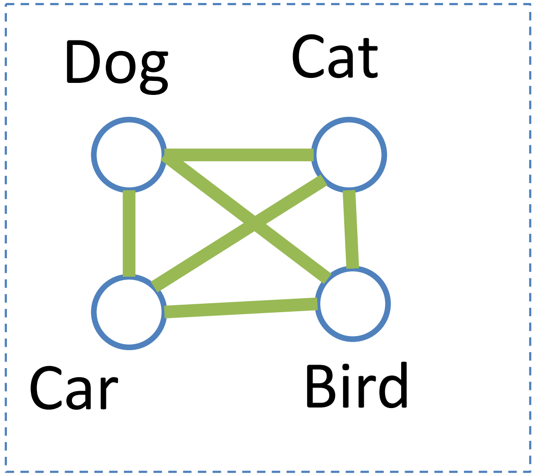
- Encoding prior knowledge (HEX graph)
- Classification model
- Efficient Exact Inference

Naïve Exact Inference is Intractable

- Inference:
  - Computing partition function
  - Perform marginalization
- HEX-CRF can be densely connected (large treewidth)



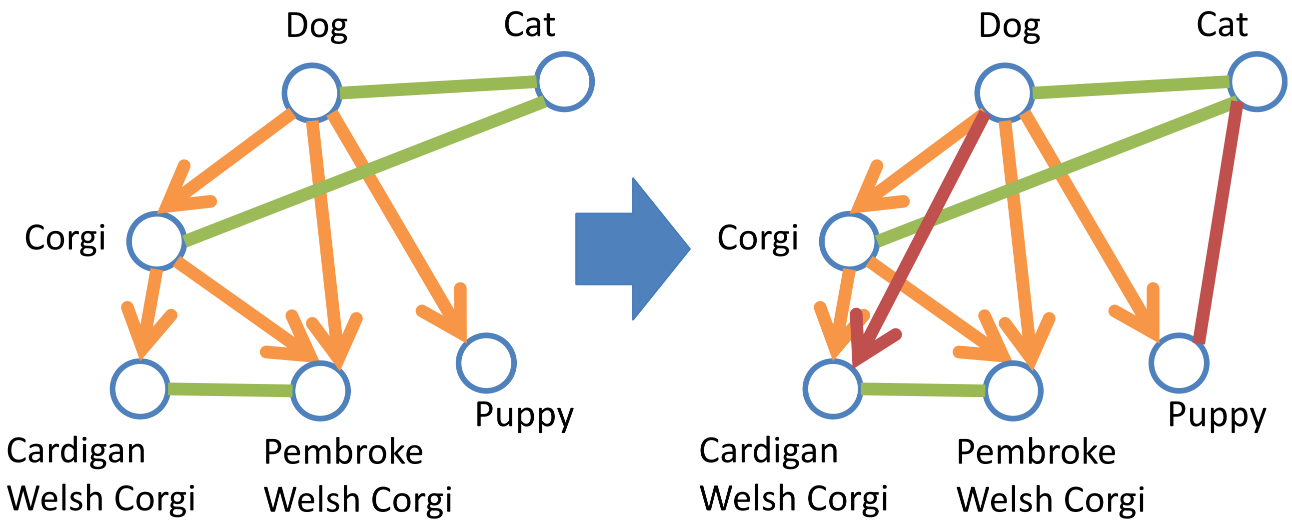
Observation 1: Exclusions are good



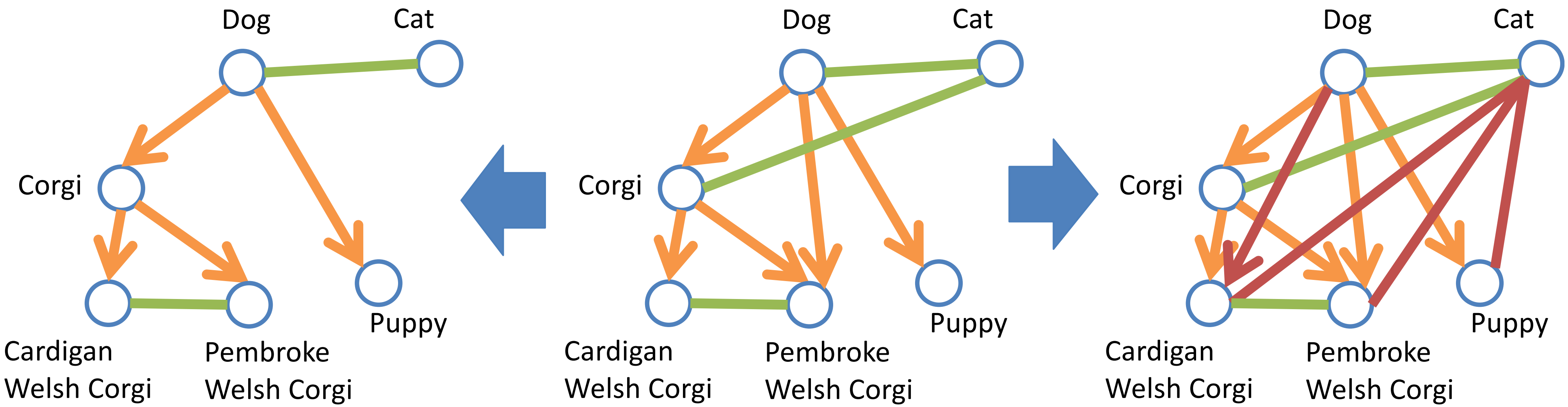
Number of legal states is O(n), not O(2^n).

- Lots of exclusions → Small state space → Efficient inference
- Realistic graphs have lots of exclusions.
- Rigorous analysis in paper.

Observation 2: Equivalent graphs



Observation 2: Equivalent graphs



Sparse equivalent

- Small Treewidth ☺
- Dynamic programming

Dense equivalent

- Prune states ☺
- Can brute force

HEX Graph Inference