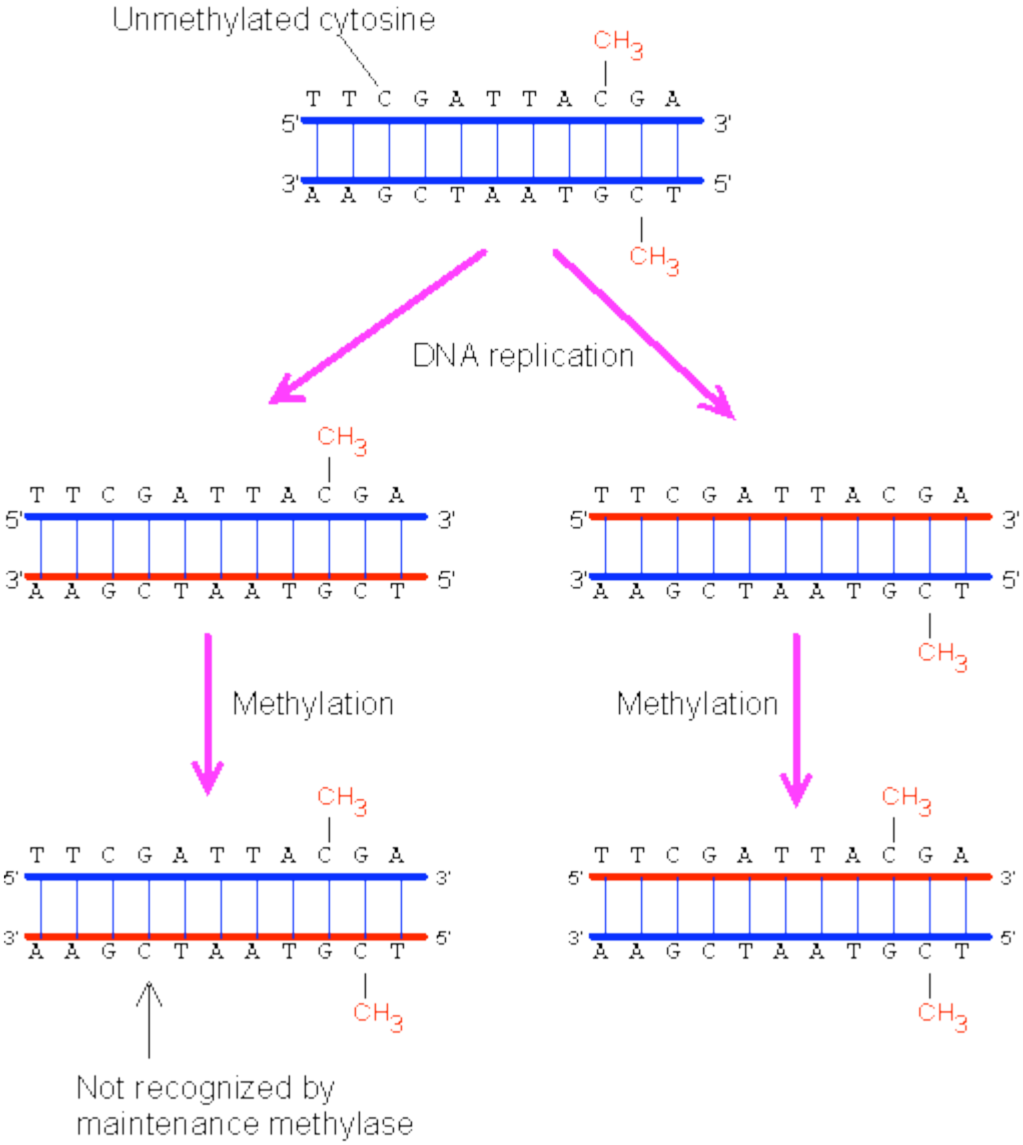
CpG Island Modeling Using Graphical Models

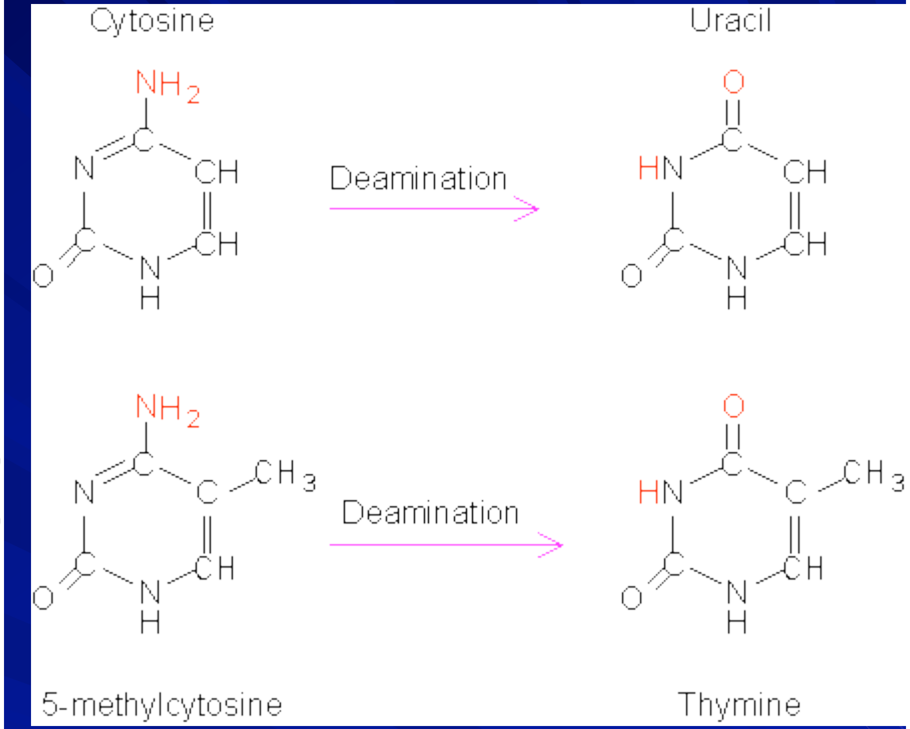
Gang Ji, Tim Hg
Dept. Electrical Engineering

Lingyun Huang
Dept. Bioengineering

CpG Island

- CpG island
  - – Short stretch in DNA with higher frequency of CG sequence
  - – Located around the promoter of house keeping Genes or other genes frequently expressed in a cell
  - – Due to different methylation level in inactive and active genes

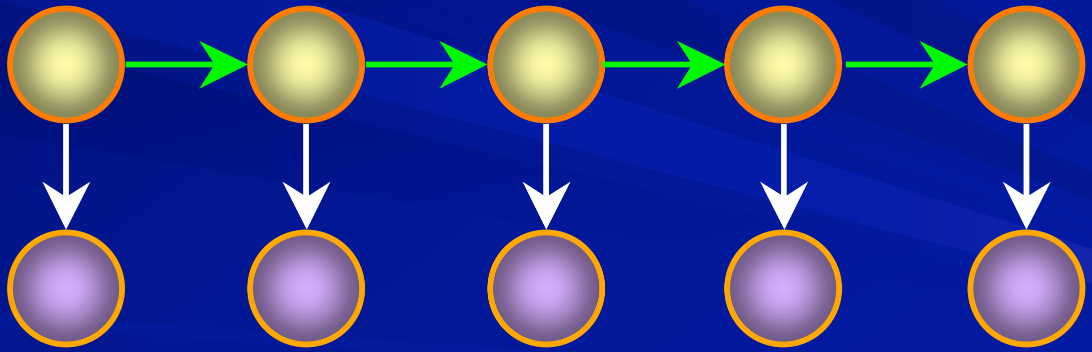




Methylased cytosine

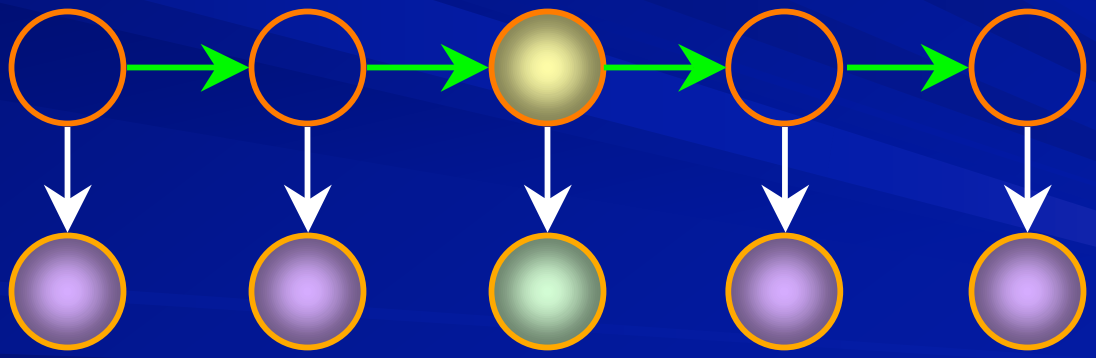
CpG Island Modeling

- Hidden Markov Models
  - States: $A^{+}_{-}, C^{+}_{-}, G^{+}_{-}, T^{+}_{-}$
  - Observations: A, C, G, T



HMMs for CpG

- HMMs are good. But…
  - Conditional independent statements too strong
  - $X_t \perp\!\!\!\perp X_{\hat{t}} \mid S_t$



HMMs for CpG

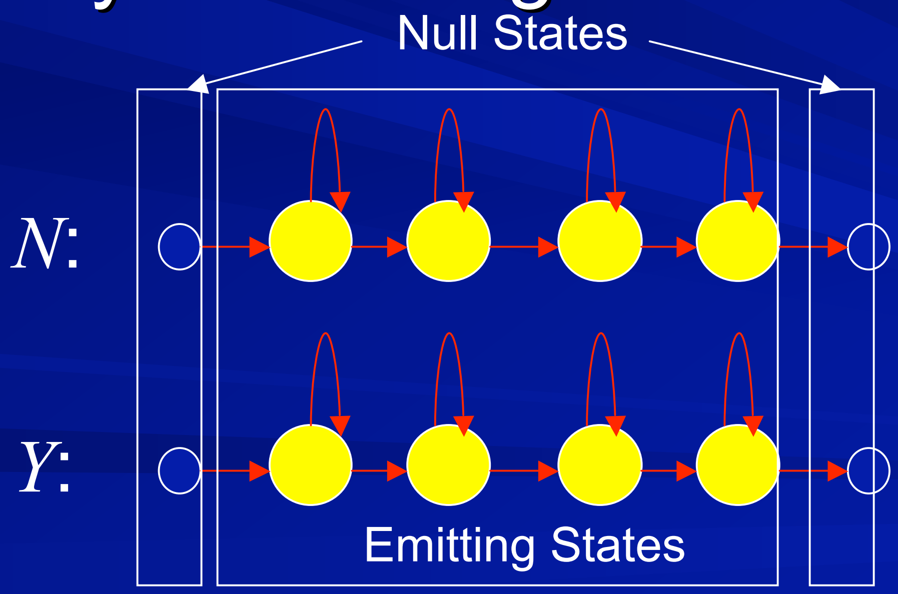
- HMMs are good. But…
  - Duration Modeling
    - State occupancy decreases exponentially with time: $d_i(t) = a_{ii}^t(1 - a_{ii})$ → poor duration modeling
  - Conditional independent statements too strong $X_t \perp\!\!\!\perp X_{\hat{t}} \mid S_t$
    - Hard to effectively handle non-stationary observations that are highly correlated.

Our Proposed Improvements

- Language models
- Change the structure of graph
- Other graphical families (MRFs)

Topology of the HMMs

- Two HMMs were used:
  - *N*: non-island
  - *Y*: island
- Strictly Left-to-Right HMMs:



Null States

N:

Y:

Emitting States

HMM Training using HTK

- Training Data:
  - Discrete Observations: Mapped in indices
    - Discrete HMMs
  - With Model Alignment:
    - Performed Baum-Welch training within the model:
      - Since only the state sequences are hidden

Decoding using HMM and Language Model

- the Cost Function:

$$\hat{W} = \arg\max_{W} P(W \mid O) = \arg\max_{W} \frac{P(W)P(O \mid W)}{P(O)}$$

$$= \arg\max_{W} P(W)P(O \mid W)$$

$$= \arg\max_{W}[\underbrace{\log P(W)}_{\text{Language Model}} + \underbrace{\log P(O \mid W)}_{\text{HMM}}]$$

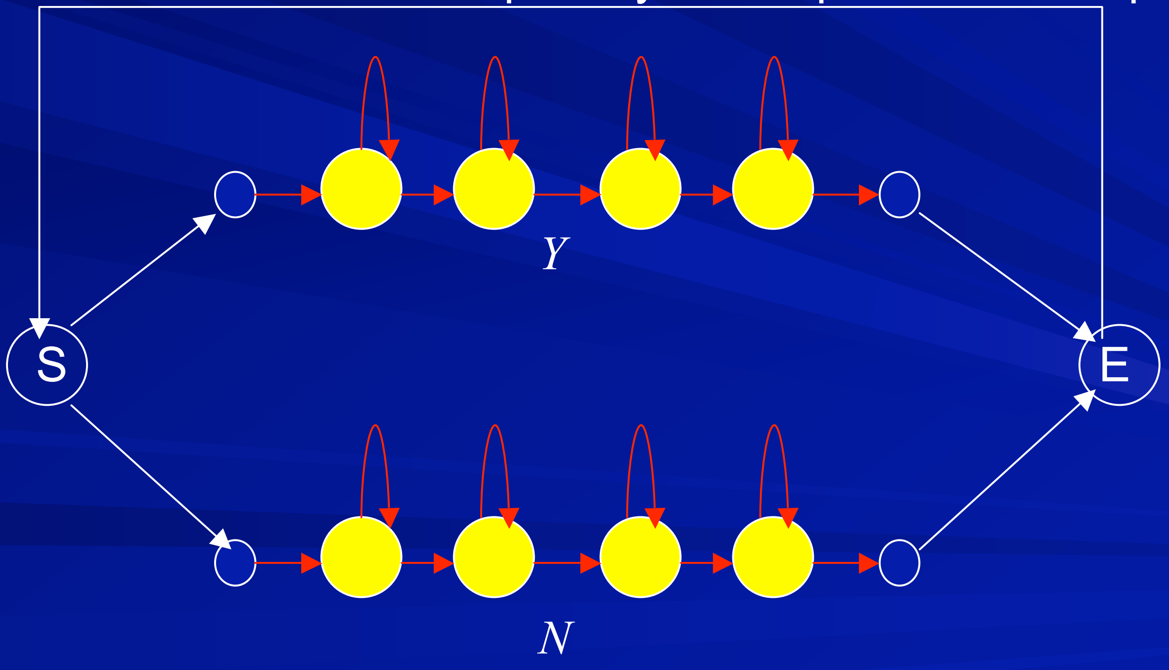
- Issues
  - $P(O|W)$ is usually underestimated due to the fallacy of the Markov and independence assumptions. →give the language model too little weight.
  - Introduce language model weight (LW) to balance the two probability quantities.
    - Usually $LW > 1.0$ and it is task dependent
  - The Cost Function becomes:

$$\hat{W} = \arg\max_{W}[LW * \log P(W) + \log P(O|W)]$$

Decoding Framework

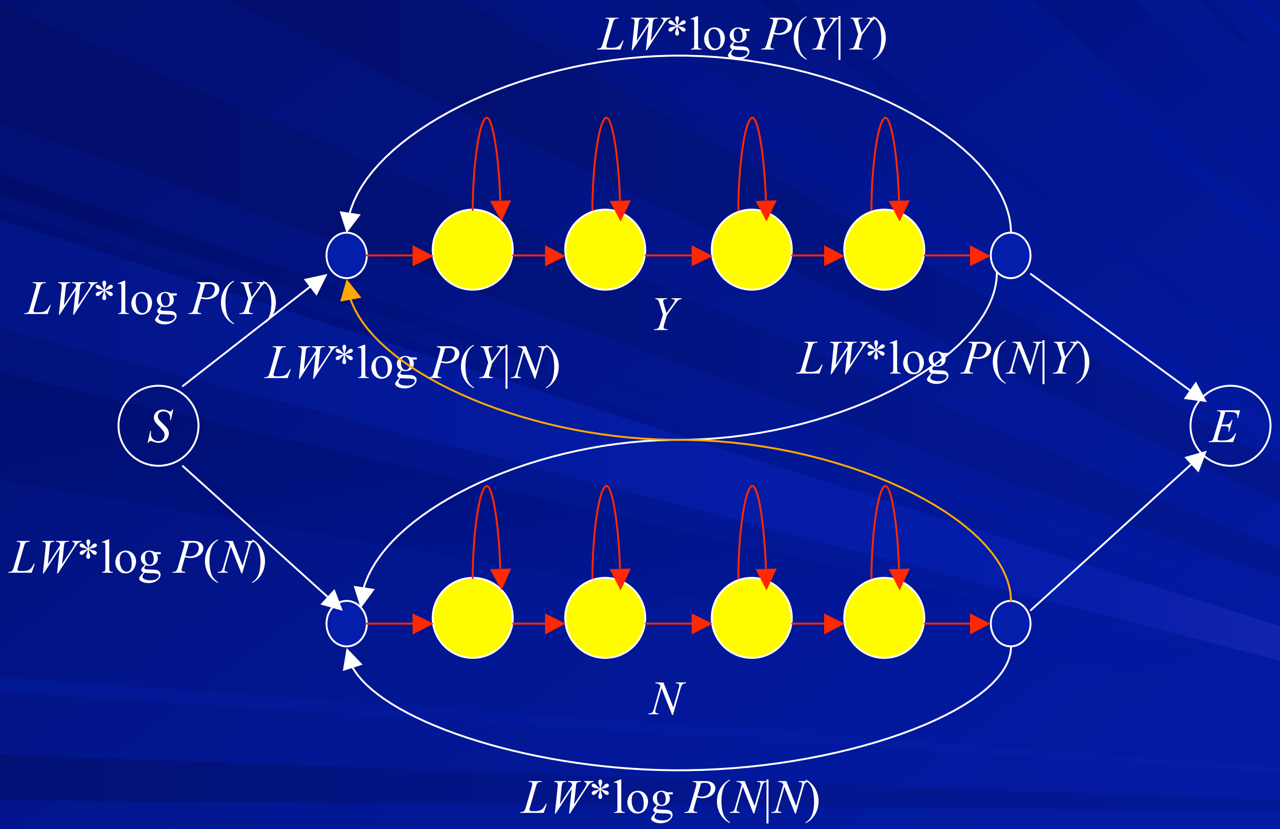
- No language model
  - Assuming all sequences are equally likely

WP: word penalty to compensate HMM prob.



Decoding with bigram LM

- $P(W) \cong P(w_1)*P(w_2|w_1)*\mathrm{P}(w_3|w_2)\ldots\mathrm{P}(w_n|w_{n-1})$



Evaluation Corpus

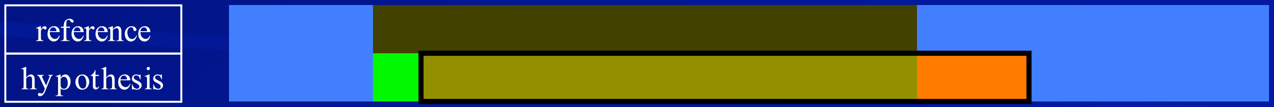
- Gene sequence
  - – EMBL, European Bioinformatics Institute
- CpG island alignment
  - – European Bioinformatics Institute
- We used
  - – Whole corpus: 1710 sq.
  - – Training: 1539 sq.
  - – Testing: 171 sq.

Corpus Statistics

	CpG island subsequence	DNA sequence
Maximum	3340	185775
Minimum	181	44
Mean value	465	3787

Evaluation Metric

- No standard quantitative metric
- Precision/Recall
  - Precision
    - *P*: True positive / all hypothesized truth
  - Recall
    - *R*: True positive / all truth



Evaluation Metric

- No standard quantitative metric
- Precision/Recall
  - Precision
    - P: True positive / all hypothesized truth
  - Recall
    - R: True positive / all truth
  - F score (when no free parameter)
    - Harmonic mean of precision and recall

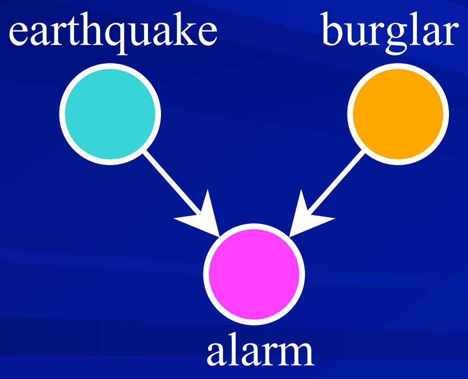
$$\frac{1}{F} = \frac{1}{P} + \frac{1}{R}$$

Language Model Results

	Precision	Recall	F Measure
Baseline	29.5%	77.7%	0.214
LM bigram	36.3%	75.0%	0.245

Graphical Models

- Graphical Models
  - – Nodes: random variables
  - – Edges: encodes conditional independent statements



$P(a \mid e)$

$P(a \mid b)$

$P(e)$

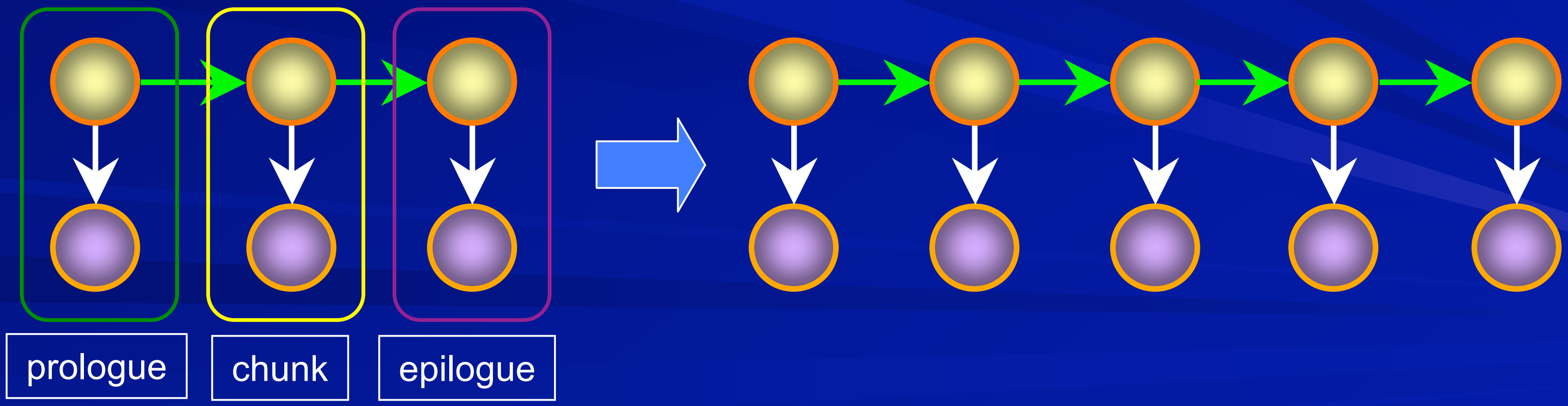
$P(b)$

Graphical Models

- Different graphical models
  - Directed: Bayesian networks
  - Undirected: Markov random fields
  - Mixture of the two
- Next work
  - Dynamic Bayesian networks (DBNs)
  - Conditional random fields (CRFs)

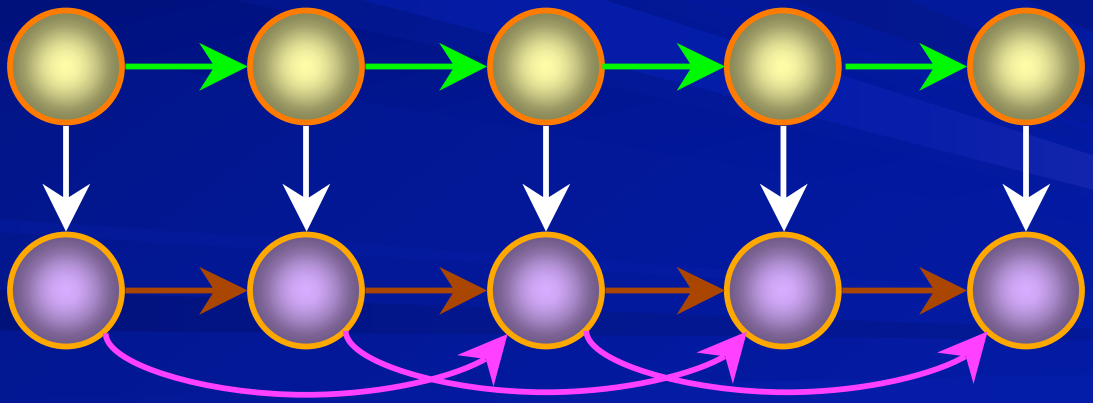
DBNs

- Dynamics Bayesian networks
  - Directed graphical model
  - Prologue/chunk/epilogue
  - Unroll to fit series
  - HMM is a DBN



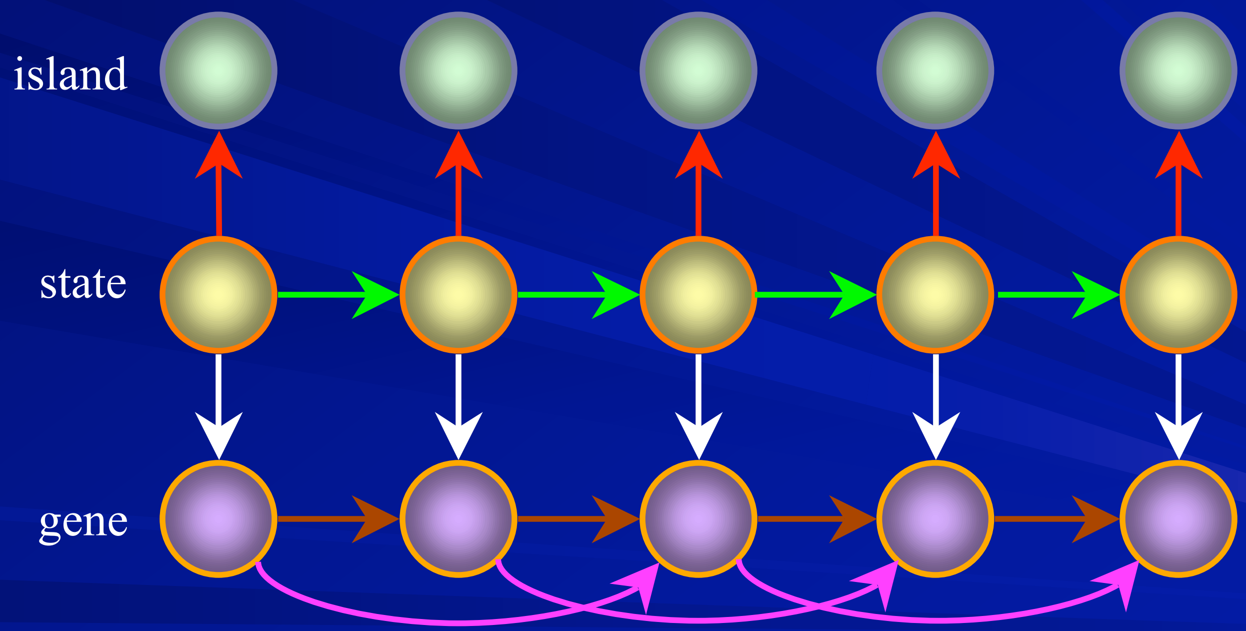
Our DBN Models

- Recall
  - HMM CI statements too strong
  - Idea: add dependencies in gene sequences
  - 8 hidden states



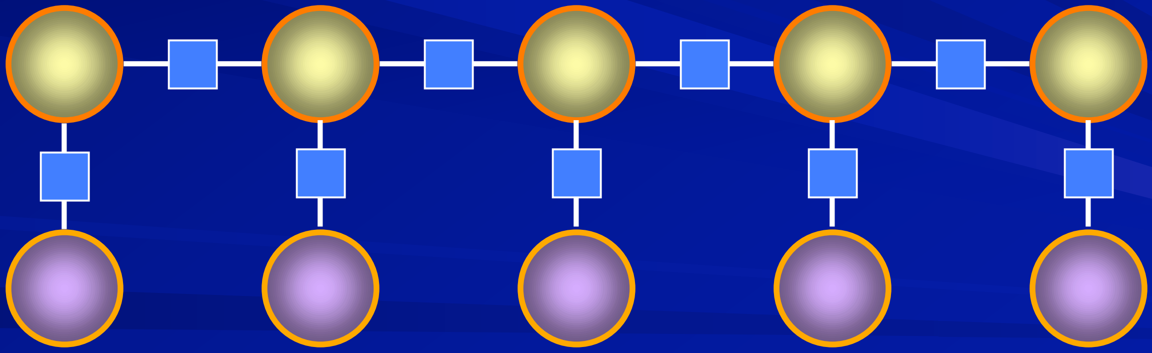
Training

- Standard EM learning



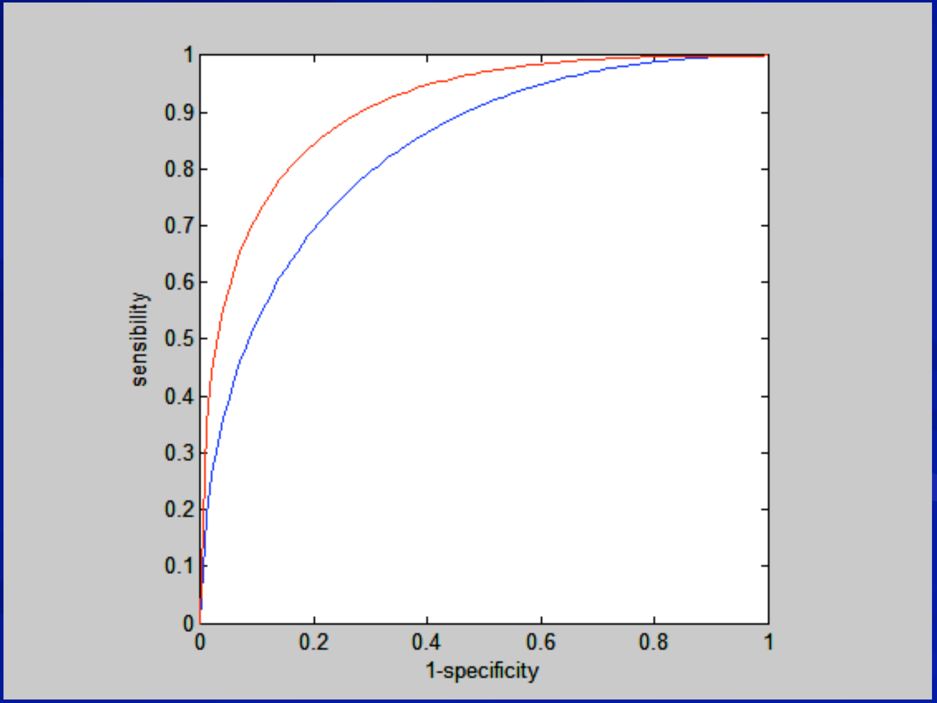
Decoding

- Junction tree algorithm
  - – Form junction tree from the graph
  - – Message passing along the tree
  - – Viterbi assumption

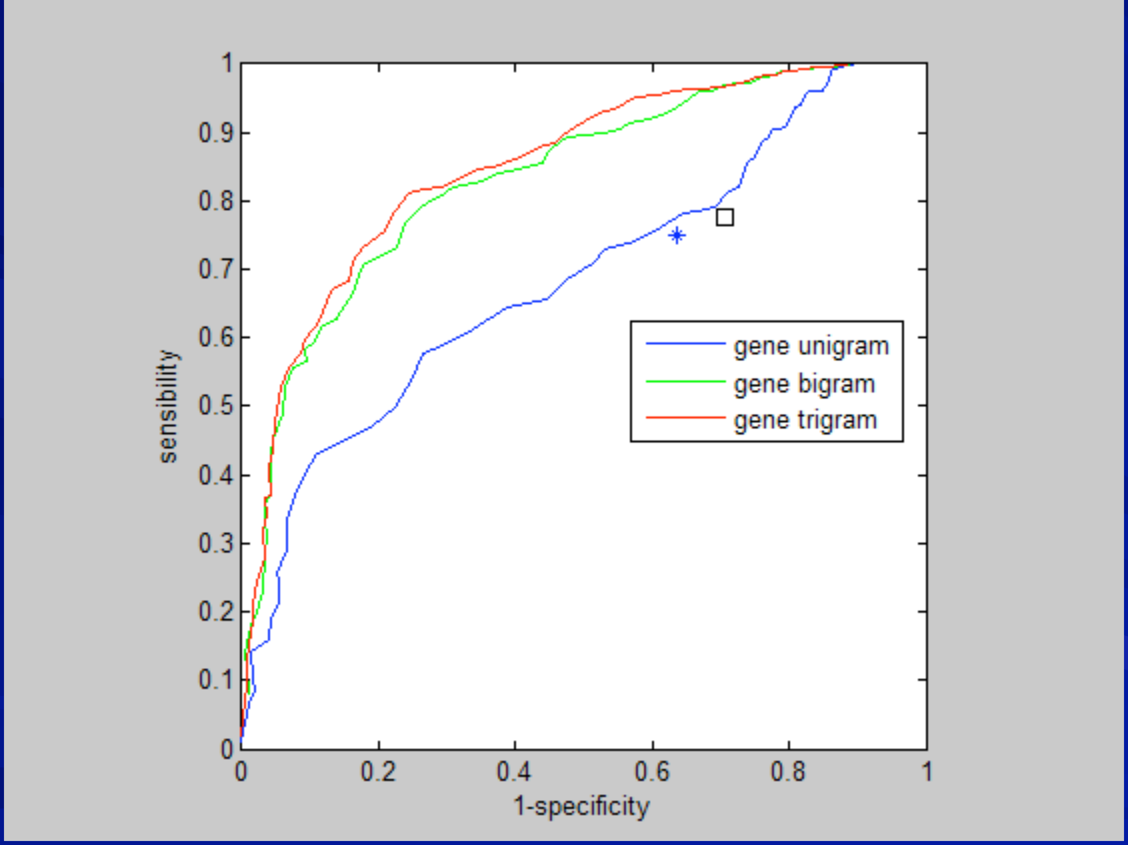


ROC Curves

- Receiver operating characteristic curves
  - – Free parameter to tune between precision and recall



DBN Results



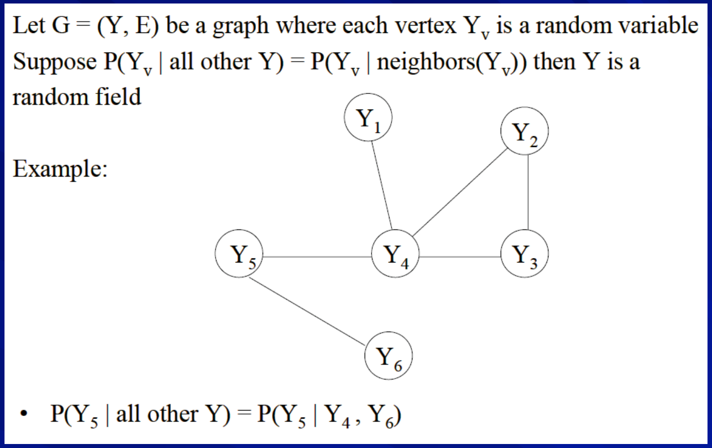
sensibility
1-specificity
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1
gene unigram
gene bigram
gene trigram

DBN Conclusions

- Conclusions
  - Adding links between gene observations helps a lot
  - Equal error rates

model	EER	rel. imp.
baseline	38.8%	-
bigram	25.5%	34.3%
trigram	22.4%	42.3%

- Conditional random field for labeling sequence
  - An undirected acyclic graph
  - Random field



Let G = (Y, E) be a graph where each vertex Y_v is a random variable Suppose $P(Y_v \mid \text{all other } Y) = P(Y_v \mid \text{neighbors}(Y_v))$ then Y is a random field

Example:

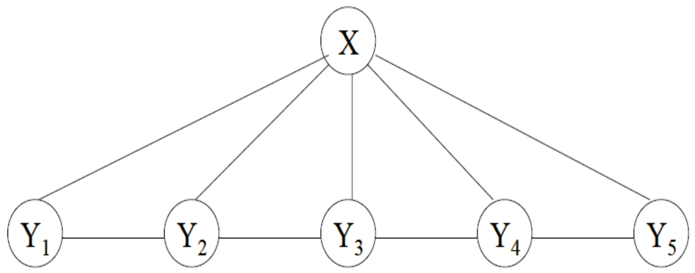
- $P(Y_5 \mid \text{all other } Y) = P(Y_5 \mid Y_4, Y_6)$

Laffetry et.al 2001

  - Definition: for X is a random variable over observation sequence and Y is a random variable over state sequence.

(X,Y) forms a conditional random field

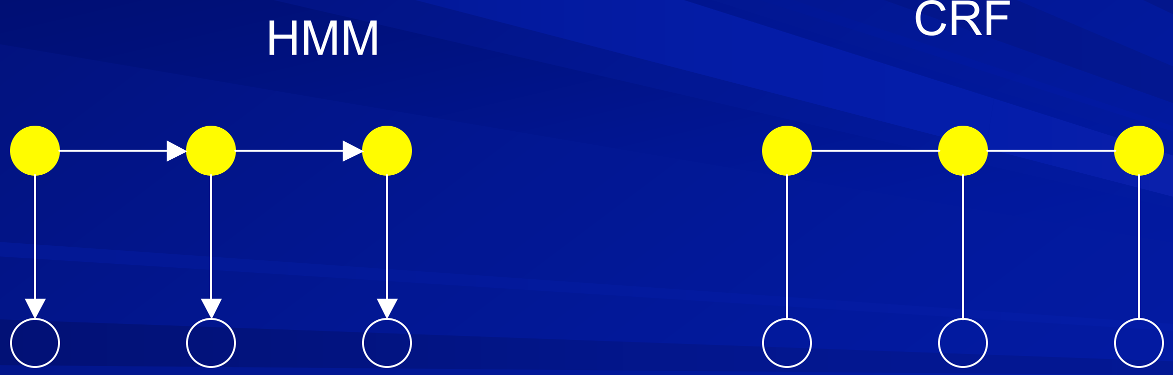
Conditional random field (CRF) example



- $P(Y_3 \mid X \text{, all other } Y) = P(Y_3 \mid X, Y_2, Y_4)$
- Think of X as observations and Y as labels

Laffetry et.al 2001

- Comparison between CRF and HMM



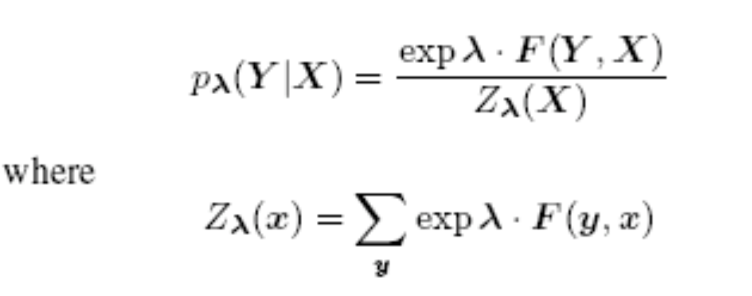
- Probabilistic Models of CRF
  - – Local features of CRF is specified by a vector *f* including

state feature

transition feature

Global feature *F(y,x)*

Conditional probability distribution defined by the CRF

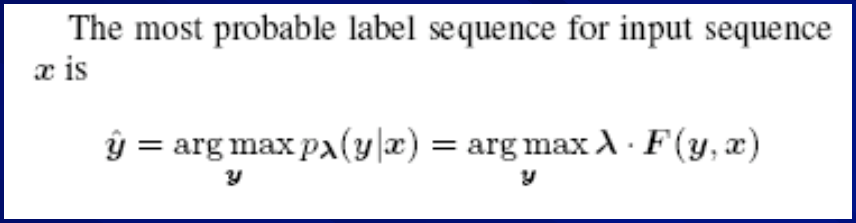


$$p_{\boldsymbol{\lambda}}(Y|X) = \frac{\exp \boldsymbol{\lambda} \cdot \boldsymbol{F}(Y, X)}{Z_{\boldsymbol{\lambda}}(X)}$$

where

$$Z_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp \boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$$

- Decoding by CRF

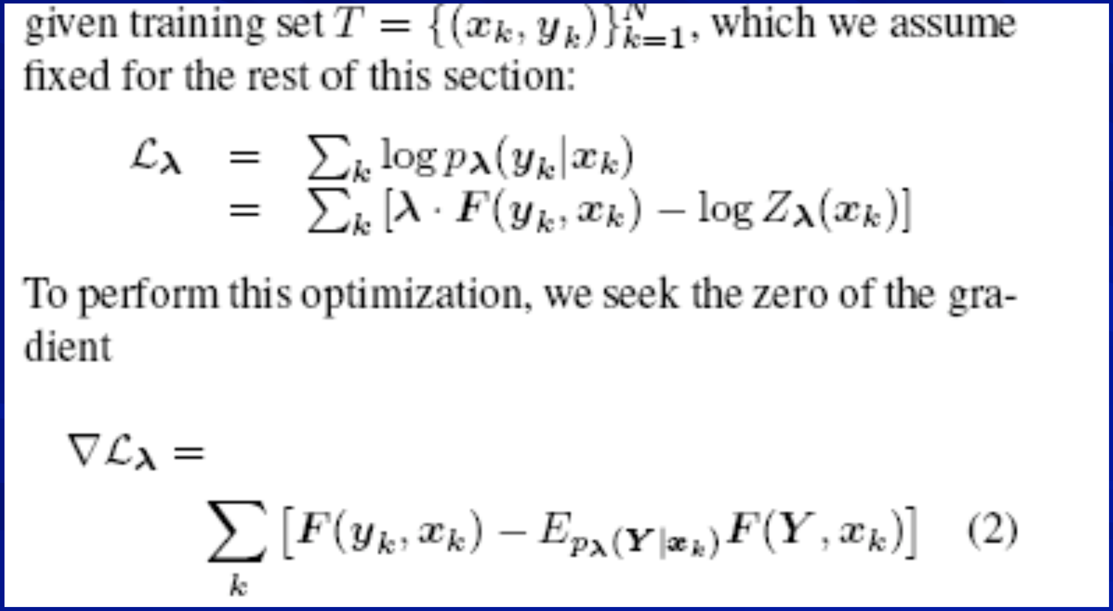


The most probable label sequence for input sequence x is

$$\hat{y} = \arg\max_{\boldsymbol{y}} p_{\boldsymbol{\lambda}}(\boldsymbol{y}|\boldsymbol{x}) = \arg\max_{\boldsymbol{y}} \boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$$

The algorithm is also Viterbi

- Training of CRF
  - Generalized iterative scaling



given training set $T = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^{N}$, which we assume fixed for the rest of this section:

$$\begin{aligned} \mathcal{L}_{\boldsymbol{\lambda}} &= \textstyle\sum_k \log p_{\boldsymbol{\lambda}}(\boldsymbol{y}_k|\boldsymbol{x}_k) \\ &= \textstyle\sum_k [\boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}_k, \boldsymbol{x}_k) - \log Z_{\boldsymbol{\lambda}}(\boldsymbol{x}_k)] \end{aligned}$$

To perform this optimization, we seek the zero of the gradient

$$\nabla \mathcal{L}_{\boldsymbol{\lambda}} = \sum_k \left[\boldsymbol{F}(\boldsymbol{y}_k, \boldsymbol{x}_k) - E_{p_{\boldsymbol{\lambda}}(\boldsymbol{Y}|\boldsymbol{x}_k)} \boldsymbol{F}(\boldsymbol{Y}, \boldsymbol{x}_k) \right] \quad (2)$$

Fei Sha et.al 2003

- In the project
  - Training data
    - Long sequence was truncated every 100 bits to get non-CpG island or CpG island sub-sequences labeled with 1 (non-CpG island) and 2 (CpG island) respectively.
  - Testing data
    - The whole sequence as input
    - Truncated sub-sequences as input

- Software
  - A CRF toolkit in Java from http://crf.sourceforge.net by Dr. Sunita Sarawagi in IIT Bombay
- Result
  - Disappointed, it DID NOT pick up any CpG island
- The possible reason
  - Truncated strategy does not fit the tool
  - Unfamiliar with the source code