CS 3750 Advanced Machine Learning

Latent Variable Generative Models II

Ahmad Diab
AHD23@cs.pitt.edu
Feb 4, 2020

Based on slides of Professor Milos Hauskrecht

Outline

- Latent Variable Generative Models
- Cooperative Vector Quantizer Model
  - Model Formulation
  - Expectation Maximization (EM)
  - Variational Approximation
- Noisy-OR Component Analyzer
  - Model Formulation
  - Variational EM for NOCA
- References

Latent Variable Generative Models

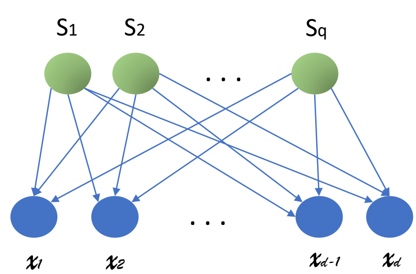
- Generative Models: Unsupervised learning models that study the underlying structure (e.g. interesting patterns) and causal structures of data to generate data like it.
- Latent (hidden) variables are random variables that are hard to observe. (ex. Length is measured, but intelligence is not), and is assumed to affect the response variable.
- The idea: introduce an unobserved latent variable, S, and use it to generate a traceable, less complex distribution.

p(x) → p(x, s) = p(x | s) p(s)

Complex Distribution → Simpler Distribution

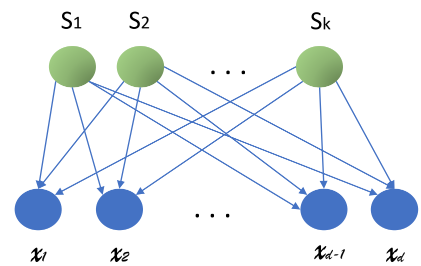
Latent Variable Generative Models

- Assumption: Observable variables are independent given latent variables.

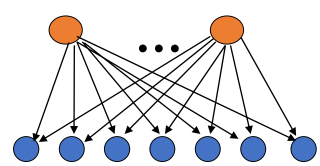


Cooperative Vector Quantizer (CVQ)

- Latent variables (s): Binary vars with Dimensionality *k*
- Observed variables (x): real valued vars Dimensionality *d*



CVQ – Model Description



- Model
  - $x = \sum_{k=1}^{K} s_k w_k + \epsilon$
- Latent variables s_i
  - ~ Bernoulli distribution parameter: π_i
  - $P(s_i \mid \pi_i) = {\pi_i}^{s_i} (1 - \pi_i)^{1-s_i}$
  - w_k is the weight output by source s_k
- Observable variables x
  - ~ Normal distributions parameters: W, Σ
  - $P(x \mid s) = N(Ws, \Sigma)$,
  - we assume $\Sigma = \sigma I$

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & .. & w_{1k} \\ w_{21} & & & \\ & .. & & \\ w_{d1} & .. & .. & w_{dk} \end{pmatrix}$$

- Joint for one instance of s and x
  - $p(x, s \mid \Theta) = 2^{-d/2} \sigma^{-d/2} \exp\{-\frac{1}{2\sigma^2}(x - Ws)^T (x - Ws) \prod_{i=1}^{k} {\pi_i}^{s_i} (1 - \pi_i)^{1-s_i}$

6

CVQ – Model Description

- Objective: to learn parameters of the model: W, π, σ
- If both x and s are observable,
  - Use loglikelihood:

$$\sum_{n=1}^{N} logP(x^{(n)}, s^{(n)}|\Theta) =$$

$$\sum_{n=1}^{N} -d\ log\ \sigma \quad - \frac{1}{2\sigma^2}(x^{(n)} - Ws^{(n)})^T(x^{(n)} - Ws^{(n)}) + \sum_{i=1}^{k} s_i^{(n)}\ log\ \pi_i$$
$$(1 - s_i^{(n)})\log(1 - \pi_i) + \mathrm{c}$$

  - Solution is nice and easy

7

CVQ – Model Description

- Objective: to learn parameters of the model: W, π, σ
- If only x are observable
  - Log likelihood of data:

$$logP(D|\Theta) = \sum_{n=1}^{N} logP(x^{(n)}\ |\Theta) = \sum_{n=1}^{N} log \sum_{\{s^n\}} P\ (x^{(n)},\ \ s^{(n)}|\Theta)$$

  - Solution is hard, we can no longer benefit from the decomposition.
  - Use Expectation Maximization (EM).

8

Expectation Maximization (EM)

- Let H be a set of all variables with hidden or missing values
  - $P(H, D|\Theta, \xi) = P(H\ |D, \Theta, \xi)P(D|\Theta, \xi)$
  - $\log P(H, D \mid \Theta, \xi) = \log P(H \mid D, \Theta, \xi) + \log P(D \mid \Theta, \xi)$
  - $\log P(D \mid \Theta, \xi) = \log P(H, D \mid \Theta, \xi) - \log P(H \mid D, \Theta, \xi)$
- Average both sides with $P(H\ |D, \Theta', \xi)$ for Θ'
  - $E_{H|D,\Theta'} logP(D|\Theta, \xi) = E_{H|D,\Theta'} logP(H, D|\Theta, \xi) - E_{H|D,\Theta'} logP(H|D, \Theta, \xi)$
  - $\log P(D \mid \Theta, \xi) = F(\Theta \mid \Theta') = E(\Theta \mid \Theta') + H(\Theta \mid \Theta')$

Log-likelihood of data

- EM uses the true posterior. $P(H|D, \Theta', \xi)$

9

Expectation Maximization (EM)

- General EM Algorithm:
  - Initialize parameters Θ
  - Set Θ'=Θ
- Expectation step
  - $E(\Theta|\Theta') = \langle logP(H, D|\Theta, \xi)\rangle_{P(H|D,\Theta')}$
- Maximization step
  - $\Theta = \text{argmax}\ E(\Theta|\Theta')$
  - Repeat until no or small improvement in Θ ((Θ = Θ')
- Problem
  - $P(H|D, \Theta') = \prod_{n=1}^{N} P(x^{(n)},\ s^{(n)}|\Theta')$
  - Each data point requires us to calculate 2^k probabilities
  - If k is large, then this is a bottleneck

10

Variational Approximation

- An alternative method to approximate inference based on stochastic sampling.
- Let *H* be a set of all variables with hidden or missing values
  - $\log P(D \mid \Theta, \xi) = \log P(H, D \mid \Theta, \xi) - \log P(H \mid D, \Theta, \xi)$
- Average both sides using a distribution $Q(H \mid \lambda)$ [*surrogate posterior*]

$$E_{H|\lambda} logP(D|\Theta,\xi) = E_{H|\lambda} logP(H,D|\Theta,\xi) - E_{H|\lambda} logQ(H\,|\lambda) + E_{H|\lambda} logQ(H\,|\lambda) - E_{H|\lambda} logP(H|\Theta,\xi)$$

$$\log P(D|\Theta,\xi) = F(Q,\Theta) + KL(Q,P)$$

$$F(Q,\Theta) = \Sigma_{\{H\}} Q(H\,|\lambda) logP(H,D|\Theta,\xi) - \Sigma_{\{H\}} Q(H\,|\lambda) logQ(H\,|\lambda)$$

$$KL(Q,P) = \Sigma_{\{H\}} Q(H\,|\lambda)[logQ(H\,|\lambda) - logP(H\,|D,\Theta)]$$

11

Variational Approximation

$$\log P(D|\Theta,\xi) = F(Q,\Theta) + KL(Q,P)$$

$$F(Q,\Theta) = \Sigma_{\{H\}} Q(H\,|\lambda) logP(H,D|\Theta,\xi) - \Sigma_{\{H\}} Q(H\,|\lambda) logQ(H\,|\lambda)$$

$$KL(Q,P) = \Sigma_{\{H\}} Q(H\,|\lambda)[logQ(H\,|\lambda) - logP(H\,|D,\Theta)]$$

- Approximation: maximize $F(Q,\Theta)$
- Parameters: Θ, λ
- Maximization of F pushes up the lower bound on the log-likelihood

  $\log P(D|\Theta,\xi) \geq F(Q,\Theta)$.

Kullback-Leibler (KL) divergence

- A method to measure the difference between two probability distributions over the same variable x
  - $KL(P \,||\, Q)$
  - Where the "||" operator indicates "*divergence*" or P's divergence from Q
- Entropy: the average amount of information for a probability distribution
  - $H(P) = E_P[I_P(X)] = -\sum_{i=1}^{n} P(i)\log(P(i))$
  - $KL(P\,||Q) = H(P,Q) - H(P) = -\sum_{i=1}^{n} P(i)\log\big(Q(i)\big) + \sum_{i=1}^{n} P(i)\log(P(i)) = \sum_{i=1}^{n} P(i)\log(\frac{P(i)}{Q(i)})$
- If we have some theoretic minimal distribution P, we want to try to find an approximation Q that tries to get as close as possible by minimizing the KL divergence

13

Variational EM

- To use Variational EM, we hope if we choose $Q(H \mid \lambda)$ well, the optimization of both λ and Θ will become easy.
- A well-behaved choice for $Q(H \mid \lambda)$ is the *mean field approximation*.
- Let H – be a set of all variables with hidden or missing values:
  - E-step: *Compute expectation over hidden variables*
    - Optimize: $F(Q, \Theta)$ with respect to λ while keeping Θ fixed.
  - M-step: *Maximize expected loglikelihood*
    - Optimize: $F\ (Q\ , \Theta)$ with respect to Θ while keeping λs fixed.

14

Mean Field Approximation

- To find the distribution Q, we use Mean Field Approximation
- Assumption:
  - $Q(H|\lambda)$ is the *mean field approximation*
  - Variables in the $Q(H)$ distribution are *independent* variables H_i
  - Q is completely *factorized*

$$Q(H|\lambda) = \prod Q_i(H_i|\lambda_i)$$

  - For our CVQ model
    - Hidden variables are binary sources

$$Q(H|\lambda) = \prod_{n=1...N} Q(s^{(n)}|\lambda^n)$$

$$Q(s^{(n)}|\lambda^n) = \prod_{i=1...k} Q(s_i^{(n)}|\lambda_i^{(n)})$$

$$Q\left(s_i^{(n)}\middle|\lambda_i^{(n)}\right) = \lambda_i^{(n)\, s_i{}^{(n)}} (1 - \lambda_i^{(n)})^{1 - s_i{}^{(n)}}$$

15

Mean Field Approximation

- Functional F for the mean field:

$$F(Q,\Theta) = \sum_{\{H\}} Q(H|\lambda) \log P(H, D|\Theta, \xi) - \sum_{\{H\}} Q(H|\lambda) \log Q(H|\lambda)$$

- Assume just one data point **x** and corresponding **s** :

$$F(Q,\Theta) = \sum_{n=1}^{N} \langle log P((x^{(n)}, s^{(n)}|\Theta) \rangle_{Q(s^{(n)}|\lambda^{(n)})} - \langle log Q(\, s^{(n)}|\lambda^{(n)}) \rangle_{Q\left(s^{(n)}|\lambda^{(n)}\right)}$$

$$= \langle -d\log\sigma - \frac{1}{2\sigma^2}(\mathbf{x} - \mathbf{Ws})^T(\mathbf{x} - \mathbf{Ws}) \rangle_{Q(s|\lambda)} \quad (1)$$

$$+ \langle \sum_{i=1}^{k} s_i \log\pi_i + (1 - s_i)\log(1 - \pi_i) \rangle_{Q(s|\lambda)} \quad (2)$$

$$- \langle \sum_{i=1}^{k} s_i \log\lambda_i + (1 - s_i)\log(1 - \lambda_i) \rangle_{Q(s|\lambda)} \quad (3)$$

16

Mean Field Approximation

- Functional F. Part (1)

$$\langle -d\log\sigma - \frac{1}{2\sigma^2}(\mathbf{x}-\mathbf{W}\mathbf{s})^T(\mathbf{x}-\mathbf{W}\mathbf{s})\rangle_{Q(s|\lambda)}$$

$$= \langle -d\log\sigma - \frac{1}{2\sigma^2}\left(\mathbf{x}-\sum_{i=1}^{k} s_i\,\mathbf{w}_i\right)^T\left(\mathbf{x}-\sum_{i=1}^{k} s_i\,\mathbf{w}_i\right)\rangle_{Q(s|\lambda)}$$

$$= \langle -d\log\sigma - \frac{1}{2\sigma^2}\left[\mathbf{x}^T\mathbf{x} - 2\sum_{i=1}^{k}(s_i\,\mathbf{w}_i)\mathbf{x} + \sum_{i=1}^{k}\sum_{j=1}^{k} s_i\,s_j\mathbf{w}_i{}^T\mathbf{w}_j\right]\rangle_{Q(s|\lambda)}$$

$$= -d\log\sigma - \frac{1}{2\sigma^2}\left[\mathbf{x}^T\mathbf{x} - 2\sum_{i=1}^{k}\langle s_i\rangle_{Q(s_i|\lambda_i)}\,\mathbf{w}_i)\mathbf{x} + \sum_{i=1}^{k}\sum_{j=1}^{k}\langle s_i s_j\rangle_{Q(s|\lambda)}\mathbf{w}_i{}^T\mathbf{w}_j\right]$$

$$\langle s_i\rangle_{Q(s_i|\lambda_i)} = \lambda_i \qquad \langle s_i s_j\rangle_{Q(s|\lambda)} = \lambda_i\lambda_j + \delta_{ij}(\lambda_i - \lambda_i{}^2)$$

Mean Field Approximation

- Functional F. Part (2)

$$\langle\sum_{i=1}^{k} s_i\log\pi_i + (1-s_i)\log(1-\pi_i)\rangle_{Q(s|\lambda)} = \sum_{i=1}^{k}\langle s_i\rangle_{Q(s_i|\lambda_i)}\log\pi_i + (1-\langle s_i\rangle_{Q(s_i|\lambda_i)})\log(1-\pi_i)$$

$$= \sum_{i=1}^{k}\lambda_i\log\pi_i + (1-\lambda_i)\log(1-\pi_i)$$

- Functional F. part (3)

$$\langle\sum_{i=1}^{k} s_i\log\lambda_i + (1-s_i)\log(1-\lambda_i)\rangle_{Q(s|\lambda)} = \sum_{i=1}^{k}\lambda_i\log\lambda_i + (1-\lambda_i)\log(1-\lambda_i)$$

Mean Field Approximation

Functional F:

$$= -d\log\sigma - \frac{1}{2\sigma^2}\left[\mathbf{x}^T\mathbf{x} - 2\sum_{i=1}^{k}\langle s_i\rangle_{Q(s_i|\lambda_i)}\,\mathbf{w}_i)\mathbf{x} + \sum_{i=1}^{k}\sum_{j=1}^{k}\langle s_i s_j\rangle_{Q(s|\lambda)}\mathbf{w}_i{}^T\mathbf{w}_j\right]$$

$$+\sum_{i=1}^{k}\lambda_i\log\pi_i + (1-\lambda_i)\log(1-\pi_i)$$

$$+\sum_{i=1}^{k}\lambda_i\log\lambda_i + (1-\lambda_i)\log(1-\lambda_i)$$

Parameters: W, π, σ

Mean field parameters: λ

Mean Field Approximation

Functional F (for all data points):

$$F(Q,\Theta) = \sum_{n=1}^{N} \langle\log P(\mathbf{x}^{(n)}, \mathbf{s}^{(n)}|\Theta)\rangle_{Q(s^{(n)}|\lambda^{(n)})} - \langle\log Q(\mathbf{s}^{(n)}|\lambda^{(n)})\rangle_{Q(s^{(n)}|\lambda^{(n)})}$$

$$= -d\log\sigma - \frac{1}{2\sigma^2}\left[\mathbf{x}^{(n)T}\mathbf{x}^{(n)} - 2\sum_{i=1}^{k}\lambda_i{}^{(n)}\,\mathbf{w}_i)\mathbf{x}^{(n)} + \sum_{i=1}^{k}\sum_{j=1}^{k}\left[\lambda_i{}^{(n)}\lambda_j{}^{(n)} + \delta_{ij}(\lambda_i{}^{(n)} - \lambda_i{}^{(n)2})\right]\mathbf{w}_i{}^T\mathbf{w}_j\right]$$

$$+\sum_{i=1}^{k}\lambda_i{}^{(n)}\log\pi_i + (1-\lambda_i{}^{(n)})\log(1-\pi_i)$$

$$+\sum_{i=1}^{k}\lambda_i{}^{(n)}\log\lambda_i{}^{(n)} + (1-\lambda_i{}^{(n)})\log(1-\lambda_i{}^{(n)})$$

Parameters: W, π, σ

Mean field parameters: $\lambda = \lambda^1, \lambda^2, \ldots, \lambda^n$

Variational EM

- E-step
  - Optimize $F(Q,\Theta)$ with respect to λ while keeping Θ fixed

$$\frac{\partial}{\partial\lambda_u}F = \frac{1}{\sigma^2}(\mathbf{x} - \sum_{j\neq u}\lambda_j \mathbf{w}_j)^T\,\mathbf{w}_u - \frac{1}{2\sigma^2}\mathbf{w}_u{}^T\mathbf{w}_u + \log\frac{\pi_u}{1-\pi_u} - \log\frac{\lambda_u}{1-\lambda_u}$$

Set $\frac{\partial}{\partial\lambda_u}F = 0$

$$\lambda_u = g(\frac{1}{\sigma^2}(x - \sum_{j\neq u}\lambda_j w_j)^T w_u - \frac{1}{2\sigma^2}w_u{}^T w_u + \log\frac{\pi_u}{1-\pi_u}),\quad g(x) = \frac{1}{1+e^{-x}}$$

Variational EM

- M-step
  - Optimize $F\ (Q\ ,\Theta\ )$ with respect to Θ while keeping λs

Start with $\boldsymbol{\pi}$:

For N data points

$$\frac{\partial}{\partial\pi_u}F = \sum_{n=1}^{N}\lambda_u{}^n\log\frac{1}{\pi_u} - (1-\lambda_u{}^n)\log\frac{1}{(1-\pi_u)}$$

Set $\frac{\partial}{\partial\pi_u}F = 0,$

$$\pi_u = \frac{\sum_{n=1}^{N}\lambda_u{}^{(n)}}{N}\qquad \text{(closed form solution)}$$

Variational EM

And for parameter w:

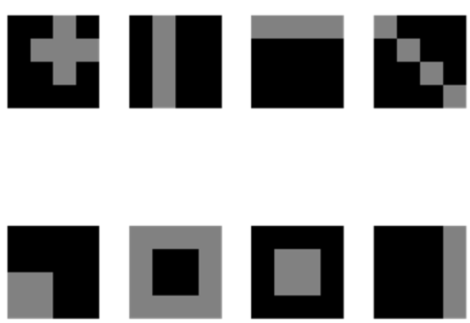
$$\frac{\partial}{\partial w_{uv}} F = \sum_{n=1}^{N} -\frac{1}{2\sigma^2}\left[\lambda_v{}^{(n)} x_u{}^{(n)} + 2\sum_{j\neq v} \lambda_v{}^{(n)}\, \lambda_j{}^{(n)} w_{uj} + 2\lambda_v{}^{(n)} w_{uv}\right] = 0$$

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & .. & w_{1k} \\ w_{21} & & & \\ & .. & & \\ w_{d1} & .. & .. & w_{dk} \end{pmatrix} \qquad \mathbf{W} = (\mathbf{w_1}\ \mathbf{w_2}\ \ldots\ \mathbf{w_k})$$

- For each variable v:

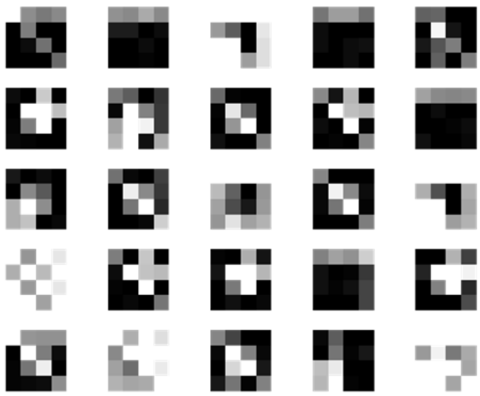
The equations define a set of k linear equations that can be solved

Image Separation Experiment



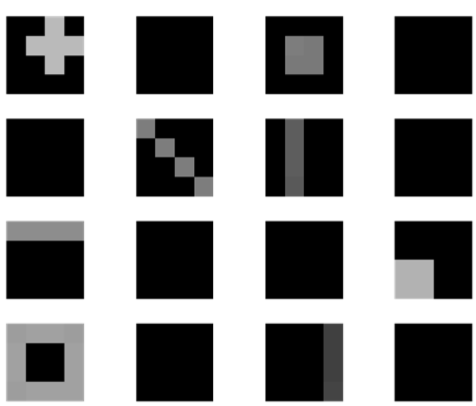
Source images associated with latent variables

Mixed images



- Images generated by the model.
- Some of the images are noise.
- Generating enough samples; the model can retrieve the original source.

Recovered sources



Modeling High-Dimensional Data

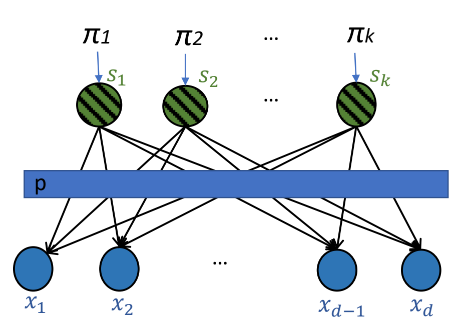
- Definition: Number of dimensions are high that makes calculations extremely difficult (number of features exceed number of observations).
- Examples of domains with High-Dimensional Data:
  - Sensor Networks
  - Document Repositories.
- Typically, variables are dependent.
- How to model dependencies?
  - Full model (intractable, overfitting)
  - All-independent (unrealistic)
  - Middle-of-the-road approaches
    - Captures dependencies in an efficient way (representation, reasoning, learning).

Noisy-OR Component Analyzer

- Objective: Capture dependencies via latent factors and combinations.
- The dependencies between observables are represented using a smaller number of *hidden binary* factors.
- NOCA model has binary nodes:
  - k parameters for each observed node, p_{1j}, …., p_{kj}
  - p_{ij} is interpreted as "strength of influence" of S_i on observable variable x_j

$$P(X_j = 0|s) = \prod_{i=1}^{K} (1 - p_{ij})^{s_i}$$

$$P(X_j = 1|\mathbf{s}) = 1 - P(X_j = 0|\mathbf{s}) = 1 - \prod_{i=1}^{K} (1 - p_{ij})^{s_i}$$



Noisy-OR Component Analyzer

- **A generalization of the logical OR**

Assumptions:

- All possible causes U_i for an event X are modeled using nodes (random variables) and their values, with T (or 1) reflecting the presence of the cause , and F (or 0) its absence
- If one needs to represent unknown causes one can add a leak node
- **Parameters:** For each cause U_i define an (independent) probability qi that represents the probability with which the cause does not lead to X = T (or 1), or in other words, it represents the probability that the positive value of variable X is inhibited when U_i is present

$$p(x = 1|U_1, \ldots, U_j, \neg U_{j+1}, \ldots, \neg U_k) = 1 - \prod_{i=1}^{j} q_i$$

$$p(x = 0|U_1, \ldots, U_j, \neg U_{j+1}, \ldots, \neg U_k) = \prod_{i=1}^{j} q_i$$

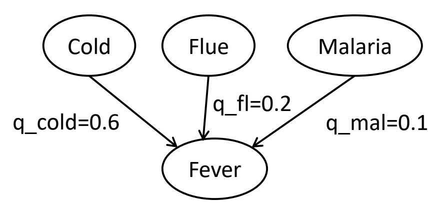
Note: The negated causes $\neg U_i$ (reflecting the absence of the cause) do not have any influence on X.Why?

29

Noisy-OR Example

$$\mu(x = 1|U_1, \ldots, U_j, \neg U_{j+1}, \ldots, \neg U_k) = 1 - \prod_{i=1}^{j} q_i$$

$$\mu(x = 0|U_1, \ldots, U_j, \neg U_{j+1}, \ldots, \neg U_k) = \prod_{i=1}^{j} q_i$$



Cold	Flu	Malaria	μ(Fever)	$\mu(\neg Fever)$
F	F	F	0	1
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	0.6
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

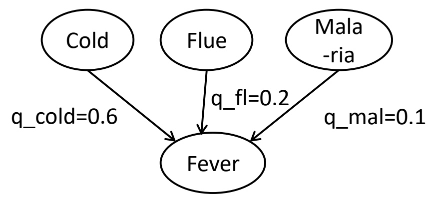
30

Noisy-OR parameter reduction

- Please note that in general the number of entries defining the CPT (conditional probability table) grows exponentially with the number of parents;
  - for q binary parents the number is : 2^q
- For the noisy-or CPT the number of parameters is q + 1

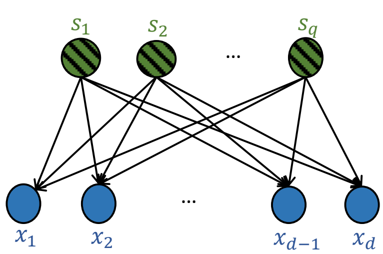
Example:
CPT: 8 different combination of values for 3 binary parents

Noisy-or: 4 parameters



31

Noisy-OR Component Analyzer (NOCA)



Latent variables s: $(q+1)$-dimensions

$s \in \{0,1\}^q, P(s_i|\pi_i) = \pi_i^{s_i}(1-\pi_i)^{1-s_i}$

Loading Matrix: $\boldsymbol{p} = \{p_{ij}\}_{j=1,\ldots,d}^{i=1,\ldots,q}$

$q < d$

Observed variables x : d-dimensions

$x \in \{0,1\}^d$

$$P(x) = \sum_{\{s\}} \left(\prod_{j=1}^{d} P(x_j|s) \right) \left(\prod_{i=1}^{q} P(s_i) \right)$$

32

Why EM won't work?

- Take N iid samples (D-dimensional binary vectors)
- We will need:
  - The joint distribution

Problem1: not a product

$$P(\mathbf{x},\mathbf{s}) = P(\mathbf{x}|\mathbf{s})P(\mathbf{s}) = P(\mathbf{s})\prod_j \left(1-\prod_{i=1}^{K}(1-p_{ij})^{s_i}\right)^{x_j}\left(\prod_{i=1}^{K}(1-p_{ij})^{s_i}\right)^{1-x_j}$$

  - Joint over observables

$$P(\mathbf{x}) = \sum_{\mathbf{s}} P(\mathbf{x},\mathbf{s}) = \sum_{\mathbf{s}}\left(\prod_j P(x_j|\mathbf{s})\right)P(\mathbf{s})$$

Problem 2: summation over 2^K terms

Variational EM for NOCA

- Similar to what we did for CVQ, we simplify the distribution with a decomposable Q(s)

$$\log(P(x|\theta)) = \log\left(\prod_{n=1}^{N} P(x_n|\theta)\right) = \sum_{n=1}^{N}\log\left[\sum_{\{s\}} P(x_n,s_n|\theta)\right]$$

$$= \sum_{n=1}^{N}\log\left[\sum_{\{s\}} P(x_n,s_n|\theta,q_n)\frac{Q(s_n)}{Q(s_n)}\right] \geq \sum_{n=1}^{N}\left[\sum_{\{s_n\}} E_{s_n}\log(P(x_n,S_n|\theta)) - E_{s_n}\log(Q(S_n))\right]$$

- $\log(P(x_n,s_n|\theta,q_n)$ still can not be solved easily
- Noisy-Or is not in exponential family

Variational EM for NOCA

A further lower bound is required

- **Jensen's inequality:** $\boldsymbol{f(a+\sum_j q_j x_j) \geq \sum_j q_j f(a+x_j)}$

$$P(x_j|s) = \left[1-(1-p_{0j})\prod_{i=1}^{q}(1-p_{ij})^{s_i}\right]^{x_j}\left[(1-p_{0j})\prod_{i=1}^{q}(1-p_{ij})^{s_i}\right]^{(1-x_j)}$$

Set $\boldsymbol{\theta_{ij} = -\log(1-p_{ij})}$

$$P(x_j|s) = \exp\left[x_j \log\left(1-\exp\left\{-\theta_{0j}-\sum_{i=1}^{q}\theta_{ij}s_i\right\}\right)+(1-x_j)\left(-\theta_{0j}-\sum_{i=1}^{q}\theta_{ij}s_i\right)\right]$$

$\boldsymbol{P(x_j|s)}$ **does not factorize for** $\boldsymbol{x_j = 1}$

$P(x_j = 1|s) = \exp[\log\left(1-\exp\left\{-\theta_{0j}-\sum_{i=1}^{q}\theta_{ij}s_i\right\}\right)]$

$$= \exp\left[\log\left(1-\exp\left\{-\theta_{0j}-\sum_{i=1}^{q}\theta_{ij}s_i\frac{q_j(i)}{q_j(i)}\right\}\right)\right] \geq \exp[\sum_{i=1}^{q} q_j(i)\log\left(1-\exp\left\{-\theta_{0j}-\frac{\theta_{ij}s_i}{q_j(i)}\right\}\right)]$$

$$= \exp[\sum_{i=1}^{q} q_j(i)\,[s_i\log\left(1-\exp\left\{-\theta_{0j}-\frac{\theta_{ij}}{q_j(i)}\right\}\right)+(1-s_i)\log(1-\exp\{-\theta_{0j}\})]]$$

$$= \prod_{i=1}^{q}\exp[q_j(i)s_i[\log\left(1-\exp\left\{-\theta_{0j}-\frac{\theta_{ij}}{q_j(i)}\right\}\right)-\log((1-\exp\{-\theta_{0j}\}))]+q_j(i)\log((1-\exp\{-\theta_{0j}\}))]]$$

Variational EM for NOCA

A further lower bound is required

$\log(P(x|\theta))$

$$\geq \sum_{n=1}^{N}\left[\sum_{\{s_n\}} E_{s_n}\log P(x_n,s_n|\theta) - E_{s_n}\log Q(s_n)\right]$$

$$\geq \sum_{n=1}^{N}\left[\sum_{\{s_n\}} E_{s_n}\log\left(\tilde{P}(x_n,s_n|\theta,q_n)\right) - E_{s_n}\log(Q(s_n))\right]$$

$$= \sum_{n=1}^{N}\left[\sum_{\{s_n\}} E_{s_n}\log\left(\tilde{P}(x_n|s_n,\theta,q_n)P(s_n|\theta)\right) - E_{s_n}\log(Q(s_n))\right]$$

$$= \sum_{n=1}^{N}\mathcal{F}_n(x_n,Q(s_n))$$

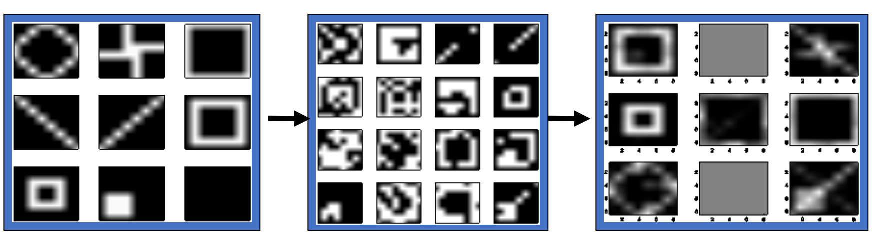
$$= \mathcal{F}(x,Q(s))$$

Variational EM for NOCA

Parameters: $q_n, \theta_{ij}, \theta_{0j}$

- **E-step: update q_n to optimize F_n**
- $q_{nj}(\mathrm{i}) \Leftarrow \langle s_{ni} \rangle_{Q(S_n)} \frac{q_{nj}(\mathrm{i})}{\log\left(1-\mathrm{e}^{-\theta_{0j}}\right)} \Big[\log\left(1 - A^n(\mathrm{i}, j)\right) - \frac{\theta_{ij}}{q_{nj}(\mathrm{i})} \frac{A^n(\mathrm{i},j)}{1-A^n(\mathrm{i},j)} -$

Structure Recovery Experiment



(a) (b) (c)

a) Image patterns associated with hidden sources.
b) Example images generated by the NOCA model
c) Images recovered from source input.

References

- X. Lu, M. Hauskrecht, R.S. Day. Variational Bayesian learning of the cooperative vector quantizer (CVQ) model. Part I: The Theory, *Technical Report*, *Computer Science Department*, *University of Pittsburgh*, 2002
- Singliar, Hauskrecht. Noisy-or Component Analysis and its Application to Link Analysis. *Journal of Machine Learning Research* 2006
- https://people.cs.pitt.edu/~milos/courses/cs3750-Fall2007/lectures/class20.pdf
- http://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class8.pdf
- http://www.blutner.de/Intension/Noisy%20OR.pdf
- Jaakkola, Tommi S., and Michael I. Jordan. "Variational probabilistic inference and the QMR-DT network." Journal of artificial intelligence research 10 (1999): 291-322.
- http://bjlkeng.github.io/posts/variational-bayes-and-the-mean-field-approximation/
- https://machinelearningmastery.com/divergence-between-probability-distributions/
- https://towardsdatascience.com/deep-latent-variable-models-unravel-hidden-structures-a5df0fd32ae2