Example: Hidden CRF for Gesture Recognition

- Sy Bor Wang, A. Quattoni, L. . -P. Morency, D. Demirdjian and T. Darrell, "Hidden Conditional Random Fields for Gesture Recognition," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006, pp. 1521-1527, doi: 10.1109/CVPR.2006.132.

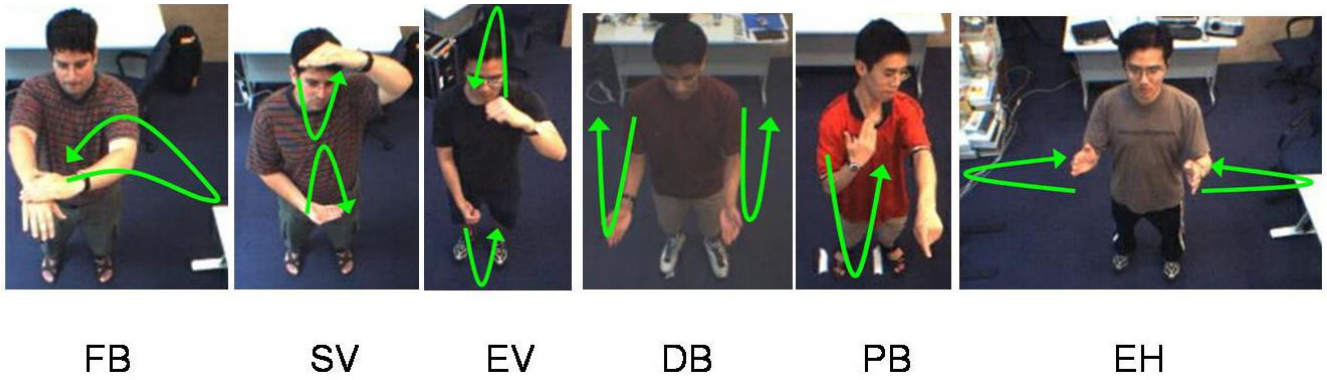


Figure 1. Illustrations of the six gesture classes for the experiments. Below each image is the abbreviation for the gesture class. These gesture classes are: FB - Flip Back, SV - Shrink Vertically, EV - Expand Vertically, DB - Double Back, PB - Point and Back, EH - Expand Horizontally. The green arrows are the motion trajectory of the fingertip and the numbers next to the arrows symbolize the order of these arrows.

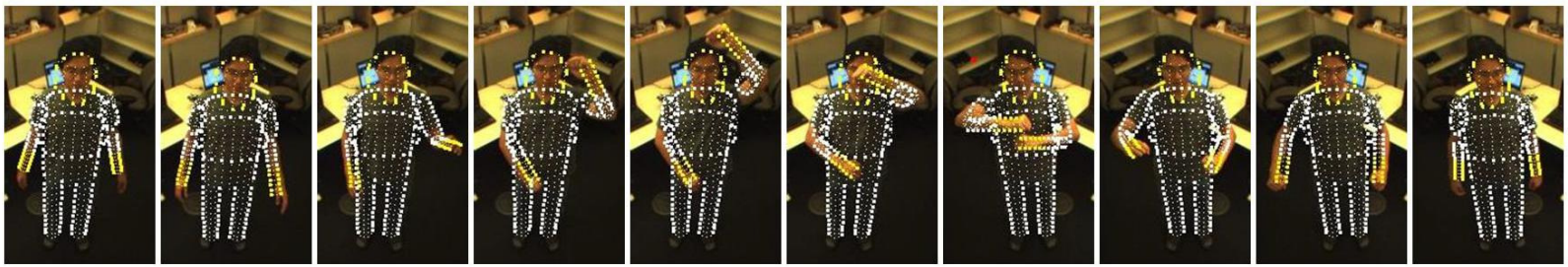


Figure 5. Sample image sequence with the estimated body pose superimposed on the user in each frame.

HCRF Example

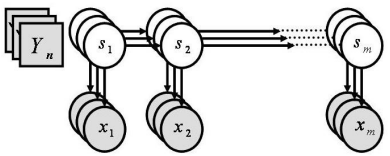


Figure 2. HMM model

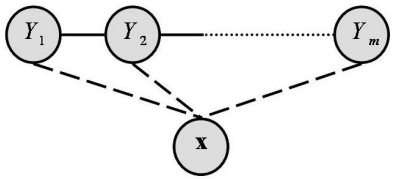


Figure 3. CRF Model

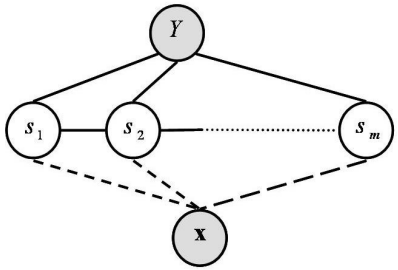


Figure 4. HCRF Model

Models	Accuracy (%)
HCRF $\omega = 0$	86.44
HCRF $\omega = 1$	96.81
HCRF $\omega = 2$	97.75

top Hidden Markov Model for each gesture.

middle Conditional Random Field: For each frame, the gesture type Y is estimated, the most frequent one is used for classification.

- Different window size ω is used.

bottom 12 hidden states for each frame are learned; the overall classification is base on these estimated states.

Potentials

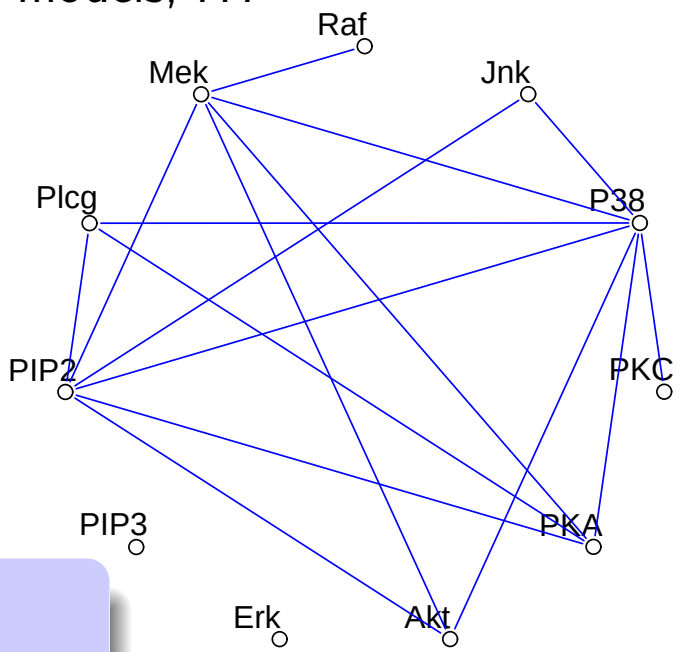
$$\Psi(y, \mathbf{s}, \mathbf{x}; \theta, \omega) = \sum_{j=1} \varphi(\mathbf{x}, j, \omega) \cdot \theta_s[s_j] + \sum_{j=1} \theta_y[y, s_j] + \sum_{(j,k)\in E} \theta_e[y, s_j, s_k] \quad (3)$$

Models	Accuracy (%)
HMM $\omega = 0$	84.22
CRF $\omega = 0$	86.03
CRF $\omega = 1$	81.75
HCRF (one-vs-all) $\omega = 0$	87.49
HCRF (multi-class) $\omega = 0$	91.64
HCRF (multi-class) $\omega = 1$	93.81

Undirected (Pairwise, Continuous) Graphical Models

- The **generative model** represents the full probability distribution $P(X)$.
- Missing edges represent conditional independence of the variables.
- Chapter 17 Elements of Statistical Learning
  - Gaussian graphical models
  - Markov random fields
  - Ising model, (restricted) Boltzmann machine
    other not mentioned
  - Bayesian networks, Mixed interaction models, ...

- Cytometry dataset (ESLII)
- $N = 7466$ cells
- $p = 11$ proteins
- We ame to model protein co-occurence probability.



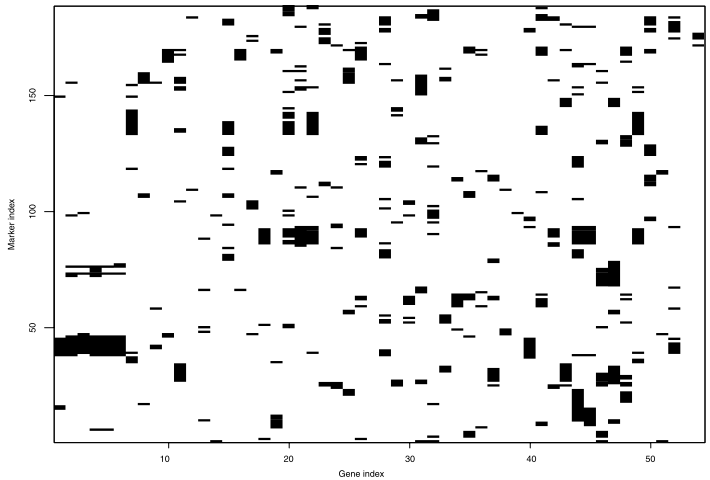
sklearn.covariance.GraphicalLasso # basics

gRbase # the recommended R package

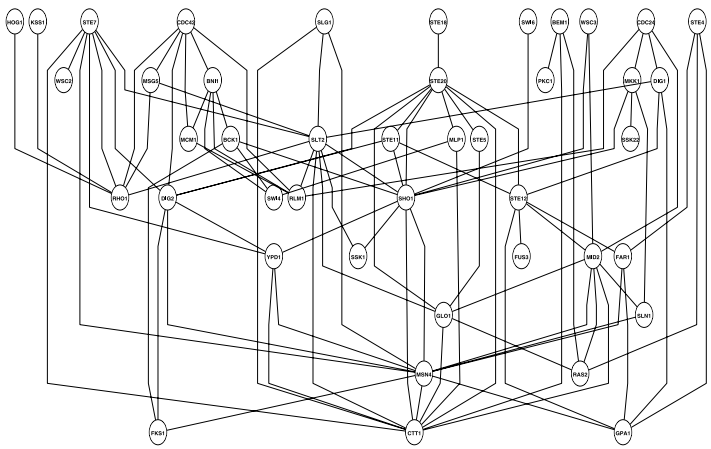
Sparse Conditional Gaussian Graphical Model Application

Yin, Jianxin & Li, Hongzhe. (2011). *A sparse conditional Gaussian graphical model for analysis of genetical genomics data.* The annals of applied statistics. 5. 2630-2650.

- Cytometry dataset (ESLII)
- $p_Y = 54$ gene level expressions
- $p_X = 188$ markers (discrete)
- $Y^{p_Y} | X^{p_X} \sim \mathcal{N}(M^{p_Y \times p_X} X^{p_X}, \Sigma^{p_Y \times p_Y})$ conditional Gaussian distribution
- Top: Black color indicates significant association $p - value < 0.01$ in the linear regression.
- Bottom: The undirected graph of 43 genes constructed on the cGGM.



(a)



Data: carcass

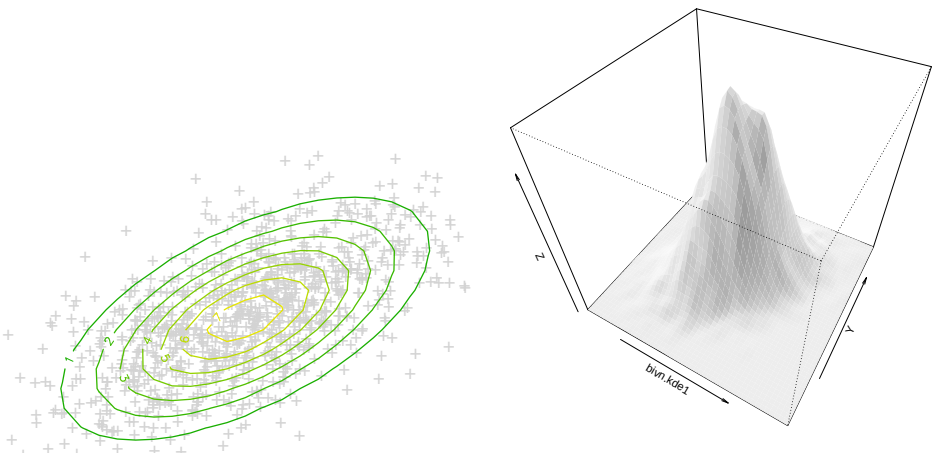
Data: carcass #Source: Soren Hojsgaard, David Edwards, Steffen Lauritzen: *Graphical Models with R*, Springer.

	mean.
Fat11	16.00
Meat11	52.00
Fat12	14.00
Meat12	52.00
Fat13	13.00
Meat13	56.00
LeanMeat	59.00

Σ	Fat11	Meat11	Fat12	Meat12	Fat13	Meat13	LeanMeat
Fat11	11.34	0.74	8.42	2.06	7.66	-0.76	-9.08
Meat11	0.74	32.97	0.67	35.94	2.01	31.97	5.33
Fat12	8.42	0.67	8.91	0.31	6.84	-0.60	-7.95
Meat12	2.06	35.94	0.31	51.79	2.18	41.47	6.03
Fat13	7.66	2.01	6.84	2.18	7.62	0.38	-6.93
Meat13	-0.76	31.97	-0.60	41.47	0.38	41.44	7.23
LeanMeat	-9.08	5.33	-7.95	6.03	-6.93	7.23	12.90

Gaussian Graphical Models (Undirected Graphs)

- **Multivariate Gaussian Distribution** on variables $X = (X_1, \ldots, X_p)$
- $\phi(\mathbf{x}) = \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)\Sigma^{-1}(\mathbf{x}-\mu)}$
  - $|.|$ is the determinant. we denote p the number of components in $\mathbf{x}$. Then $|2\pi\Sigma| = (2\pi)^p|\Sigma|$.
- If Σ is not invertible it has dependent columns. It means that the variables $\mathbf{x}_j$ are lineary dependent.
  - If the **rank** of Σ is ℓ then there exists a matrix A and a vector ν so:
  - $x = Az + \nu$ for new coordinates z with ℓ dimensions
  - We just consider the new coordinates and assume Σ has a full rank.



Concentration matrix

- **Concentration (Precision, koncentrační) matrix**

$$K = \Sigma^{-1}$$

Lemma

For $u \neq v$, $k_{uv} = 0$ if and only if y_u and y_v are conditionally independent given all other variables.

k*100	Fat11	Meat11	Fat12	Meat12	Fat13	Meat13	LeanMeat
Fat11	44	3	-20	-7	-16	4	10
Meat11	3	16	-3	-6	-6	-6	-3
Fat12	-20	-3	54	6	-21	-5	9
Meat12	-7	-6	6	14	-1	-9	-0
Fat13	-16	-6	-21	-1	56	3	7
Meat13	4	-6	-5	-9	3	16	-1
LeanMeat	10	-3	9	-0	7	-1	26

- If looking for small values better to 'scale' the entries into Partial Correlation matrix.

Definition (Partial correlation matrix)

Partial correlation matrix is defined from K by

$$\rho_{uv|V\setminus\{uv\}} = \frac{-k_{uv}}{\sqrt{k_{uu}k_{vv}}}.$$

Compare to Pearson correleation $\frac{cov(u,v)}{\sigma_u\sigma_v}$.

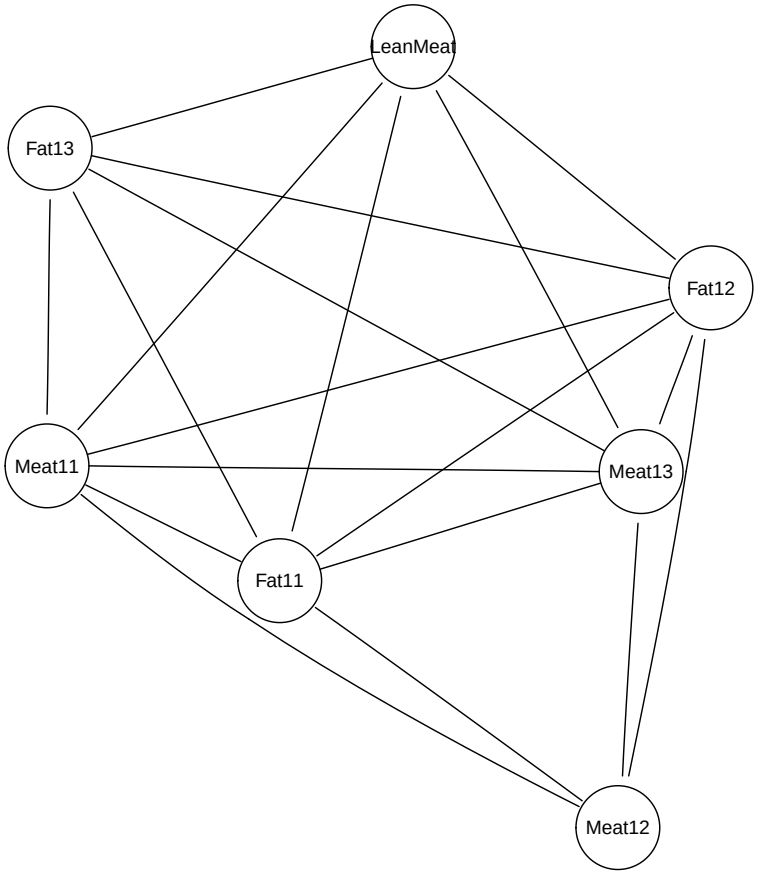
Lemma

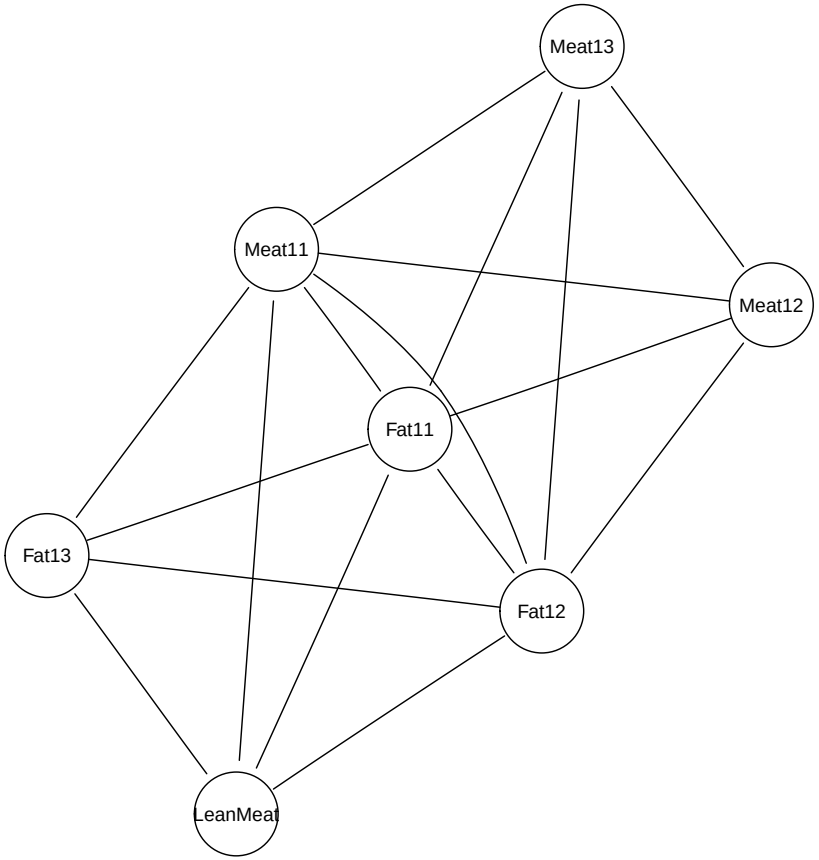
In contrast to concentrations, the partial correlations are invariant under a change of scale and origin in the sense that if $X_j^* = a_jX_j + b_j$, $j = 1,\ldots,p$ *then* $a_va_uk_{uv}^* = k_{uv}$ *and* $\rho_{uv|V\setminus\{uv\}}^* = \rho_{uv|V\setminus\{uv\}}$.

$\rho * 100$	Fat11	Meat11	Fat12	Meat12	Fat13	Meat13	LeanMeat
Fat11	-	-11	41	30	32	-16	-29
Meat11	-11	-	9	41	19	35	16
Fat12	41	9	-	-24	38	18	-24
Meat12	30	41	-24	-	2	61	2
Fat13	32	19	38	2	-	-9	-18
Meat13	-16	35	18	61	-9	-	7
LeanMeat	-29	16	-24	2	-18	7	-

Models

- The simplest model just removes edges with small $|\rho_{uv|V\setminus\{uv\}}|$. Penalized criteria will be introduced later.





Undirected Gaussian graphical model

Definition (Undirected Gaussian graphical model)

An **undirected Gaussian graphical model** is represented by an undirected graph $\mathcal{G} = (X, E)$, $X = \{X_1, \ldots, X_p\}$ represent the set of variables and E is a set of undirected edges.
When a random vector $\mathbf{x}$ follows a Gaussian distribution $N_p(\mu, \Sigma)$, the graph G represents the model where $K = \Sigma^{-1}$ is a positive definite matrix with $k_{u,v} = 0$ whenever there is no edge between vertices u, v in G.
This graph is called the **dependence graph** of the model.

Lemma

For any non adjacent vertices $u, v \in \mathcal{G}$ it holds: $u \perp\!\!\!\perp v | \mathbf{X} \setminus \{u, v\}$.

Definition (Generating class)

Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be the set of cliques of the dependence graph $\mathcal{G}$. A set of functions $g_1(), g_2(), \ldots, g_k()$ defined on $g_i(\mathbf{x}_{C_i})$ is called a **generating class** for the distribution

$$f(\mathbf{x}) = \prod_{i=1,\ldots,k} g_i(\mathbf{x}_{C_i}).$$

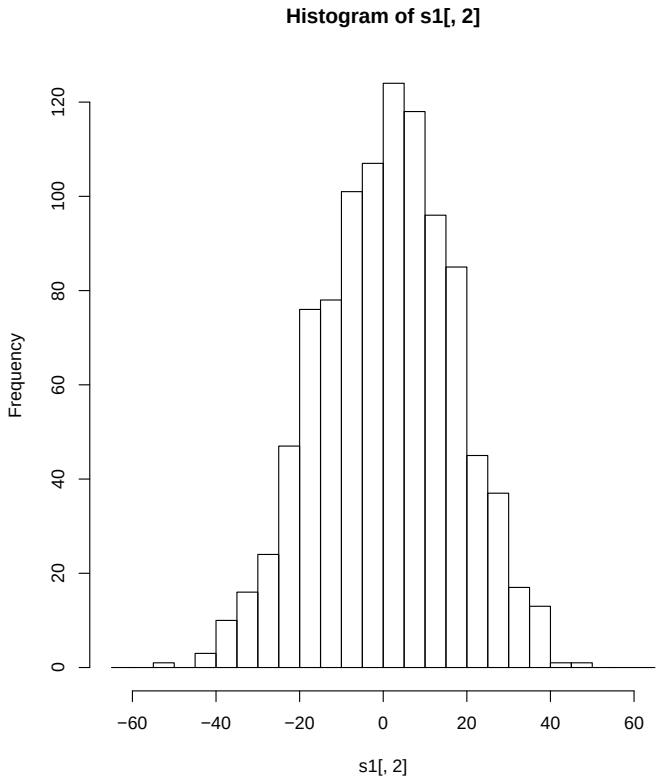
Marginalization

- We have $\frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)\Sigma^{-1}(\mathbf{x}-\mu)}$
- We want the distribution over variables $\{x_3, x_5, x_7\} \subset \{x_1, \ldots, x_p\}$

Marginal of a Gaussian Distribution

The marginal of a Gaussian distribution is calculated by removing appropriate dimensions from the mean and covariance matrix.

- $\mu_{3,5,7} = (\mu_3, \mu_5, \mu_7)$ and
$$\Sigma_{3,5,7} = \begin{bmatrix} \Sigma_{33} & \Sigma_{35} & \Sigma_{37} \\ \Sigma_{53} & \Sigma_{55} & \Sigma_{57} \\ \Sigma_{73} & \Sigma_{75} & \Sigma_{77} \end{bmatrix}$$
- $\phi_{x_3,x_5,x_7} = \frac{1}{\sqrt{|2\pi\Sigma_{3,5,7}|}} e^{-\frac{1}{2}(x_{3,5,7}-\mu_{3,5,7})\Sigma^{-1}_{3,5,7}(x_{3,5,7}-\mu_{3,5,7})}$



Conditioning

- We ame for $\phi(A|B)$ where
  - $A \subset \{x_1, \ldots, x_p\}$ having q elements,
  - the rest $B = \{x_1, \ldots, x_p\} \setminus A$ has $(p - q)$ elements.
- We rearrange the rows and columns to have A together. Then we get

$$x = \begin{bmatrix} x_A \\ x_B \end{bmatrix} \text{ (one column)}, \ \mu = \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \text{ (one column)},$$

$$\Sigma = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix} \text{ with dimensions } \begin{bmatrix} q \times q & q \times (p-q) \\ (p-q) \times q & (p-q) \times (p-q) \end{bmatrix}.$$

Conditional Gaussian

The parameters of the conditional Gaussian distribution $\phi(A|B = b) = N(\mu_{A|B=b}, \Sigma_{A|B=b})$ are:

$$\begin{aligned} \mu_{A|B=b} &= \mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(b - \mu_B) \\ \Sigma_{A|B=b} &= \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}. \end{aligned}$$

Covariance matrix differs but does not depend on the observation b. It depends on the fact B was observed.

Conditional Gaussian Example

- $\mu^T = (1, 2, 3, 4)$
- $\Sigma = \begin{bmatrix} 10 & 1 & 5 & 4 \\ 1 & 10 & 2 & 6 \\ 5 & 2 & 10 & 3 \\ 4 & 6 & 3 & 10 \end{bmatrix}$
- We observed (X_3, X_4) to be $(2.8, 4.1)$
- We ask for $\phi(A|B) = \phi(\{X_1, X_2\}|\{X_3, X_4\})$
- $\Sigma_{AB} = \begin{bmatrix} 5 & 4 \\ 2 & 6 \end{bmatrix}$
- $\Sigma_{BB} = \begin{bmatrix} 10 & 3 \\ 3 & 10 \end{bmatrix}$
- $\Sigma_{BB}^{-1} \doteq \begin{bmatrix} 0.11 & -0.033 \\ -0.033 & 0.11 \end{bmatrix}$
- $\Sigma_{AB}\Sigma_{BB}^{-1} \doteq \begin{bmatrix} 0.418 & 0.275 \\ 0.0220 & 0.593 \end{bmatrix}$
- $\mu_{A|B=b} = \mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(b - \mu_B)$
- $\mu_{A|B} \doteq \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 0.418 & 0.275 \\ 0.0220 & 0.593 \end{bmatrix} \begin{bmatrix} (2.8 - 3) \\ (4.1 - 4) \end{bmatrix}$
- $\mu_{A|B} \doteq \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} -0.056 \\ 0.055 \end{bmatrix} = \begin{bmatrix} 0.944 \\ 2.055 \end{bmatrix}$
- $\Sigma_{A|B=b} = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$
- $\Sigma_{A|B=b} \doteq \begin{bmatrix} 10 & 1 \\ 1 & 10 \end{bmatrix} - \begin{bmatrix} 2.53 & 2.26 \\ 2.26 & 4.13 \end{bmatrix}$
- $\Sigma_{A|B=b} \doteq \begin{bmatrix} 7.47 & -1.26 \\ -1.26 & 3.65 \end{bmatrix}$

Partition Matrix Inverse Properties

- The concentration matrix $K = \Sigma^{-1}$ is the inverse of the correlation matrix, therefore:

$$\begin{pmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{pmatrix} \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix} = \begin{pmatrix} I_{AA} & \mathbf{0} \\ \mathbf{0} & I_{BB} \end{pmatrix}$$

- From the top right part we get:

$$\begin{aligned} K_{AA}\Sigma_{AB} &+ K_{AB}\Sigma_{BB} = \mathbf{0} \\ -K_{AA}\Sigma_{AB}\Sigma_{BB}^{-1} &= K_{AB}(1) && (1) \\ \Sigma_{AB}\Sigma_{BB}^{-1} &= -K_{AA}^{-1}K_{AB}(2). && (2) \end{aligned}$$

- Take the top left part and substitute (1):

$$\begin{aligned} K_{AA}\Sigma_{AA} &+ K_{AB}\Sigma_{BA} = I_{AA} \\ K_{AA}\Sigma_{AA} &+ (-K_{AA}\Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}) = I_{AA} \\ K_{AA}^{-1} &= \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}. \end{aligned}$$

Regression Coefficients

$$\begin{aligned} \mu_{A|B=b} &= \mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(b - \mu_B) \\ \Sigma_{A|B=b} &= \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA} \end{aligned}$$

- Consider x_1 to be a linear function of others with the noise $\epsilon_1 \sim N(0, \sigma_1^2)$:

$$x_{1|2\ldots p} = \beta_1 + \beta_{12}x_2 + \beta_{13}x_3 + \ldots + \beta_{1p}x_p + \epsilon_1$$

- Set A the first dimension, B the remaining $(p-1) \times (p-1)$ matrix:

$$x_{1|B=(x_2,\ldots,x_p)^T} = \mu_{A|B} + \Sigma_{AB}\Sigma_{BB}^{-1}\left(\begin{bmatrix} x_2 \\ \ldots \\ x_p \end{bmatrix} - \mu_B\right) + \epsilon$$

- Recall (2): $\Sigma_{AB}\Sigma_{BB}^{-1} = -K_{AA}^{-1}K_{AB}$
- then $\sigma_1^2 = \frac{1}{k_{11}}$ with coefficients β

$$(\beta_{12}, \ldots, \beta_{1p}) = -\frac{(k_{12}, \ldots, k_{1p})}{k_{11}}. \tag{3}$$

Fit Linear Gaussian CPD

- To fit ML model of a linear gaussian CPD,
  - you fit the linear regression.

$$
\begin{aligned}
y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon_1 \\
\hat{\beta} &= (\mathbf{X^T X})^{-1} \mathbf{X^T y} \\
\hat{\sigma}^2_Y &= Cov(Y, Y) - \sum_i \sum_j \beta_i \beta_j Cov[X_i; X_j] \\
Cov(X_i; X_j) &= \mathbb{E}\left[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])\right] \\
\mathbb{E}[X_j] &= \frac{1}{N_{rows}} \sum_{i \in rows} x_{ij}
\end{aligned}
$$

```
from pgmpy.factors.continuous import LinearGaussianCPD
ml=maximum_likelihood_estimator(data, states)
cpdY.fit(data, states, estimator=ml, complete_samples_only=True)
```

https://cedar.buffalo.edu/~srihari/CSE674/Chap7/7.2-GaussBNs.pdf

Parameter Learning for a Gaussian Graphical Model

- Let us have the data $\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T$ over variables $\mathbf{x} \sim N_p(\mu, \Sigma)$.
- $S = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{x})(\mathbf{x}_i - \bar{x})^T$ is the empirical covariance matrix.
- Our model is represented by the concentration matrix $\Theta = \Sigma^{-1}$ and mean μ.
- Log-likelihood of the data is

$$loglik(\Theta, \mu) = \frac{N}{2} \log |\Theta| - \frac{N}{2} tr(\Theta S) - \frac{N}{2} (\bar{x} - \mu)^T \Theta (\bar{x} - \mu).$$

- for a fixed Θ is the maximum for μ: $\mu = \bar{x}$ and the last term is 0. We get
- $loglik(\Theta, \mu) \propto log|\Theta| - tr(\Theta S)$
- where $tr(\Theta S) = \sum_u \sum_v \theta_{uv} s_{uv}$, therefore only s_{uv} corresponding to non-zero θ_{uv} are considered by the sum.
- We replace the equality conditions by Lagrange multiplyers: $\ell_C(\Theta) = log|\Theta| - tr(\Theta S) - \sum_{(j,k) \notin E} \gamma_{jk} \theta_{jk}$
- We maximize. The derivative Θ should be zero (Γ is a matrix with non-zero for missing edges):

$$\Theta^{-1} - S - \Gamma = 0$$

Towards the Algorithm

- We iterate one row/column after another.
- We start with the sample covariance matrix

$$W_0 \leftarrow S$$

- We derive the formula for the last row/column: the derivative

$$\begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix} - \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix} - \begin{pmatrix} \Gamma_{11} & \gamma_{12} \\ \gamma_{12}^T & \gamma_{22} \end{pmatrix} = 0$$

- The upper right block can be written as $w_{12} - s_{12} - \gamma_{12} = 0$.
- W is inverse of Θ

$$\begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix} \begin{pmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^T & \theta_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^T & 1 \end{pmatrix}$$

- therefore the last column without last row is:

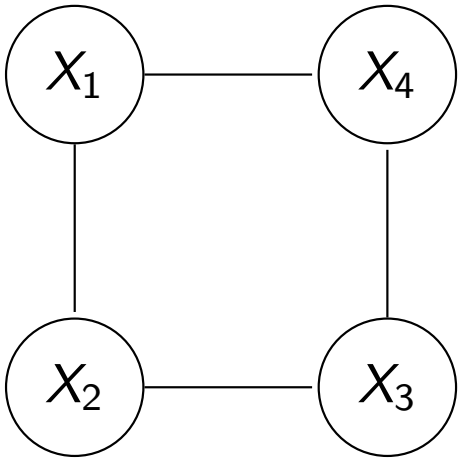
$$w_{12} = -W_{11}\theta_{12}/\theta_{22} = W_{11}\beta$$

- Substitute into the derivative $W_{11}\beta - s_{12} - \gamma_{12} = 0$
- we solve for the rows with zero γ: $\hat{\beta}^* = (W_{11}^*)^{-1} s_{12}^*$.
- The diagonal θ_{22} is (1 bottom right): $\frac{1}{\theta_{22}} = w_{22} - w_{12}^T \beta$.

Estimation of an Undirected Graphical Model Parameters

1: **procedure** GRAPHICAL REGRESSION:(S sample covariance)
2: $W \leftarrow S$ initialize
3: **repeat**
4: **for** $j = 1, 2, \ldots, p$ **do**
5: Partition W; jth row and column, W_{11} the rest
6: solve $W_{11}^{*}\beta^{*} - s_{12}^{*} = 0$ for reduced system
7: $\hat{\beta} \leftarrow \hat{\beta}^{*}$ by padding with zeros
8: update $w_{12} \leftarrow W_{11}\hat{\beta}$
9: **end for**
10: **until** convergence
11: **for** $j = 1, 2, \ldots, p$ **do**
12: lines 5:-8: above and set
13: $\hat{\theta}_{22} \leftarrow \frac{1}{w_{22} - w_{12}^{T}\hat{\beta}}$ ▷ the last row on previous slide
14: $\hat{\theta}_{12} \leftarrow -\hat{\beta} \cdot \hat{\theta}_{22}$ ▷ (3)
15: **end for**
16: **end procedure**

Example (ESLII)



$$W_0 = S = \begin{bmatrix} 10.00 & 1.00 & 5.00 & 4.00 \\ 1.00 & 10.00 & 2.00 & 6.00 \\ 5.00 & 2.00 & 10.00 & 3.00 \\ 4.00 & 6.00 & 3.00 & 10.00 \end{bmatrix}$$

$$W_{11} = \begin{bmatrix} 10.00 & 2.00 & 6.00 \\ 2.00 & 10.00 & 3.00 \\ 6.00 & 3.00 & 10.00 \end{bmatrix} \quad W_{22} = \begin{bmatrix} 10.00 & 1.16 & 4.00 \\ 1.16 & 10.00 & 3.00 \\ 4.00 & 3.00 & 10.00 \end{bmatrix}$$

$$W_{11}^* = \begin{bmatrix} 10.00 & 6.00 \\ 6.00 & 10.00 \end{bmatrix} \quad W_{22}^* = \begin{bmatrix} 10.00 & 1.16 \\ 1.16 & 10.00 \end{bmatrix}$$

$$W_{11}^{*,-1} = \begin{bmatrix} 0.156 & -0.094 \\ -0.094 & 0.156 \end{bmatrix} \quad W_{22}^{*,-1} = \begin{bmatrix} 0.101 & -0.012 \\ -0.012 & 0.101 \end{bmatrix}$$

$$\beta^* = [-0.22, 0.53]^T \quad \beta 2^* = [0.08, 0.19]^T$$

$$\beta = [-0.22, 0, 0.53]^T \quad \beta 2 = [0.08, 0.19, 0]^T$$

$$w_{12} \leftarrow [1.00, 1.16, 4.00]^T \quad w_{2r} \leftarrow [1.00, 2, 0.88]^T$$

Structure Learning

- We add a lasso penalty $||\Theta||_1$ which denotes the L_1 norm
  - the sum of the absolute values of the elements of Θ and we ignore the diagonal.
  - The negative penalized log-likelihood is a convex function of Θ.
- we maximize penalized log-likelihood

$$log|\Theta| - tr(\Theta S) - \lambda||\Theta||_1 \tag{4}$$

- the gradient equation is now

$$\Theta^{-1} - S - \lambda Sign(\Theta) = 0 \tag{5}$$

  - sub-gradient notation
  - $Sign(\theta_{jk}) = sign(\theta_{jk})$ for $\theta_{jk} \neq 0$
  - $Sign(\theta_{jk}) \in [-1, 1]$ for $\theta_{jk} = 0$
- the update for the first row and column will be

$$W_{11}\beta - s_{12} + \lambda Sign(\beta) = 0 \tag{6}$$

  - since β and θ_{12} have opposite signs.

procedure GRAPHICAL LASSO:(S sample covariance,λ penalty)
  $W \leftarrow S + \lambda I$ initialize
  repeat
    for $j = 1, 2, \ldots, p$ **do**
      Partition W; jth row and column, W_{11} the rest
      solve $W_{11}\beta - s_{12} + \lambda Sign(\beta) = 0$ using the cyclical
      ...coordinate-descent algorithm for the modified lasso
      update w_{12} by $W_{11}\hat{\beta}$
    end for
  until convergence
  for $j = 1, 2, \ldots, p$ **do**
    solve $\hat{\theta}_{22} \leftarrow \frac{1}{s_{22} - w_{12}^T \hat{\beta}}$
    solve $\hat{\theta}_{12} \leftarrow -\hat{\beta} \cdot \hat{\theta}_{22}$
  end for
end procedure
procedure COORDINATEDESCENT:($V \leftarrow W_{11}$)
  repeat $j = 1, 2, \ldots, p-1$
    $\hat{\beta}_j \leftarrow S(s_{12j} - \sum_{k \neq j} V_{kj}\hat{\beta}_k, \lambda)/V_{jj}$
  until convergence
end procedure

$$\#S(x,t) = sign(x)(|x| - t)_+$$

Example (glasso)

- $\lambda \leftarrow 1$

$$W_0 = S + \lambda I = \begin{bmatrix} 11.00 & 1.00 & 5.00 & 4.00 \\ 1.00 & 11.00 & 2.00 & 6.00 \\ 5.00 & 2.00 & 11.00 & 3.00 \\ 4.00 & 6.00 & 3.00 & 11.00 \end{bmatrix}$$

$$\begin{aligned} W_{11} &= \begin{bmatrix} 11.00 & 2.00 & 6.00 \\ 2.00 & 11.00 & 3.00 \\ 6.00 & 3.00 & 11.00 \end{bmatrix} \\ s_{12}^T &= \begin{bmatrix} 1.00 & 5.00 & 4.00 \end{bmatrix} \\ \beta^{T,(0)} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \\ V &\leftarrow W_{11} \\ \beta_2^{(1)} &= S(1 - 0,1)/11 = 0 \\ \beta_3^{(1)} &= S(5 - 0,1)/11 = \frac{4}{11} \\ \beta_4^{(1)} &= S(4 - \frac{3 \cdot 4}{11}, 1)/11 = \frac{21}{121} \end{aligned}$$

$$\beta_2^{(2)} = S(1 - \frac{2 \cdot 4}{11} - \frac{6 \cdot 21}{121}, 1)/11 \approx -0.16$$

$$\beta_3^{(2)} = S(5 + 0.32 - \frac{3 \cdot 21}{121}, 1)/11 \approx 0.35$$

$$\beta_4^{(2)} = \dots$$

$$\dots$$

$$\hat{\beta}_1 \approx [-0.22; 0.32; 0.30]$$

$$W_1 \approx \begin{bmatrix} 11.00 & 0.05 & 4.03 & 3.01 \\ 0.05 & 11.00 & 2.00 & 6.00 \\ 4.03 & 2.00 & 11.00 & 3.00 \\ 3.01 & 6.00 & 3.00 & 11.00 \end{bmatrix}$$

Graphical Lasso Properties

- Computational speed
  - The graphical lasso algorithm is extremely fast
  - can solve a moderately sparse problem with 1000 nodes in less than a minute.
  - It can be modified to have edge–specific penalty parameters λ_{jk}
  - setting $\lambda_{jk} = \infty$ will force $\hat{\theta}_{jk}$ to be zero
  - graphical lasso subsumes the parameter learning algorithm.
- Missing data
  - some missing observations may be imputed by EM algorithm from the model
  - latent – fully unobserved variables – do not bring more power in Gaussian graphical model
  - latent variables are very important in discrete distributions.

sklearn.covariance.graphical_lasso

Model Quality (Model Selection)

Definition (Saturated model, GGM Deviance, iDeviance, Likelihood Ratio Test)

- **saturated model** - full model with all edges, it has maximal loglikelihood
- **Deviance**

$$D = dev = 2 \cdot (\hat{\ell}_{sat} - \hat{\ell}) = N \log \frac{|S^{-1}|}{|\hat{K}|} = -N \log |S\hat{K}|$$

- **independent model** - no edges, it has minimal likelihood
- **iDeviance**

$$iD = idev = 2 \cdot (\hat{\ell} - \hat{\ell}_{ind}) = N \left(\log |\hat{K}| + \sum_{i=1}^{p} \log s_{ii} \right)$$

- **lrt likelihood ratio test** for models $\mathcal{M}_1 \subseteq \mathcal{M}_0$

$$lrt = 2 \cdot (\hat{\ell}_0 - \hat{\ell}_1) = N \log \frac{|\hat{K}_0|}{|\hat{K}_1|}.$$

Undirected Graphical Models and Their Properties

Definition (Undirected Graphical Model, Markov Graph)

An **Undirected Graphical Model** (Markov graph, Markov network) is a graph $\mathcal{G} = (V, E)$, where nodes V represent random variables and the absence of an edge (A, B) denoted $A \perp\!\!\!\perp_{\mathcal{G}} B$ implies that the corresponding random variables are conditionally independent given the rest in the probability distribution $P(V)$.

$$A \perp\!\!\!\perp_{\mathcal{G}} B \Longrightarrow A \perp\!\!\!\perp_{P} B | V \setminus \{A, B\}. \tag{7}$$

is known as the **pairwise Markov independencies** of $\mathcal{G}$.

Definition (Separators)

- If A, B and C are subgraphs, then C **is said to separate** A **and** B if every path between A and B intersects a node in C.
- C is called a **separator**.

- Separators break the graph into conditionally independent pieces.

Markov Properties

Definition (Global Markov Property)

A probability measure P over V is **(globally) Markov** with respect to an undirected graph $\mathcal{G}$ iff for any subgraphs A, B and C holds:

- if C separates A and B then the conditional independence $A \perp\!\!\!\perp_P B|C$ holds, that is

$$A \perp\!\!\!\perp_{\mathcal{G}} B|C \Longrightarrow P(A|C) \cdot P(B|C) = P(A, B|C). \tag{8}$$

Theorem

The pairwise and global Markov properties of a graph are equivalent for graphs with strictly positive distributions.

- Gaussian distribution is always positive.
- We may infer global independence relations from simple pairwise properties.
- The global Markov property allows us to decompose graphs into smaller more manageable pieces.

Markov Random Fields (Markovská náhodná pole)

- A probability density function f over a Markov graph $\mathcal{G}$ with the set of maximal cliques $\{C_1, \ldots, C_k\}$ can be represented as

$$f(x) = \prod_{i=1,\ldots,k} \psi_i(x_{C_i}) = \psi_1(x_{C_1}) \cdot \ldots \cdot \psi_k(x_{C_k}) \tag{9}$$

- where ψ_i are positive functions called **clique potentials**.
- they capture the dependence in X_{C_i} by scoring certain instances x_{C_i} higher than others.
- with the **normalizing constant** (partition function) Z

$$Z = \int_X exp\left(\sum_{i=1,\ldots,k} \log g_i(x_{C_i})\right).$$

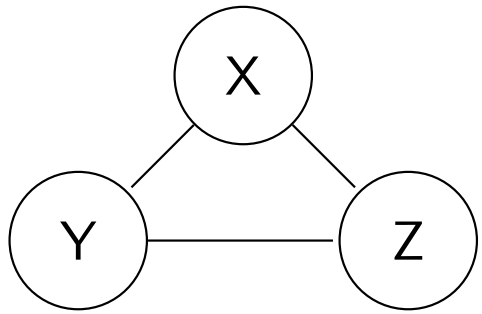
- Such set of random variables is called **Markov Random Field** or **Markov graph**. If the potentials represent conditional probabilities with respect to some observation, it is called a **Conditional Markov Random Field**.
- For Markov networks with positive distributions the probability density function (9) implies a graph with independence properties defined by the cliques in the product.

Pairwise Markov Graphs

- A graphical model does not always uniquely specify the higher–order dependence structure of ta joint probability distribution.

$$f^{(2)}(x,y,z) = \frac{1}{Z}\psi_1(x,y)\psi_2(x,z)\psi_3(y,z)$$

$$f^{(3)}(x,y,z) = \frac{1}{Z}\psi(x,y,z)$$



- For Gaussian distribution, pairwise interactions fully specify the model.
- We focus on **pairwise Markov Graphs**
  - where at most second order interactions are represented (like $f^{(2)}$).

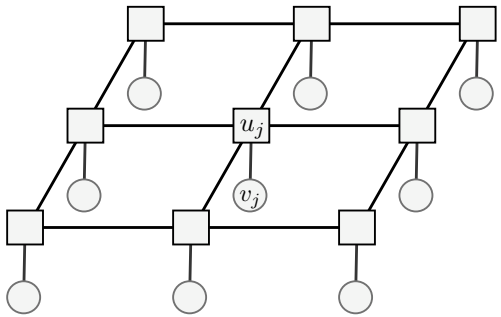
MRF for Image Denoising

- Given a noisy image v, perhaps with missing pixels, recover an image u, $u_{n,m} \in \mathbb{R}$ that is both smooth and close to v.
- Let each pixel be a node in a graph $\mathcal{G} = (V, E)$, with 4-connected neighborhood. Only pairwise interactions are present.
- We minimize the **energy function** (add missing margin on your own)

$$E(u) = \sum_{(m,n)\in P} (u_{m,n} - v_{m,n})^2 + \lambda \sum_{(m,n)\in P} \left[(u_{n+1,m} - u_{n,m})^2 + (u_{n,m+1} - u_{n,m})^2\right]$$

We can solve u iteratively

- $s_{m,n} = u_{n-1,m} + u_{n+1,m} + u_{n,m-1} + u_{n,m+1}$,
  - $u_{n,m}^{(t+1)} = \left\{ \begin{array}{ll} \frac{1}{1+4\lambda}(v_{n,m} + \lambda s_{n,m}^{(t)}) & \text{for } (n,m) \in v \\ \frac{1}{4} s_{n,m}^{(t)} & \text{for missing } v \end{array} \right\}$



- The goal is to find the signal u that minimizes the energy $E(u)$.
- Estimation by Max-flow/Min-Cut in a specific graph or Gibbs sampling.
- https://www.cs.toronto.edu/~fleet/courses/2503/fall11/Handouts/mrf.pdf

Definition (Ising Model, Boltzmann Machine (in ESLII))

- The **Ising model** is defined by a graph $\mathcal{G} = (\mathcal{X}, E)$ of binary variables $X_i \in \mathcal{X}$ and a set of parameters Θ. The joint probabilities are given by:

$$\begin{aligned} p(X, \Theta) &= e^{\sum_{(j,k)\in E} \theta_{jk} X_j X_k - \Phi(\Theta)} \quad \text{for } X \in \mathcal{X} \\ \Phi(\Theta) &= \log \sum_{x \in \mathcal{X}} \left[e^{\sum_{(j,k)\in E} \theta_{jk} x_j x_k} \right]. \end{aligned}$$

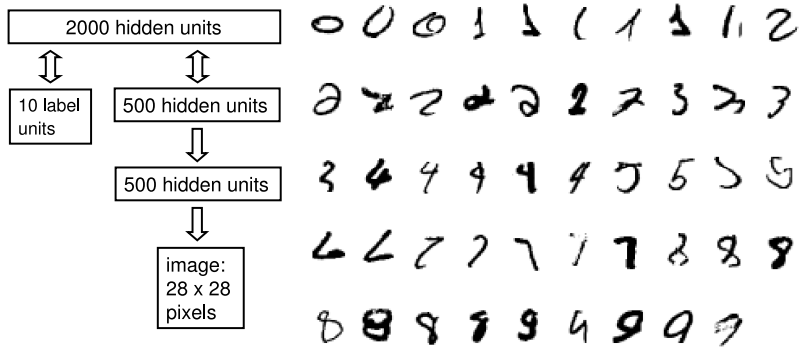
- it models only binary interactions (and unary)
- for technical reasons requires *constant* node $X_0 \equiv 1$ to be included.
- Originally from statistical mechanics.
- This model is equivalent to a first-order-interaction Poisson log-linear model for multiway tables of counts (Bishop et al., 1975).
- it implies a logistic form for each node conditional on the others

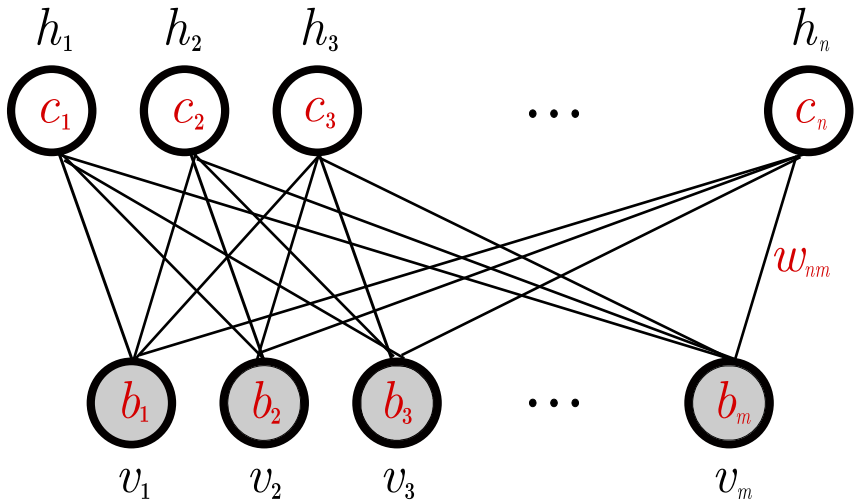
$$P(X_j = 1 | X_{-j} = x_{-j}) = \frac{1}{1 - \exp(-\theta_{j0} - \sum_{(j,k)\in E} \theta_{jk} x_k)}.$$

- Θ is fitted iteratively (Iterative proportional fitting, gradient descend, Poisson log-linear modeling, Mean field approximation, Gibbs sampling).

Restricted Boltzmann Machines (RBM)

- We have visible $\mathcal{V}$ and hidden $\mathcal{H}$ variables.
- 'restricted' means the variables are organized in two layers:
  - the hidden layer
  - the visible layer that is split to input $\mathcal{V}_1$ and output variables $\mathcal{V}_2$. (pixels of image/digit label).
  - no edges inside any layer
  - an edge between each hidden and visible variable.
- The structure enables faster parameter estimation.





$$E(v,h) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

Contrastive divergence

procedure CONTRASTIVE DIVERGENCE:(S batch for update)
  $\Delta w_{ij}, \Delta b_j, \Delta c_i \leftarrow 0$
  repeat
    for each training sample in a batch S **do**
      Sample $\mathcal{H}$ given $\mathcal{V}_1$, $\mathcal{V}_2$
      Sample $\mathcal{V}_1^{(last)}$, $\mathcal{V}_2^{(last)}$ given $\mathcal{H}$
      Sample $\mathcal{H}$ given $\mathcal{V}_1^{(last)}$, $\mathcal{V}_2^{(last)}$
      $\Delta w_{ij} \leftarrow \Delta w_{ij} + p(H_i = 1|v^{(0)}) \cdot v_j^{(0)} - p(H_i = 1|v^{(last)}) \cdot v_j^{(last)}$
      $\Delta b_j \leftarrow \Delta b_j + v_j^{(0)} - v_j^{(last)}$
      $\Delta c_i \leftarrow \Delta c_i + p(H_i = 1|v^{(0)}) - p(H_i = 1|v^{(last)})$
    end for
  until convergence
  return Δw_{ij}, Δb_j, Δc_i to adjust the parameters.
end procedure

Fischer, A., Igel, C. (2012). An Introduction to Restricted Boltzmann Machines. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2012. Lecture Notes in Computer Science, vol 7441. Springer, Berlin, Heidelberg

Mixed interaction models

- Discrete and Gaussian variables together.
- *Conditional Gaussian density* for $x = (i, y)$, i is the list of discrete variables, y is the list of continuous variables.
- **Directed graphs**: a discrete child of a continuous parent is not allowed.
- **Undirected graphs**: If there is a path between two discrete variables A, B, then they are connected by a path without any continuous variable.
- $f(i, y) = p(i)(2\pi)^{-\frac{q}{2}}|\Sigma|^{-\frac{1}{2}} \exp(-\frac{1}{2}(y - \mu(i))\Sigma^{-1}(y - \mu(i)))$
- The parameters $p(i), \mu(i), i \in \mathcal{I}, \Sigma$ are called **moment parameters**
- in the exponential form we get

$$\begin{aligned} f(i, y) &= \exp\left\{g(i) + h(i)^T y - \frac{1}{2} y^T K y\right\} \\ &= \exp\left\{g(i) + \Sigma_u h_u(i) y_u - \frac{1}{2}\Sigma_{u,v} K_{u,v} y_u y_v\right\} \end{aligned}$$

- parameters $g(i), h(i), i \in \mathcal{I}, K$ are called **canonical parameters**.

Mixed interaction models

- A marginal distribution is not necessarily a conditional Gaussian distribution
- it is a mixture of conditional Gaussians
- it is still tractable for evaluation
- and learning.

- Is this possible for other kind of distributions?

Markov Properties (Zeros are dangerous)

Definition (Markov properties: Global, Local, Pairwise)

Let G be an undirected graph over V, let P be a probability measure P over V.

(GM) P is **(globally) Markov** with respect to $\mathcal{G}$ iff

$$\forall(\mathcal{A}, \mathcal{B} \in V, \mathcal{C} \subseteq V)\ \mathcal{A} \perp\!\!\!\perp_{\mathcal{G}} \mathcal{B}|\mathcal{C} \Rightarrow \mathcal{A} \perp\!\!\!\perp_{P} \mathcal{B}|\mathcal{C} \text{ in P.}$$

(LM) A probability measure has the **local Markov property** iff
$(\forall A \in V):\ A \perp\!\!\!\perp_{P} V \setminus Fa_A | N_A$

(PM) P has the **pairwise Markov property** iff $\forall A, B \in V, A \neq B$ not connected by an edge holds $A \perp\!\!\!\perp_{P} B | V \setminus \{A, B\}$.

Theorem

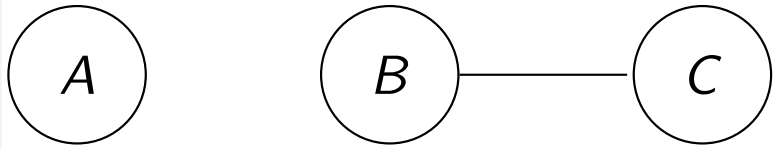
These properties are equivalent for strictly positive measures.

Counterexamples for measures with zero probability everywhere except $(0,0,0)$ and $(1,1,1)$.
See [Milan Studený:*Struktury podmíněné nezávislosti, Matfyzpress 2014*].

Examples

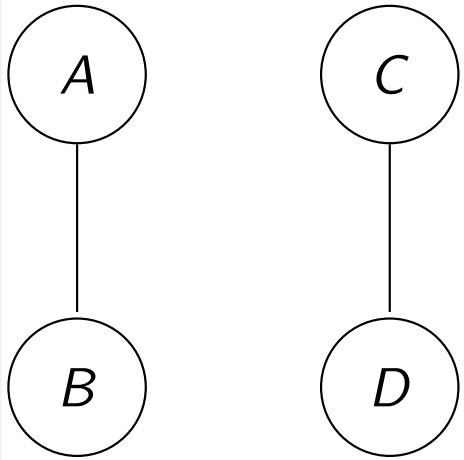
Example (P has the pairvise but not the local property)

$V = \{A, B, C\}, E = \{(b, c)\}$. Let us have a binary probability measure V nonzero at points $(0, 0, 0)$ and $(1, 1, 1)$ [Studený p.101].
$A \perp\!\!\!\perp B|\{C\}$
$A \perp\!\!\!\perp C|\{B\}$ & does not imply $A \perp\!\!\!\perp BC|\{\}$.



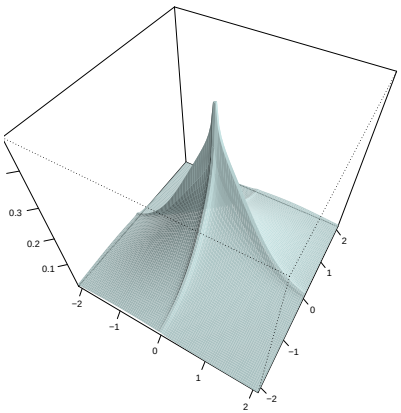
Example (P has the local but not the global property)

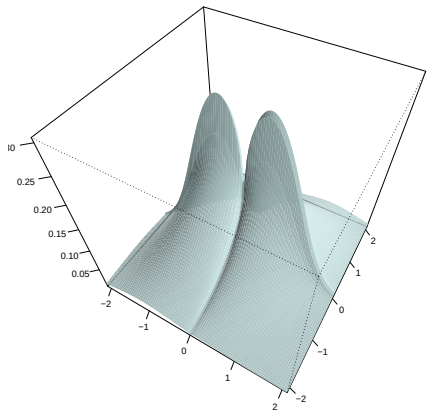
$V = \{A, B, C, D\}, E = \{(a, b), (c, d)\}$. Let $P(V)$ be nonzero only at points $(0, 0, 0, 0)$ and $(1, 1, 1, 1)$ [Studený p.101].
$A \perp\!\!\!\perp CD|\{B\}$
$B \perp\!\!\!\perp CD|\{A\}$
$C \perp\!\!\!\perp AB|\{D\}$ & does not imply $A \perp\!\!\!\perp C|\{\}$.
$D \perp\!\!\!\perp AB|\{C\}$

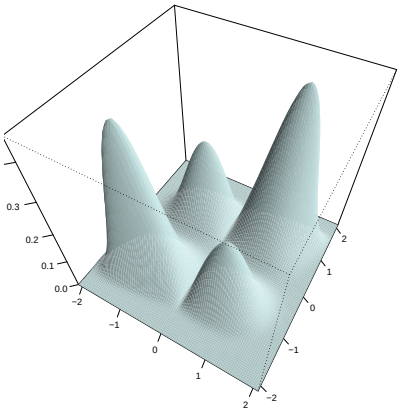


Nonparanormal Graphical Models

- A continuous pairwise interaction model.
- We model marginal distributions,
- and the most important relations by gaussian copula.
- https://www.stat.cmu.edu/~larry/=sml/GraphicalModels.pdf







List of topics

1. Linear, ridge, lasso regression, k-neares neighbours,(formulas) overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, generalized additive models
4. Train/test error and data split, square error, 0-1, crossentropy, AIC, BIC,(formulas) crossvalidation, one-leave-out CV, wrong estimate example
5. decision trees, information gain/entropy/gini, CART prunning,(formulas)
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost(formulas) and gradient boosting), stacking, MARS,
7. Bayesian learning: MAP, ML hypothesis (formulas), Bayesian optimal prediction, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, deviance, MRF
12. Gaussian processes: estimation of the function and its variance (figures, ideas).

List of topics

1. Linear, ridge, lasso regression, k-neares neighbours,(formulas) overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, generalized additive models
4. Train/test error and data split, square error, 0-1, crossentropy, AIC, BIC,(formulas) crossvalidation, one-leave-out CV, wrong estimate example
5. decision trees, information gain/entropy/gini, CART prunning,(formulas)
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost(formulas) and gradient boosting), stacking, MARS,
7. Bayesian learning: MAP, ML hypothesis (formulas), Bayesian optimal prediction, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, deviance, MRF
12. Gaussian processes: estimation of the function and its variance (figures, ideas).

Table of Contens