

# LECTURE 8: ADVANCED BAYESIAN CONCEPTS

- Probabilistic graphical models (Bayesian networks)
- Hierarchical Bayesian models
- Motivation: we want to write down the probability of the data  $d$  given some parameters  $\theta$  we wish to determine. But the relation between the two is difficult to write in a closed form. For example, the parameters determine some probability distribution function (PDF) of perfect data  $x$ , but what we measure is  $d$ , a noisy version of  $x$ , and noise is varying between measurements.

# LECTURE 8: ADVANCED BAYESIAN CONCEPTS

- We can introduce  $x$  as latent variables and model them together with  $\theta$ . Then  $\theta$  can be viewed as hyperparameters for  $x$ . The advantage is that at each stage PDF is easier to write down. However, we now have a lot of parameters to determine, most of which we do not care about.
- Modern trend in statistics is to use the hierarchical modeling approach, enabled by advances in MC, specially HMC.
- We can also try to marginalize over  $x$  analytically: convolve true PDF with noise PDF and do this for each measurement. This works, but requires doing the convolution integrals. The advantage is fewer variables, just  $\theta$ .

# Graphical Models for Probabilistic and Casual Reasoning

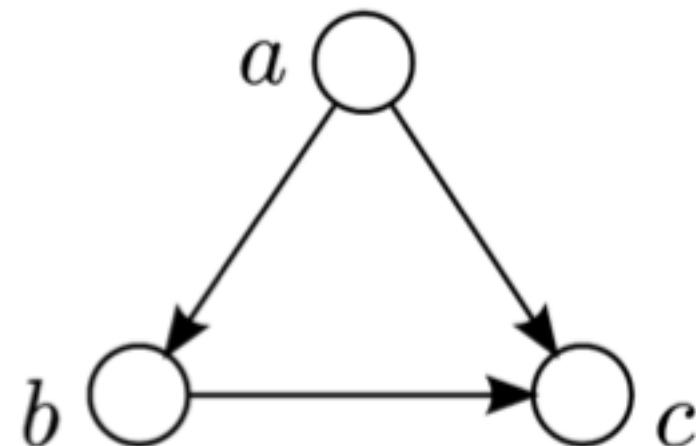
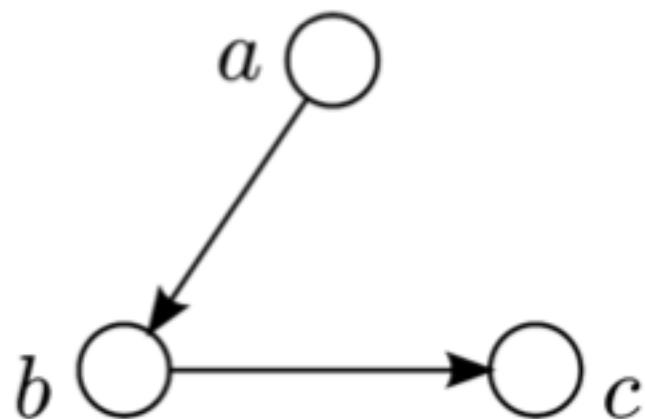
- We would like to describe the causal flow of events such that we can generate (simulate) events in a probabilistic setting (a flowchart of generating data)
- We can describe this with **directed acyclic graphs (DAG)**
- Typically we divide the process into components each of which generates a single variable  $x$  (given all other variables), which we can generate using random number generator for  $p(x)$
- We can also use the same process to describe inference of latent (unobserved) variables from data
- This also goes under the name of **Bayesian networks** and **probabilistic graphical models (PGM)**

# Approach of Bayesian Networks/PGMs

- We infer the causal (in)dependence of variables
- Write factorized joint probability distributions
- Perform data analysis by posterior inference

## PGM Rules

- Each circle is a probability distribution for the variable inside it
- Each arrow is a conditional dependence

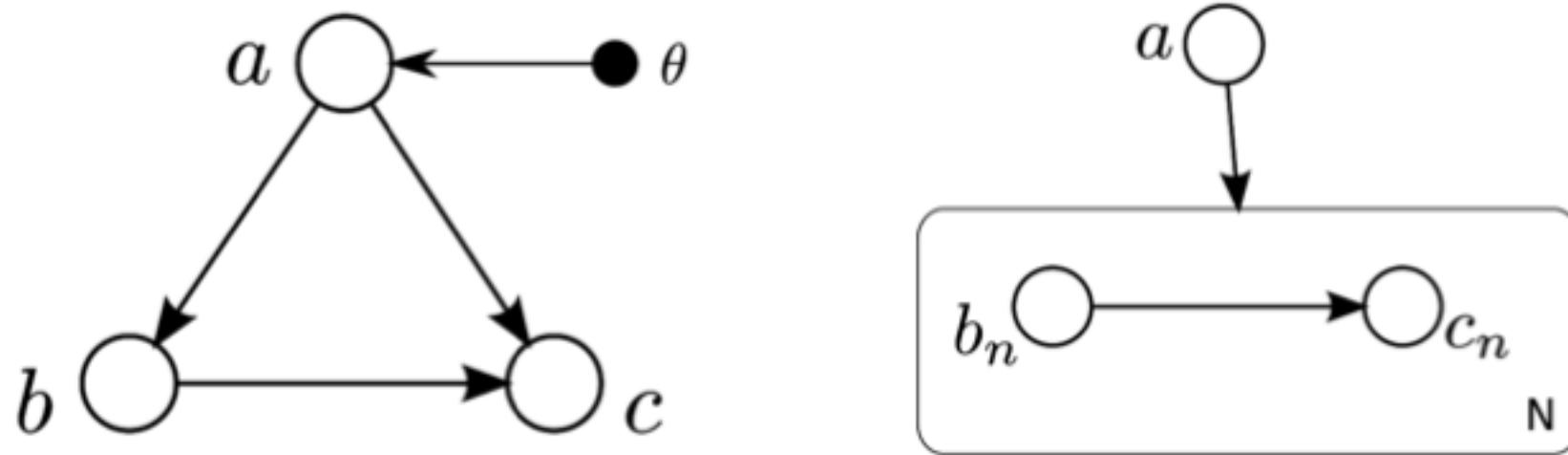


$$p(a, b, c) = p(c|b)p(b|a)p(a)$$

$$p(a, b, c) = p(c|a, b)p(b|a)p(a)$$

# PGM Rules

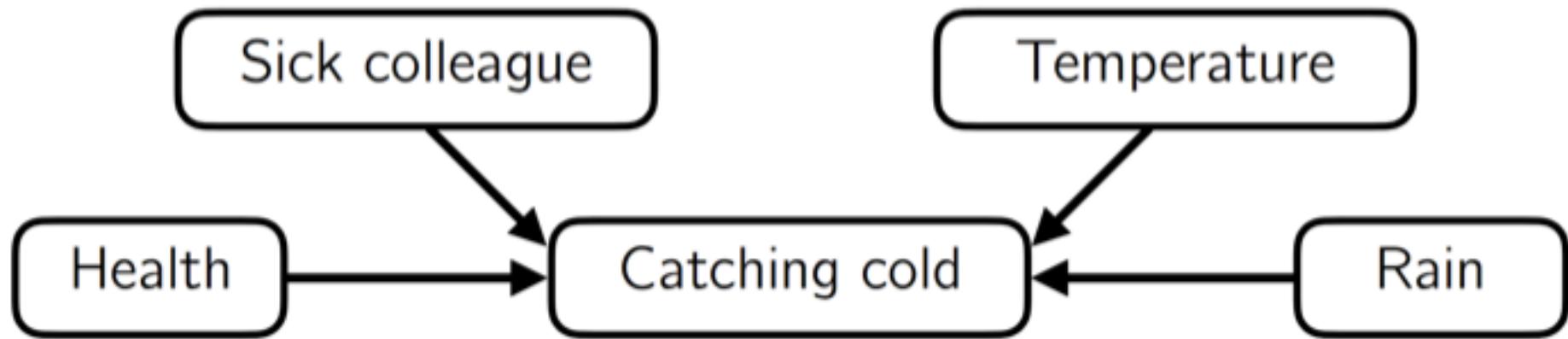
- Each solid point is a fixed variable (pdf is a delta function)
- Each plate contains conditionally independent variables: repetition, compressed notation for many nodes



$$p(a, b, c) = p(c|a, b)p(b|a)p_\theta(a)$$

$$p(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \prod_{n=1}^N [p(c_n|a, b_n)p(b_n|a)] p(a)$$

# Breaking causality down into components

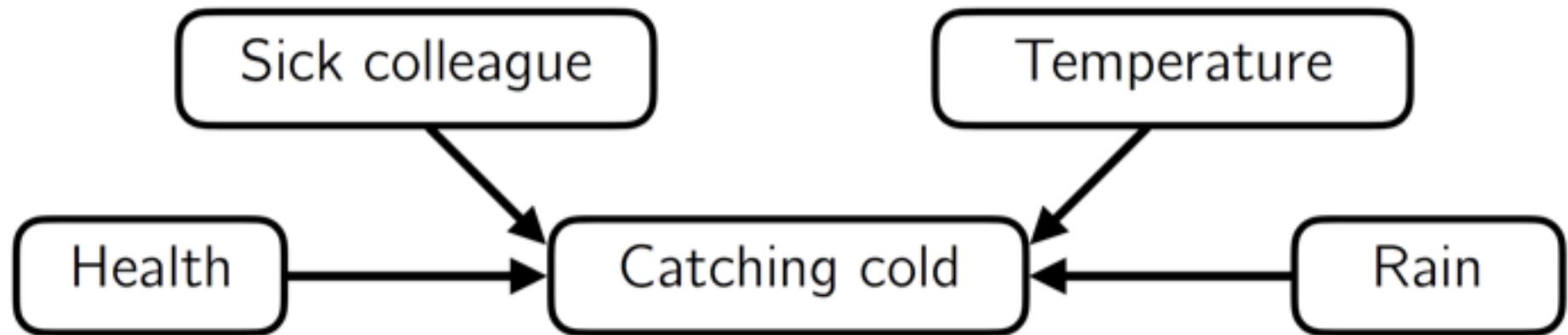


Independent causes of Catching cold

$\text{Health} \perp\!\!\!\perp \text{Sick colleague} \perp\!\!\!\perp \text{Temperature} \perp\!\!\!\perp \text{Rain}$

Credit: Slides from B. Leistedt

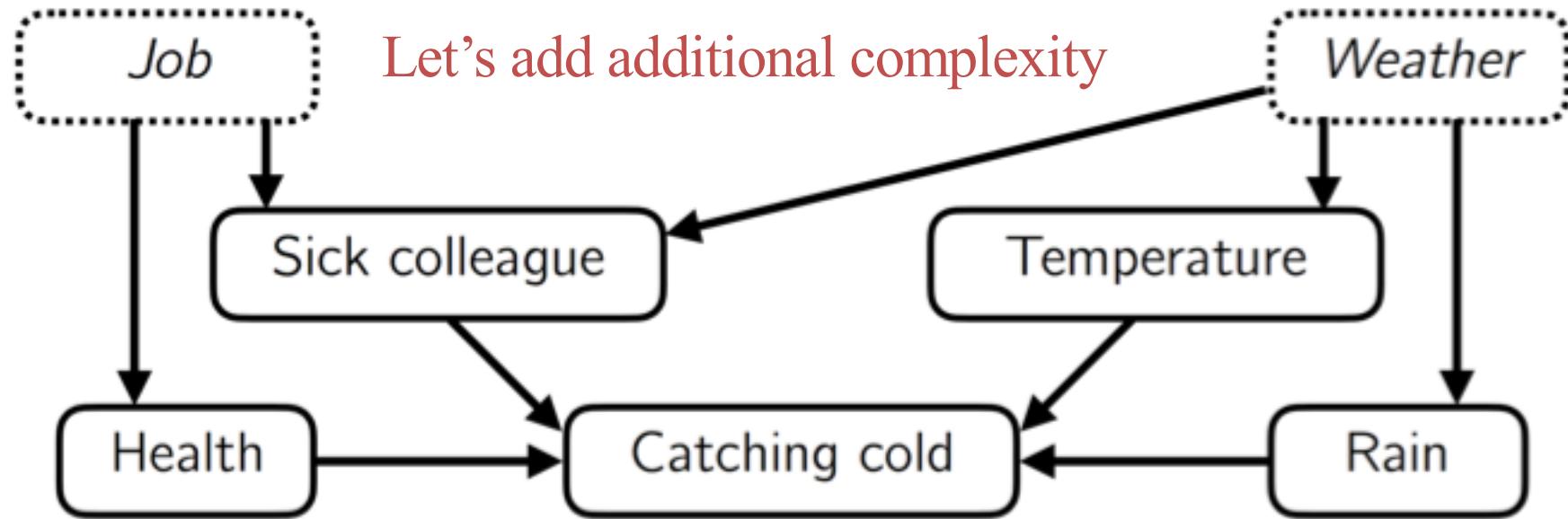
# Breaking causality down into components



$$p(C, S, H, T, R) = p(C | S, H, T, R) p(S, H, T, R)$$

$$p(S, H, T, R) = p(S) p(H) p(T) p(R)$$

Credit: Slides from B. Leistedt



$$\begin{aligned}
 p(C, S, H, T, R, J, W) &= p(C | S, H, T, R, J, W) \\
 &\times p(S, H, T, R | J, W) \times p(J, W)
 \end{aligned}$$

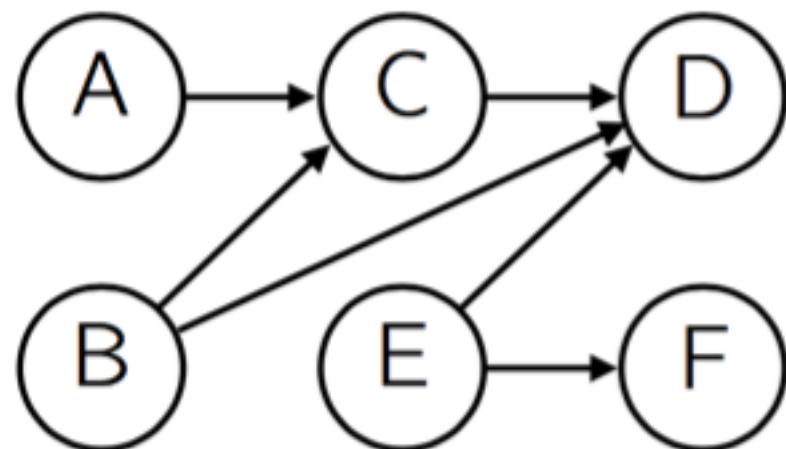
$$p(C | S, H, T, R, J, W) = p(C | S, H, T, R)$$

$$p(S, H, T, R | J, W) = p(S|J, W) \ p(H|J) \ p(T|W) \ p(R|W)$$

$$p(J, W) = p(J)p(W)$$

## Example

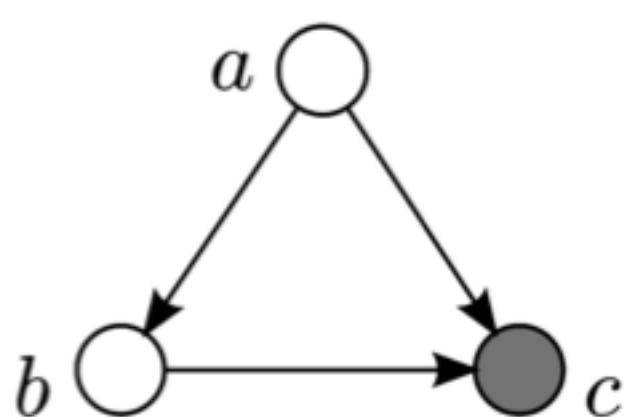
- Write down corresponding probability expressions for this graph and discuss the meaning with your neighbor



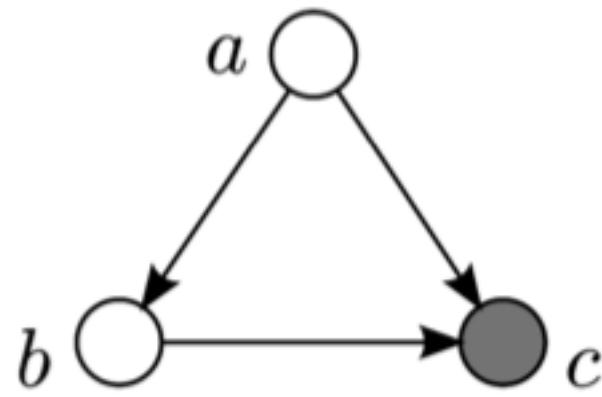
$$\begin{aligned} p(A, B, C, D, E, F) = \\ \times p(F|E) p(D|B, E, C) \\ \times p(E) p(C|A, B) p(A) p(B) \end{aligned}$$

# PGM Rules: Observables and Inference

- Each shaded (or double) circle implies an observable ( $c$ ), everything else ( $a, b$ ) is not an observable, but a latent (hidden) variable
- If we want to determine latent variables ( $a, b$ ) from observables  
we do posterior inference (inverse problem requires Bayes rule)



$$p(a, b, c) = p(c|a, b)p(b|a)p(a)$$

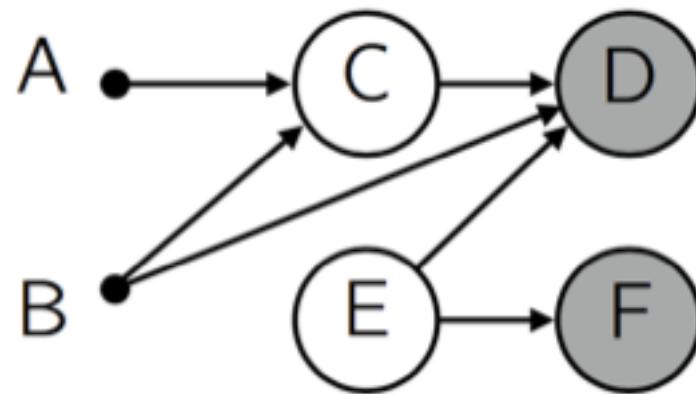


$$\underbrace{p(a, b|c)}_{\text{posterior}} \propto \underbrace{p(c|a, b)}_{\text{likelihood}} \underbrace{p(b|a)p(a)}_{\text{prior}}$$

11

# Posterior Inference

- Here D, F are data
- C, E parameters
- A, B fixed parameters



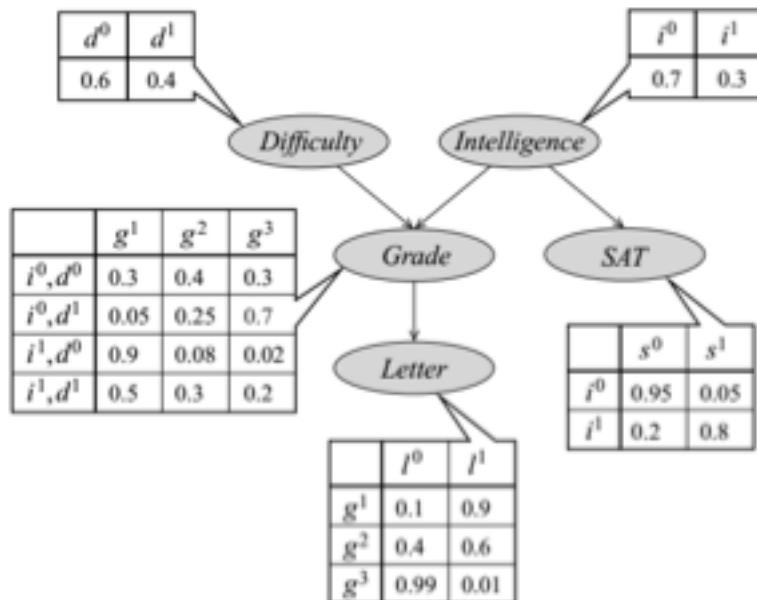
$$\begin{aligned} P(\text{Parameters} | \text{Data, Model}) &= p(C, E | D, F, A, B) \\ &= p(F|E) p(D|C, E) p(C | A, B) p(E) p(A) p(B) \end{aligned}$$

- We need all these conditional PDFs (probability distribution functions):  $p(F|E)$ ,  $p(D|C,E)$ ,  $p(C|A,B)$ ,  $p(E)$ , note that  $p(A)$  and  $p(B)$  are delta functions (fixed parameters)

- An upper case non-bold letter indicates a single random variable ('RV'). The same letter lower cased with a super script indicates a specific value that RV may take. For example,  $X = x^1$  is the *event* the RV  $X$  took on the value  $x^1$ . We call this event an **assignment**. The set of unique values an RV may take is  $Val(X)$ . So we might have  $Val(X) = \{x^0, x^1\}$  in this case.
- A bold upper case letter indicates a *set* of RVs (like  $\mathbf{X}$ ) and a bold lower case letter indicates a set of values they may take. For example, we may have  $\mathbf{X} = \{A, B\}$  and  $\mathbf{x} = \{a^3, b^1\}$ . Then the event  $\mathbf{X} = \mathbf{x}$  is the event that  $A = a^3$  happens *and*  $B = b^1$  happens. Naturally,  $Val(\mathbf{X})$  is the set of all possible unique joint assignments to the RVs in  $\mathbf{X}$ .
- If you see  $\mathbf{x}$  (or  $\mathbf{y}$  or  $\mathbf{z}$  etc...) within a probability expression, like  $P(\mathbf{x} | \dots)$  or  $P(\dots | \mathbf{x})$ , that's *always* an abbreviation of the event ' $\mathbf{X} = \mathbf{x}$ '.
- Perhaps confusingly, we also abbreviate the event ' $\mathbf{X} = \mathbf{x}$ ' as ' $\mathbf{X}$ ', though this isn't a clean abbreviation. Omission of  $\mathbf{x}$  means one of two things: either we mean this for *any* given  $\mathbf{x}$  or for *all* possible  $\mathbf{x}$ 's. As an example for the latter case, 'calculate  $P(\mathbf{X})$ ' would mean calculate the set of probabilities  $P(\mathbf{X} = \mathbf{x})$  for all  $\mathbf{x} \in Val(\mathbf{X})$ .
- $\sum_{\mathbf{X}} f(\mathbf{X})$  is shorthand for  $\sum_{\mathbf{x} \in Val(\mathbf{X})} f(\mathbf{X} = \mathbf{x})$ . This is similarly true for  $\prod_{\mathbf{X}} (\cdot)$  and  $\operatorname{argmin}_{\mathbf{X}} (\cdot)$ . Look out for this one - it can sneak in there and change things considerably.

# Bayesian network: conditional independence

- Conditional independence



Given subsets of RVs  $X$ ,  $Y$  and  $Z$  from  $\mathcal{X}$ , we say  $X$  is conditionally independent of  $Y$  given  $Z$  if

$$P(x, y|z) = P(x|z)P(y|z)$$

for all  $x \in Val(X)$ ,  $y \in Val(Y)$  and  $z \in Val(Z)$ . This is stated as ' $P$  satisfies  $(X \perp Y|Z)$ '

$$P_B(I, D, G, S, L) = P_B(I)P_B(D)P_B(G|I, D)P_B(S|I)P_B(L|G)$$

So we would calculate a given assignment as:

$$\begin{aligned} P_B(i^1, d^0, g^2, s^1, l^0) &= P_B(i^1)P_B(d^0)P_B(g^2|i^1, d^0)P_B(s^1|i^1)P_B(l^0|g^2) \\ &= 0.3 \cdot 0.6 \cdot 0.08 \cdot 0.8 \cdot 0.4 \\ &= 0.004608 \end{aligned}$$

The BN graph, just those nodes and edges, implies a set of CI statements regarding its accompanying  $P_B$ .

$$P_B(X_1, \dots, X_n) = \prod_{i=1}^n P_B(X_i | \text{Pa}_{X_i}^G) \quad \text{Chain rule}$$

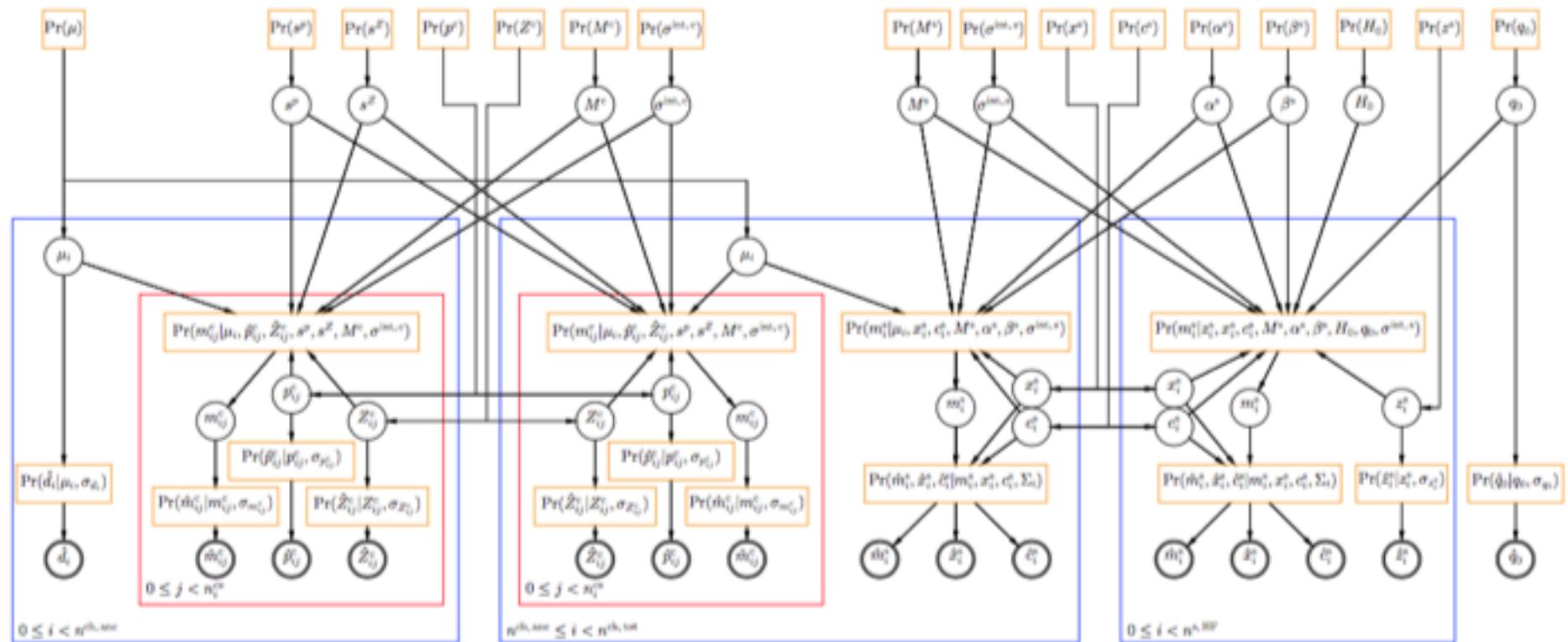
$$\begin{aligned} &(L \perp I, D, S|G) \\ &(S \perp D, G, L|I) \\ &(G \perp S|I, D) \\ &(I \perp D) \\ &(D \perp I, S) \end{aligned}$$

Example from Koller & Friedman

14

# A Big PGM (Distance Ladder)

- This is a way to organize the generative probabilistic model

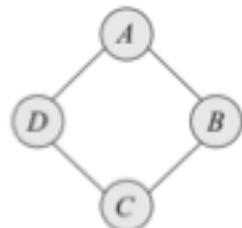


# Markov Network

- Not every probabilistic model can be represented by BN
- Markov networks: edges are undirected
- We divide the graph into complete subgraphs
- simple CI

Gibbs rule

Graph Representation



Independencies

$$\begin{aligned}(A \perp C \mid B, D) \\ (B \perp D \mid A, C)\end{aligned}$$

$$P(A, B, C, D) = \frac{1}{Z} \phi_1(A, B) \phi_2(B, C) \phi_3(C, D) \phi_3(D, A)$$

where

$$Z = \sum_{x \in Val(\mathcal{X})} \phi_1(A, B) \phi_2(B, C) \phi_3(C, D) \phi_3(D, A)$$

$$P_M(X_1, \dots, X_n) = \frac{1}{Z} \underbrace{\prod_{i=1}^n \phi_i(\mathbf{D}_i)}_{\text{we call this } \tilde{P}_M(X_1, \dots, X_n)}$$

- Direct analogy with statistical physics: e.g. Ising spin model
- $\Phi_i$  are Boltzmann factors  $\exp(-H_i)$ , describing interaction energy  $H_i$  between neighboring spins,  $P_M$  is Boltzmann factor for total energy,  $Z$  is partition function, i.e. the sum of the Boltzmann factors over all configurations

16

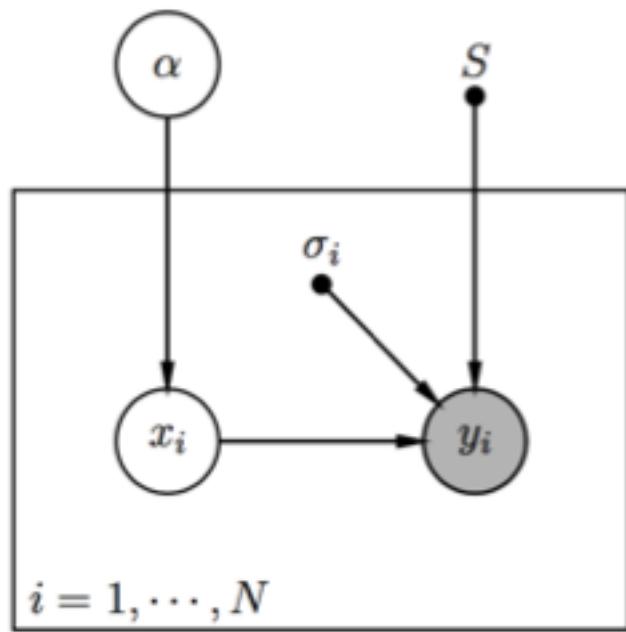
# Hierarchical Bayesian Models

- PGMs encode a hierarchical causal structure: D depends on C which depends on A
- In many problems we have hierarchical structure of parameters
- For example, we measure some data  $d$ , which are noisy and related to some underlying true values  $x$ , but what we want is the parameters that determine their distribution  $\theta$ .
- $d$ : **observable**
- Variables that are not observed are called **latent variables**:  $\theta, x$
- Variables we do not care about are called *nuisance variables*:  $x$ . We want marginalize over them to determine  $\theta$

# Exchangeability

- When we do not know anything about latent variables  $x_i$  we can place them on equal footing:  $p(x_1, x_2, \dots, x_J)$  is invariant under permutation of  $(1, 2, \dots, J)$  indexes.
- Their joint probability distribution cannot change upon exchanging  $x_i$  with  $x_j$  ...
- A simple way to enforce this is to say  $p(x_1, x_2, \dots, x_J) = \prod_{j=1}^J p(x_i | \theta)$
- This does not always work (e.g. a die has 6 exchangeable  $x_i$ , but their values must add to 1), but works in large  $J$  limit (de Finetti theorem).

## Example

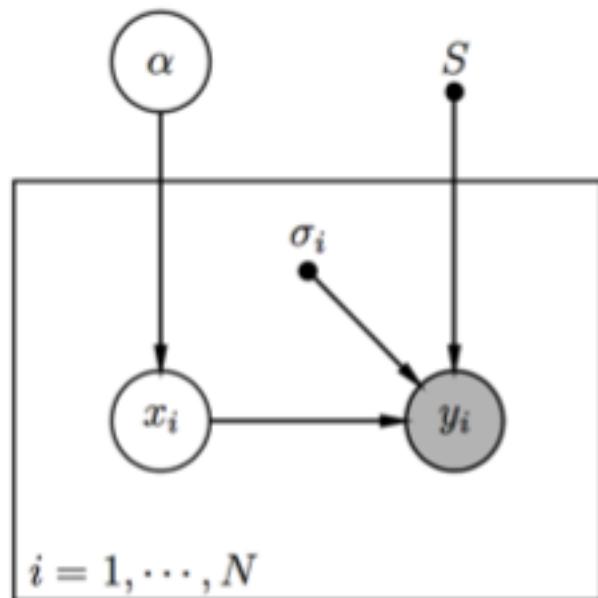


- ▶  $y_i$ 's : measurements of the temperature in a room
- ▶  $\sigma_i$  : Gaussian noise
- ▶  $x_i$ 's : true temperature
- ▶  $\alpha$  : parameter parametrizing the true distribution of temperatures.
- ▶  $S$  : some selection effect (e.g., no measurement if temperature is < 0 degrees)

# Marginalization over Latent Variables

We are interested in inferring  $\alpha$

We need to **marginalize over** the latent  $x_i$ 's, numerically or analytically



$$p(\alpha | \{y_i, \sigma_i\})$$

$$\propto \prod_i p(y_i | \alpha, \sigma_i) p(\alpha)$$

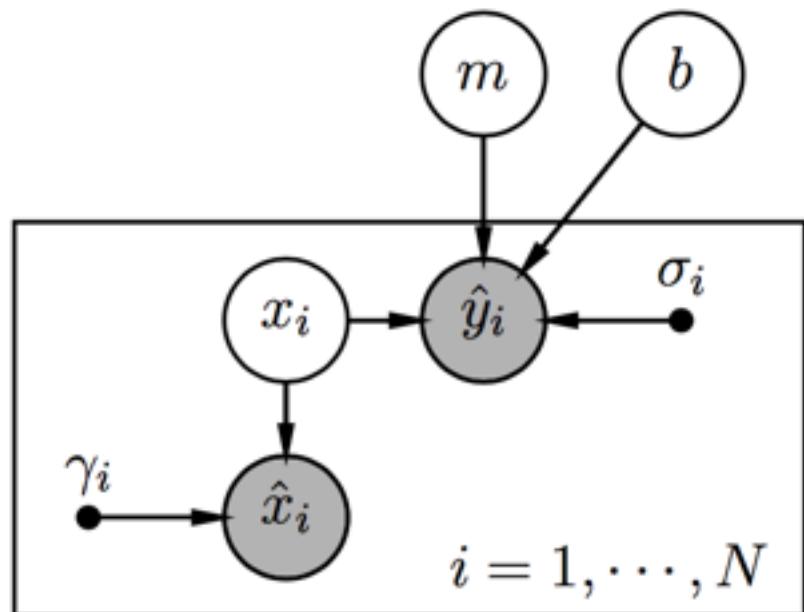
$$= \prod_i \int p(y_i, x_i | \alpha, \sigma_i) p(\alpha) dx_i$$

$$= \prod_i \int p(y_i | x_i, \alpha, \sigma_i) p(x_i | \alpha, \sigma_i) p(\alpha) dx_i$$

$$= \prod_i \int p(y_i | x_i, \sigma_i) p(x_i | \alpha) p(\alpha) dx_i$$

## Additional Complication: Noise in $x$

- ▶ We now observe noisy versions of the  $x_i$ 's, with known Gaussian noises  $\gamma_i$ 's
- ▶ The  $x_i$ 's are now latent parameters. They need to be estimated or marginalized over.

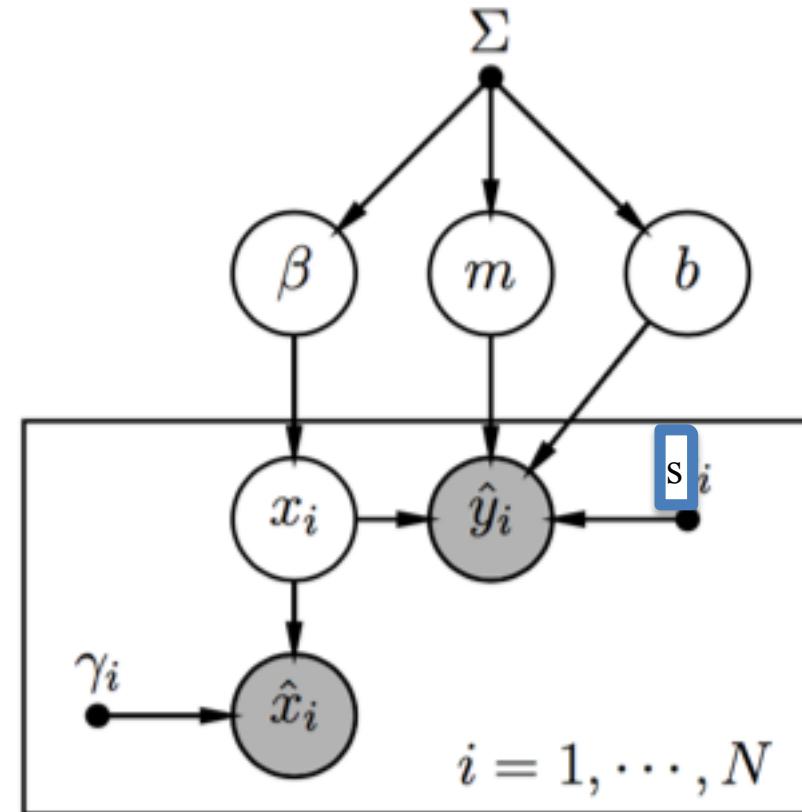


## Sometimes marginalization can be done analytically

$$\begin{aligned} & p(m, s | \{\hat{y}_i, \hat{x}_i, \sigma_i, \gamma_i\}) \\ &= \int d\{x_i\} p(m, s, \{x_i\} | \{\hat{y}_i, \hat{x}_i, \sigma_i, \gamma_i\}) \\ &\propto \prod_{i=1}^N \int dx_i \mathcal{N}(\hat{y}_i - mx_i - b; \sigma_i^2) \mathcal{N}(\hat{x}_i - x_i; \gamma_i^2) p(\{x_i\}, m, s) \\ &\propto \prod_{i=1}^N \mathcal{N}(\hat{y}_i - m\hat{x}_i - b; \sigma_i^2 + \gamma_i^2) p(s, m) \end{aligned}$$

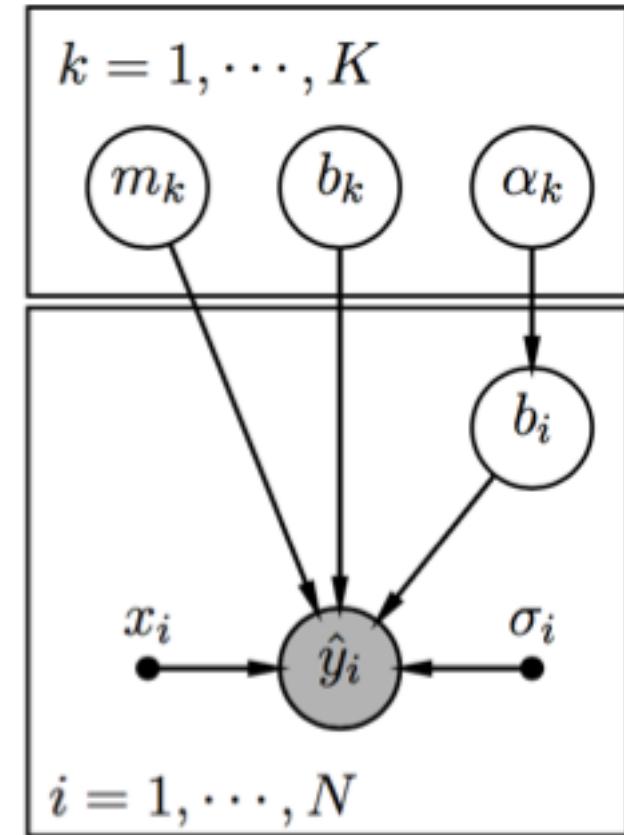
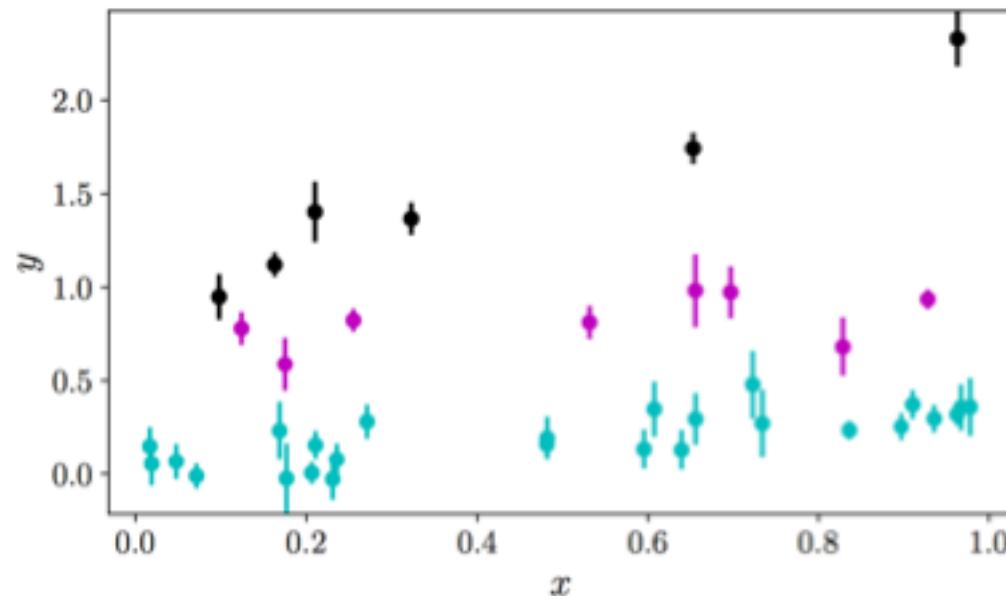
# We should also put hyperpriors onto parameters

- We have hyperprior  $\Sigma$ : each parameter should have a prior, which can be non-informative
- A proper PGM should start with hyperpriors and end with observables



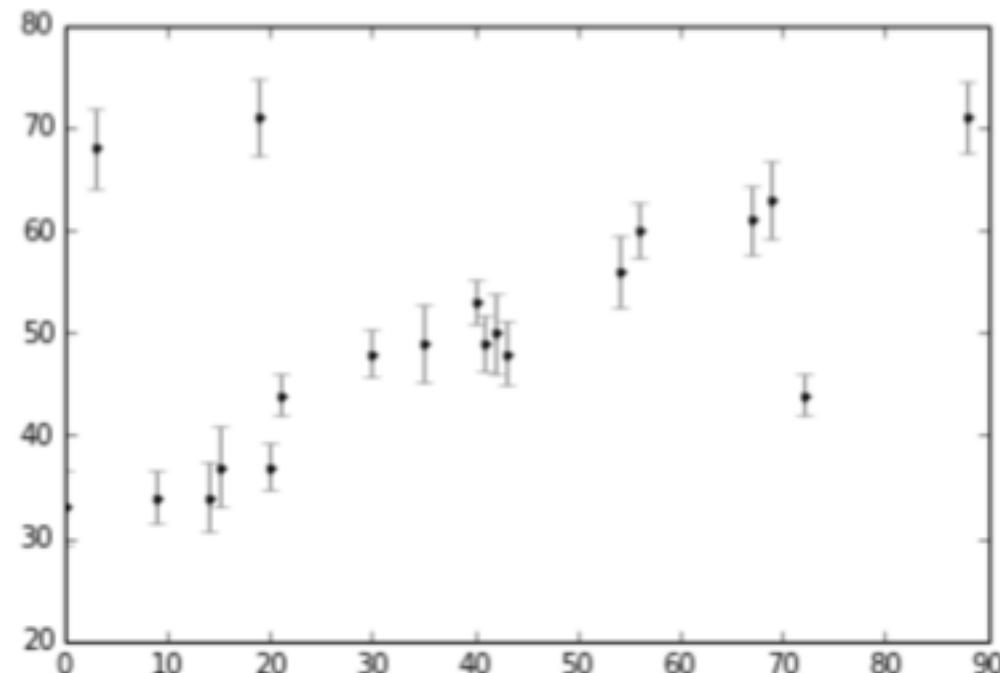
## Another Extension: Mixture Models

- Mixture models try to fit the data with a mixture of components
- For example, we can fit multiple lines to the data assuming the data are drawn from one of the components



# Mixture Model for Outliers

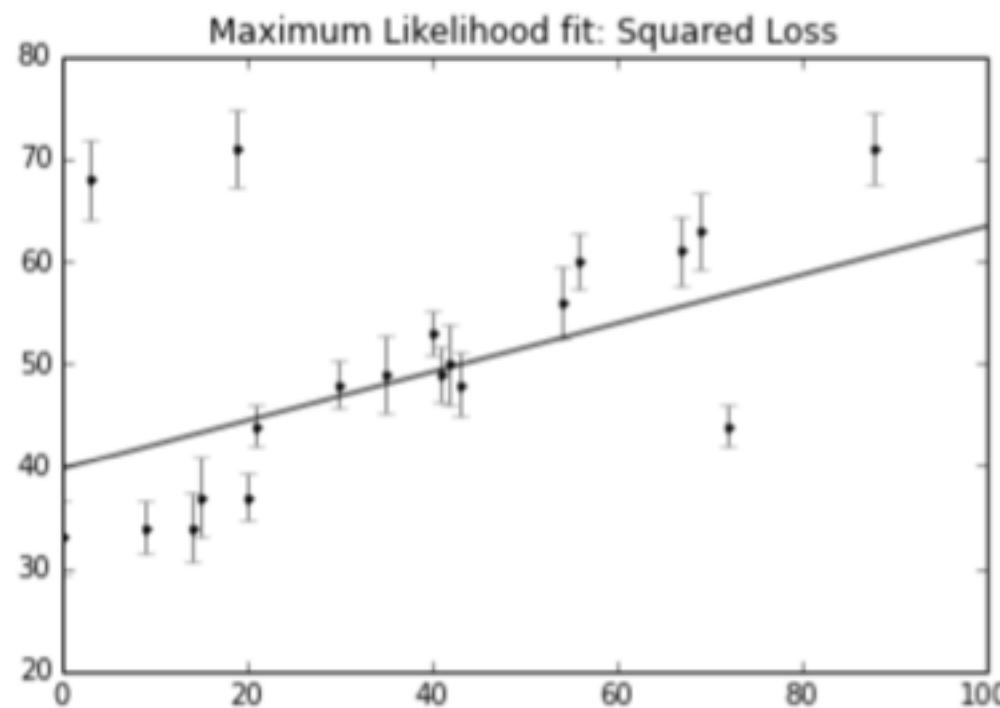
- Suppose we have data that can be fit to a linear regression, apart from a few outlier points
- It is always better to understand the underlying generative model of outliers
- But suppose we just want to identify them



## Let us model this as a Gaussian

$$p(x_i, y_i, e_i \mid \theta) \propto \exp\left[-\frac{1}{2e_i^2} (y_i - \hat{y}(x_i \mid \theta))^2\right]$$

- We get a poor fit to the data, goodness of fit is poor (we will discuss more formally what that means in the next lecture)



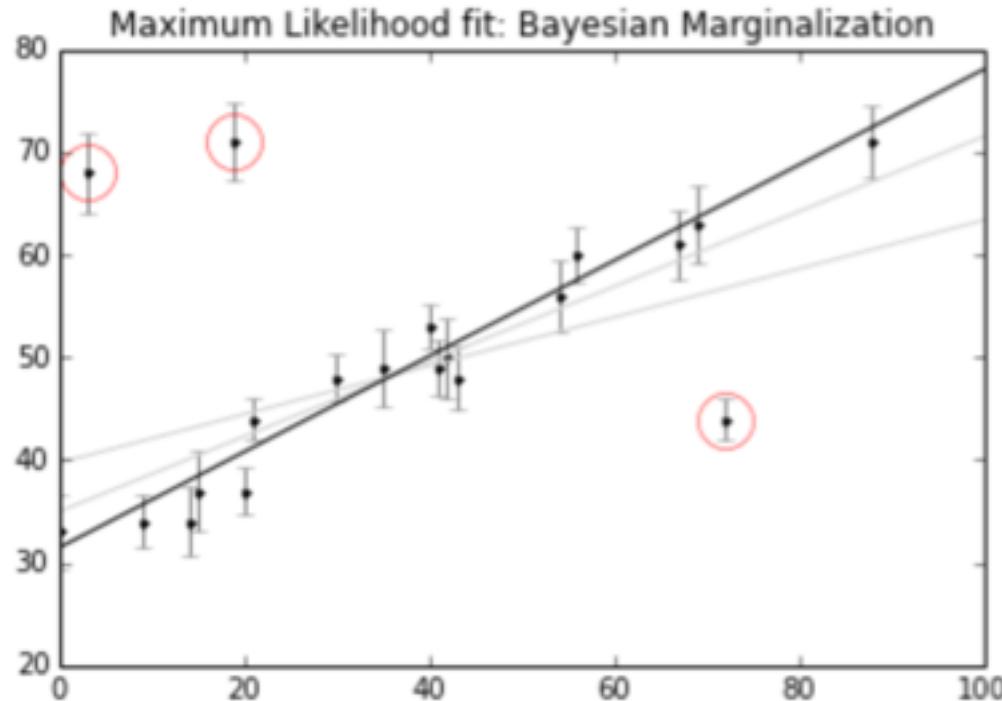
26

# Let us model this as a double Gaussian

$$p(\{x_i\}, \{y_i\}, \{e_i\} | \theta, \{g_i\}, \sigma, \sigma_b) = \frac{g_i}{\sqrt{2\pi e_i^2}} \exp\left[\frac{-(\hat{y}(x_i | \theta) - y_i)^2}{2e_i^2}\right] + \frac{1-g_i}{\sqrt{2\pi \sigma_b^2}} \exp\left[\frac{-(\hat{y}(x_i | \theta) - y_i)^2}{2\sigma_b^2}\right]$$

- Now we allow the model to have a nuisance parameter  $0 < g_i < 1$  for each data point:  $g_i = 0$  indicates an outlier. We can also allow  $\sigma_b$  to be a nuisance parameter to marginalize over (or just make it a large number)
- We can define an outlier (circle) whenever posterior  $E(g_i) < 0.5$
- prior on  $g_i$ : we can adopt a noninformative (uniform) prior, or we could have adopted a double peaked prior (one peaked at 0 one at 1) to force the solutions into 0 or 1: this does buy us that much when compared to simply using  $E(g_i) < 0.5$  criterion.

# Result of Gaussian 2 Mixture Model



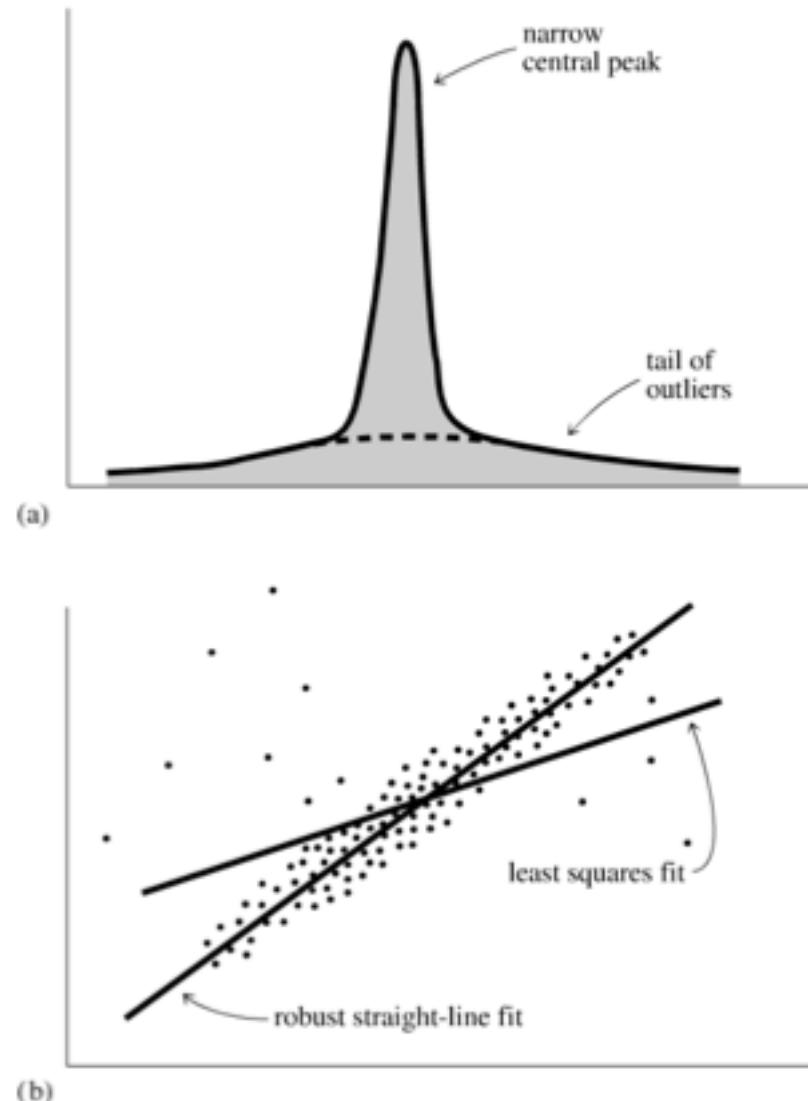
Note that this may not be what we want: outliers may be a source of information, so labeling them and discarding may destroy useful information

# Pooling

- In previous example we have assumed each event has its own  $g_i$  without any connection between them. In the context of drawing data from separate experiments (which is not the case in this example) this is called no pooling.
- We could have also used  $g_i = g$ , which is to say that all data are drawn from the same pdf: complete pooling in the context of separate experiments
- Or we could have grouped data into separate groups each with its own prior, if we have a priori reasons to separate them into such groups: partial pooling

# Alternatives: Robust Analysis

- So far we used L2 norm, justified by Gaussian error distribution, as in least squares fit. We used a mixture of Gaussians to treat outliers
- If we know the error probability distribution we can use it instead: Gaussian is the most compact and any other distribution will reduce sensitivity to outliers
- This is equivalent to changing the norm



# Error PDF

- Suppose we know PDF of the error P

$$P = \prod_{i=0}^{N-1} \{\exp[-\rho(y_i, y\{x_i|\mathbf{a}\})] \Delta y\}$$

- We then want to minimize  $\sum_{i=0}^{N-1} \rho(y_i, y\{x_i|\mathbf{a}\})$
- If this is only a function of difference between model and data we can minimize over  $\mathbf{a}$

$$\begin{aligned} & \sum_{i=0}^{N-1} \rho\left(\frac{y_i - y(x_i|\mathbf{a})}{\sigma_i}\right) \quad \psi(z) \equiv \frac{d\rho(z)}{dz} \\ & 0 = \sum_{i=0}^{N-1} \frac{1}{\sigma_i} \psi\left(\frac{y_i - y(x_i)}{\sigma_i}\right) \left(\frac{\partial y(x_i|\mathbf{a})}{\partial a_k}\right) \quad k = 0, \dots, M-1 \end{aligned}$$

# M-Estimators and Norms

- Gaussian (L2)  $\rho(z) = \frac{1}{2}z^2$   $\psi(z) = z$
- Laplace (double exponential, L1)  $\rho(x) = |z|$   $\psi(z) = \text{sgn}(z)$
- Lorentzian (Cauchy)  $\rho(z) = \log\left(1 + \frac{1}{2}z^2\right)$   $\psi(z) = \frac{z}{1 + \frac{1}{2}z^2}$
- All are special cases of Student t:  $\rho(z) = \log(n+z^2)$
- Student t can also be viewed as a mixture of gaussians with the same mean and variances distributed as inverse- $\chi^2$  with n degrees of freedom
- Norms: L<sub>p</sub> norm defined as  $\|Z\|_p = \left(\sum_{i=1}^N |Z_i|^p\right)^{1/p}$
- L2: ridge, L1: lasso

# Regularization

- In image processing, machine learning etc. we often work with many more parameters than we can determine from the data: this is a form of non-parametric analysis (i.e. we have many more parameters than we can handle)
- Because of this the parameters will fit noise: overfitting
- If there is no noise by sampling is sparse the parameters will fit the data where measured and the model will make little sense elsewhere: overfitting
- To prevent that we regularize the solutions by imposing some smoothness
- Easiest way to achieve this is to minimize the sum of  $\chi^2$  and norm of parameters, with the relative contribution determining the overall level of smoothness

# Tikhonov (ridge, L2) Regularization

- We use L2 norm and add it to linear least squares

$$\|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

- $\Gamma$  can be a general matrix, but for L2  $\Gamma = \alpha I$
- Normal equation solution  $\hat{\mathbf{x}} = (A^\top A + \Gamma^\top \Gamma)^{-1} A^\top \mathbf{b}$
- SVD solution:  $A = U\Sigma V^\top$      $\hat{\mathbf{x}} = V D U^\top b$      $D_{ii} = \frac{\sigma_i}{\sigma_i^2 + \alpha^2}$
- We see that regularization reduces condition number of the matrix: it regularizes it

# Bayesian Regression

$$t = \underbrace{y(\mathbf{x}, \mathbf{w})}_{\text{deterministic}} + \underbrace{\epsilon}_{\text{Gaussian noise}}$$

- In the Bayesian context we perform regression of coefficients  $a_j$  assigning them some prior distribution, such as a gaussian with some precision  $\alpha$ . If we also have noise precision  $\beta$  then  $\ln p$  is

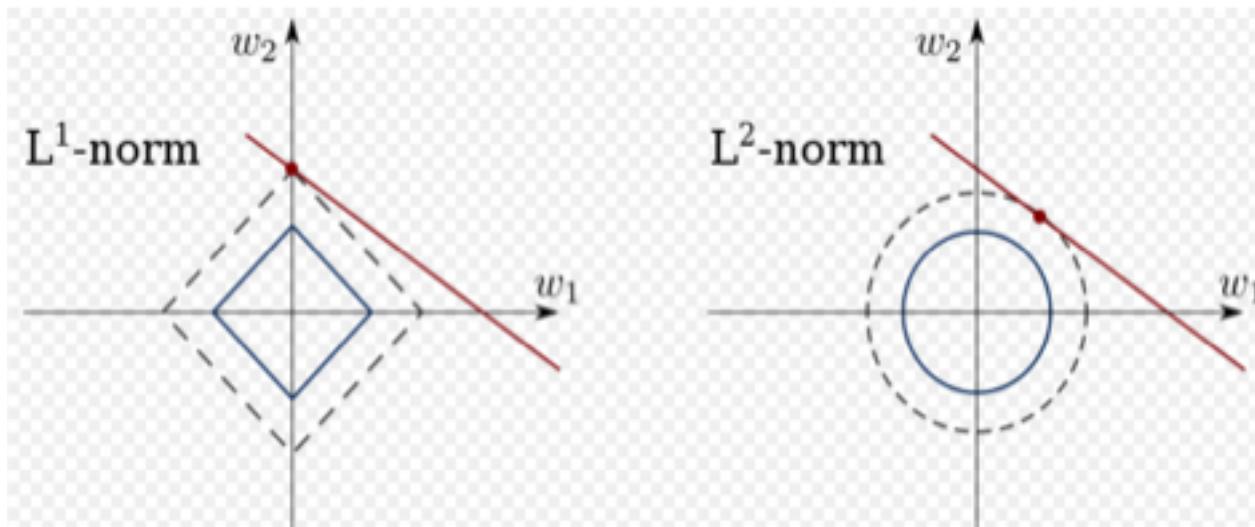
$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2} \sum_{n=1}^N (t_n - \mathbf{w}^T \phi(\mathbf{x}_n))^2 - \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + \text{const}$$

- So regularizing parameter is  $\lambda = \alpha/\beta$
- How do we set  $\alpha$ ? It is a hyper-parameter that we can determine separately from the data itself. We will return to this when we discuss gaussian process and regression (lecture 11)

35

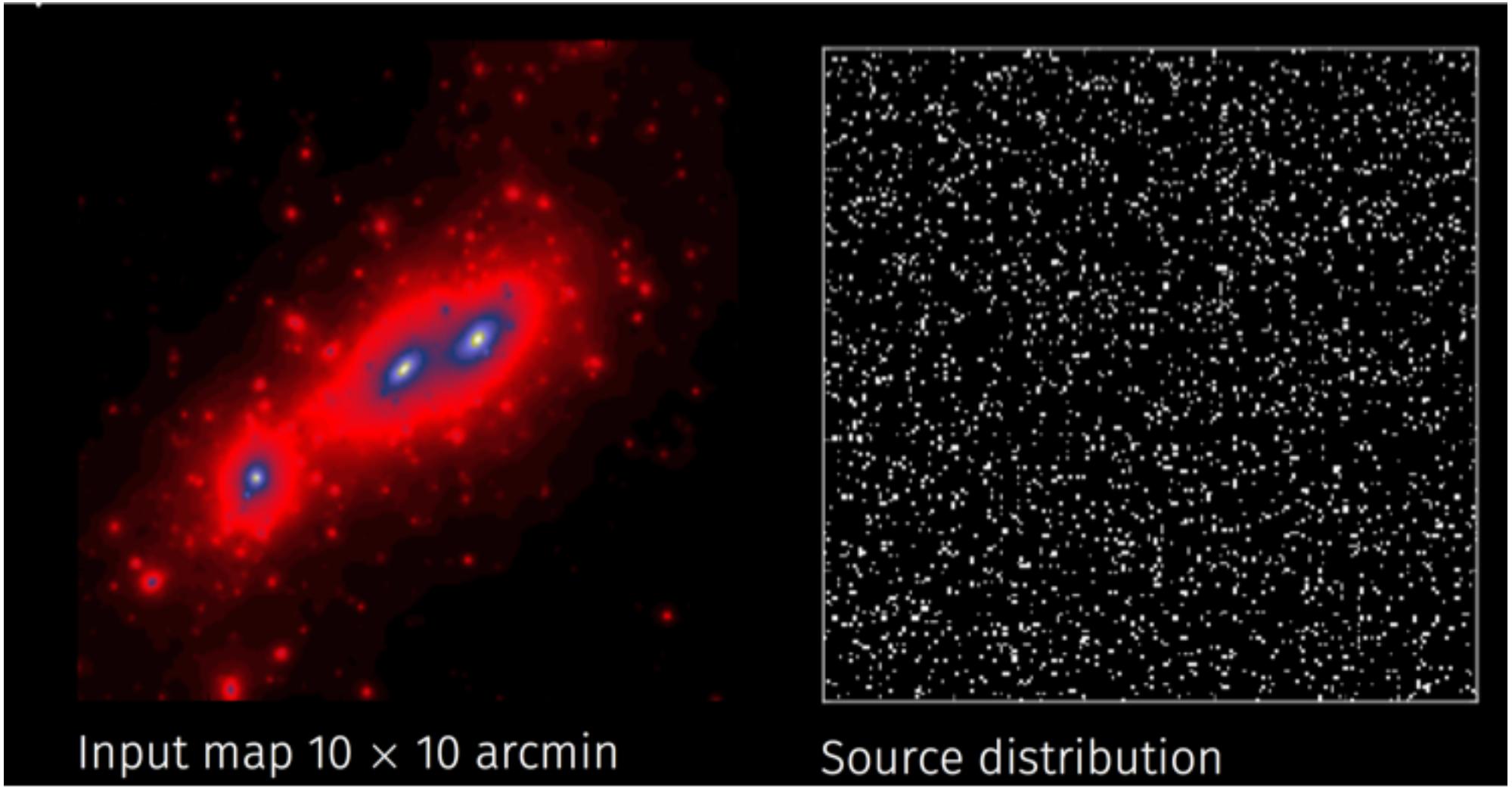
# L1 vs. L2 Norm for Regularization

- Suppose we have just one linear relation and 2 parameters: we must regularize. We want to find  $w_1$  and  $w_2$  subject to their linear relation  $E_{11}w_1 + E_{12}w_2 = c_1$  (normal eq., red line) and minimizing the norm L1 or L2



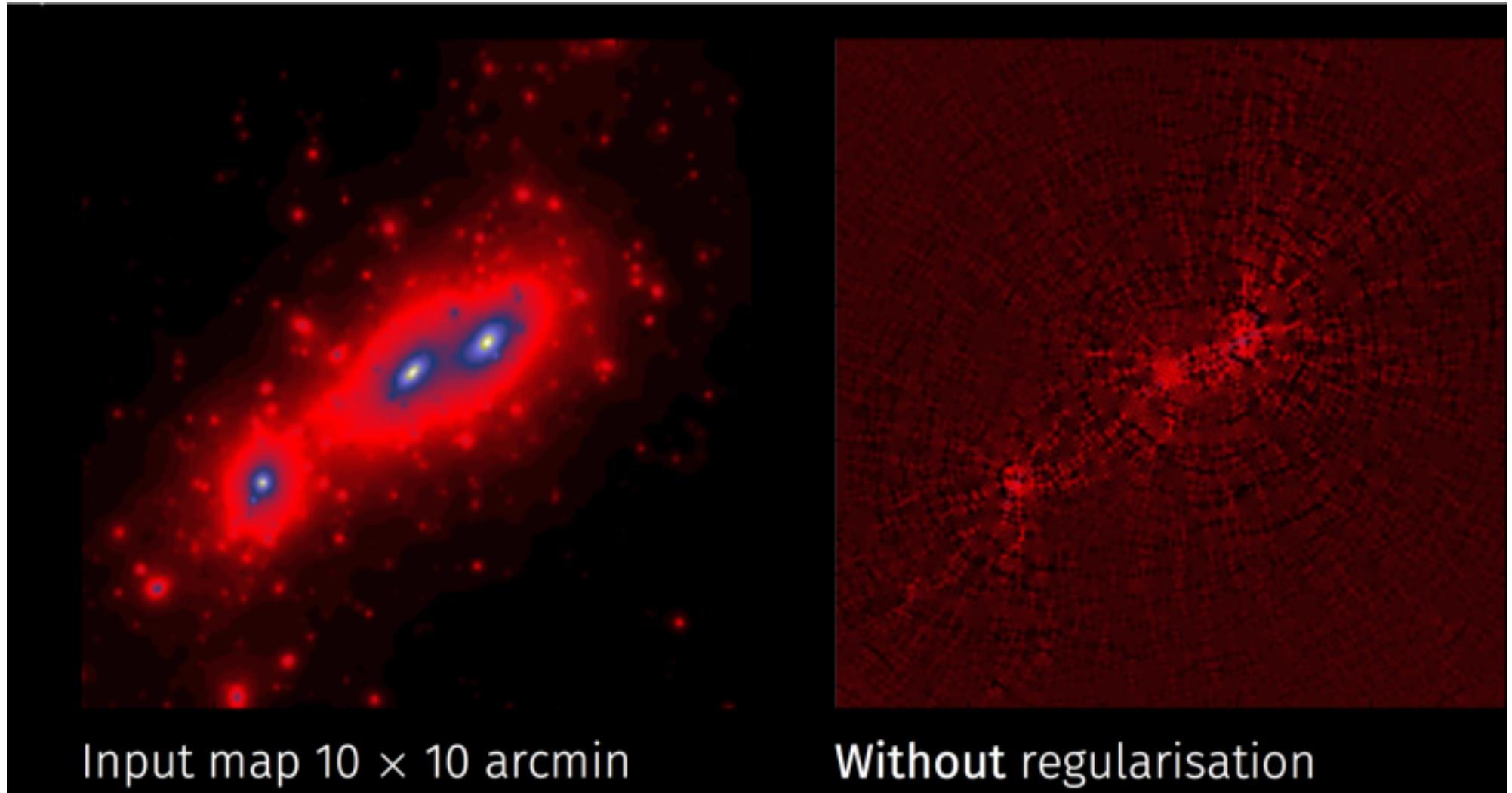
- We see that L1 norm is minimized at  $w_1 = 0$ : L1 norm enforces sparseness, L2 does not
- Bayesian view: Laplace distribution is sharply peaked at 0
- L1 regularization is called LASSO: can both regularize and reduce dimensionality (shrinkage). Least absolute shrinkage and selection operator

## Example: Image sampled at discrete points

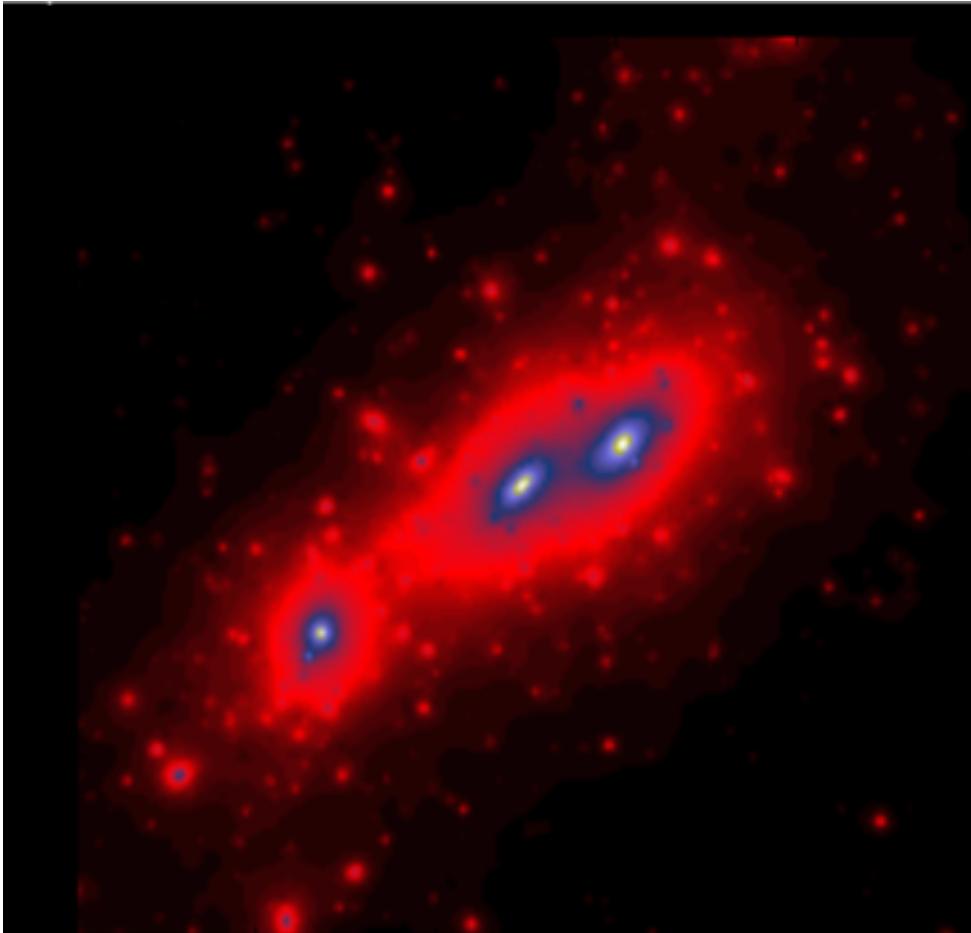


Credit: F. Lanusse **37**

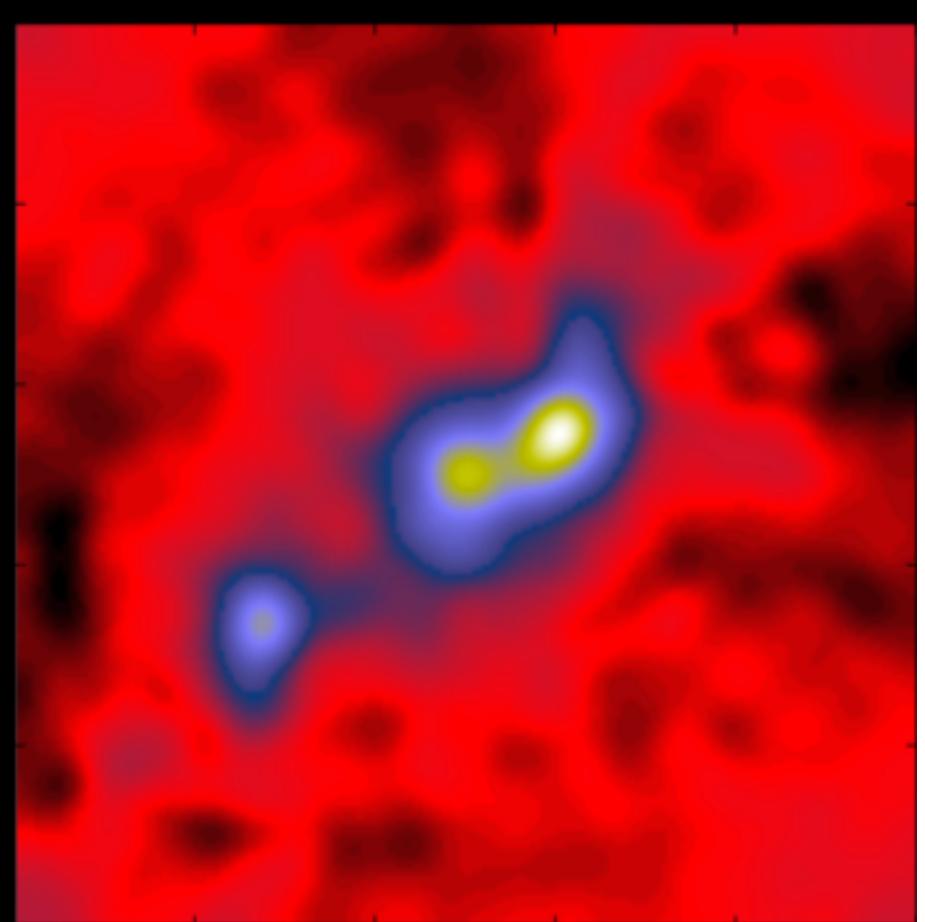
# No Regularization Reconstruction



## L2 Norm Regularization

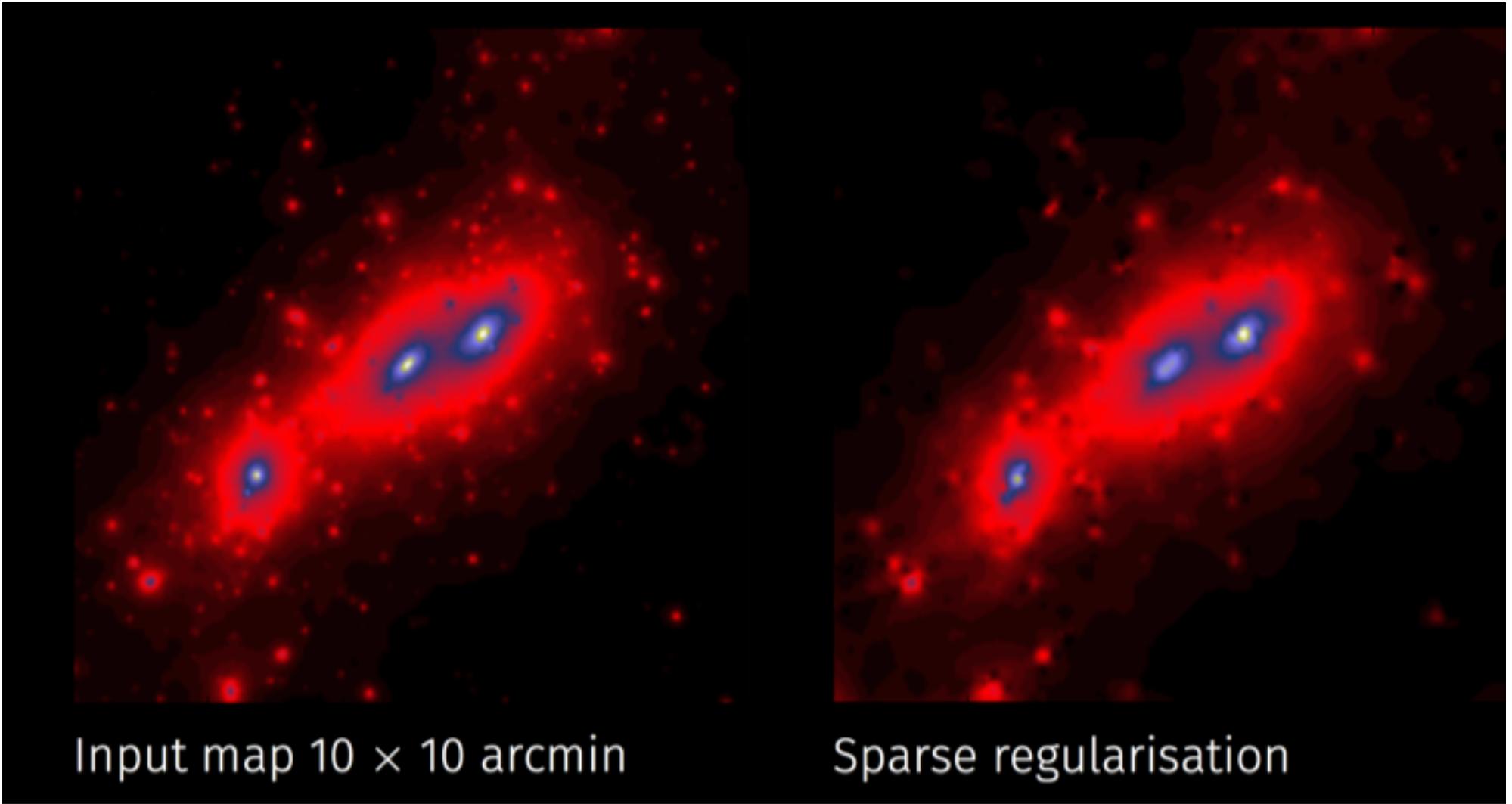


Input map  $10 \times 10$  arcmin



Binning + Gaussian Smoothing

# L1 Norm Regularization



# Linmix: Fitting with correlated errors in x and y

Perform linear regression of  $y$  on  $x$  when there are measurement errors in both variables. The regression assumes:

$$\eta = \alpha + \beta * x + \epsilon$$

$$x = x_i + x_{\text{err}}$$

$$y = \eta + y_{\text{err}}$$

Here,  $(\alpha, \beta)$  are the regression coefficients,  $\epsilon$  is the intrinsic random scatter about the regression,  $x_{\text{err}}$  is the measurement error in  $x$ , and  $y_{\text{err}}$  is the measurement error in  $y$ .  $\epsilon$  is assumed to be normally-distributed with mean zero and variance  $\sigma^2$ .  $x_{\text{err}}$  and  $y_{\text{err}}$  are assumed to be normally-distributed with means equal to zero, variances  $x_{\text{sig}}^2$  and  $y_{\text{sig}}^2$ , respectively, and covariance  $x_{\text{cov}}$ . The distribution of  $x_i$  is modeled as a mixture of normals, with group proportions  $\pi_i$ , means  $\mu_i$ , and variances  $\tau^2$ . The following graphical model illustrates, well..., the model...

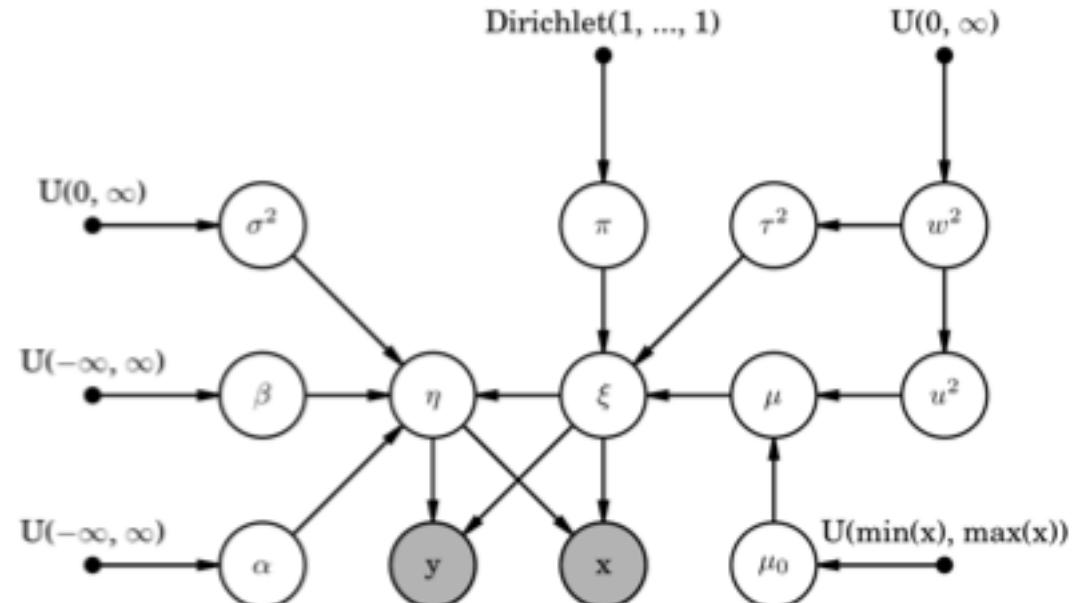
$U(x,y)$ : uniform between  $x$  and  $y$

Dirichlet distribution  $f$ :

$$f(x_1, \dots, x_K; \alpha_1, \dots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i-1}$$

$$\sum_{i=1}^K x_i = 1 \text{ and } x_i \geq 0 \text{ for all } i \in [1, K]$$

This could have been solved by integrating out latent variables analytically: this does not change hierarchical modeling approach



# Summary

- The simplest way to write the full probabilistic model is to break it down into individual conditional probabilities, which often includes several levels of hierarchy of parameters
- Doing this is facilitated with the help of directed acyclic graphs (Bayesian networks) or undirected graphs (Markov networks)
- The price one pays is a large number of parameters: one either works with all of them or tries to marginalize analytically over nuisance parameters that are not of interest
- A few typical examples are regression with errors in both variables, regression with outliers etc.
- A more heuristic approach to outliers is robust analysis with M-estimators where the error distribution is generalized beyond gaussian to a Student t distribution
- This is related to the concept of L-norms, where L1 lasso norm corresponds to Laplace distribution which enforces sparsity
- This in turn is related to regularization in the context of image processing with incomplete and noisy data

43

# Literature

- *Numerical Recipes*, Press et al., Chapter 15
- *Bayesian Data Analysis*, Gelman et al. , Chapter 5