

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 θ 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 10: ADVANCED BAYESIAN CONCEPTS

- We can introduce x as latent variables and model them together with θ . Then θ 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 θ .

Graphical Models for Probabilistic and Casual Reasoning

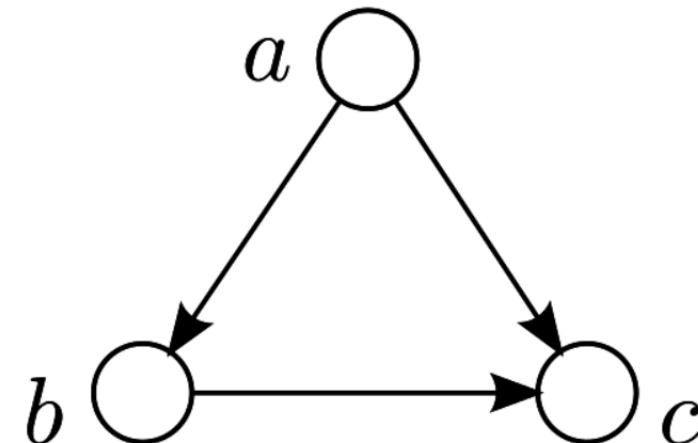
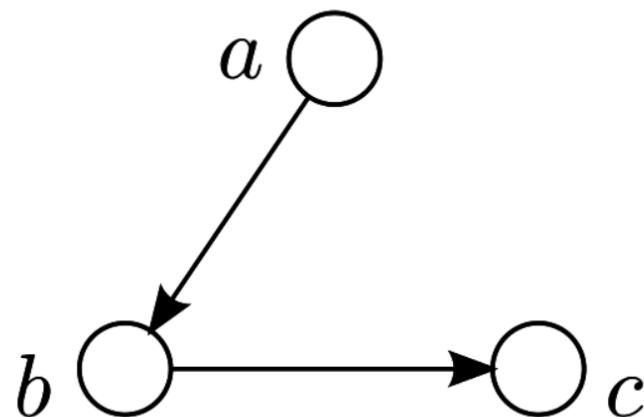
- We would like to describe the causal flow of events such that we can generate (simulate) events in a probabilistic setting (a flowchart of generating data)
- We can describe this with **directed acyclic graphs (DAG)**
- Typically we divide the process into components each of which generates a single 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

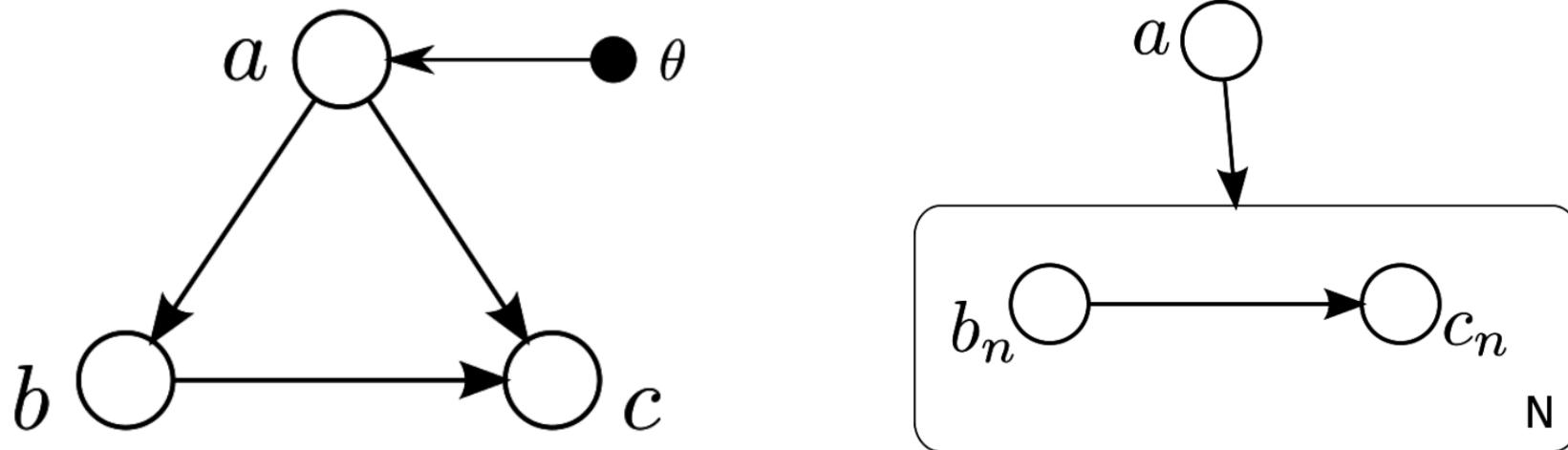


$$p(a, b, c) = p(c|b)p(b|a)p(a)$$

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

PGM Rules

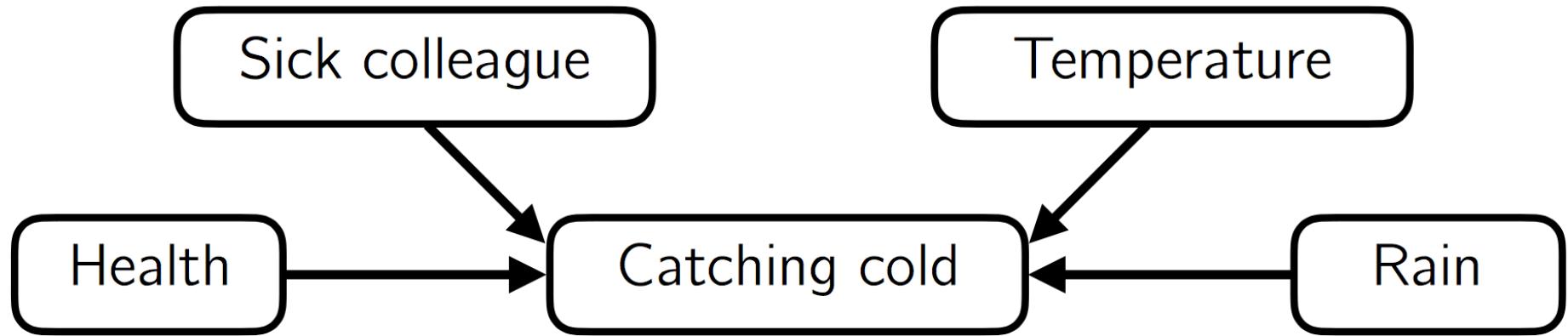
- Each solid point is a fixed variable (pdf is a delta function)
- Each plate contains conditionally independent variables: repetition, compressed notation for many nodes



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

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

Breaking causality down into components

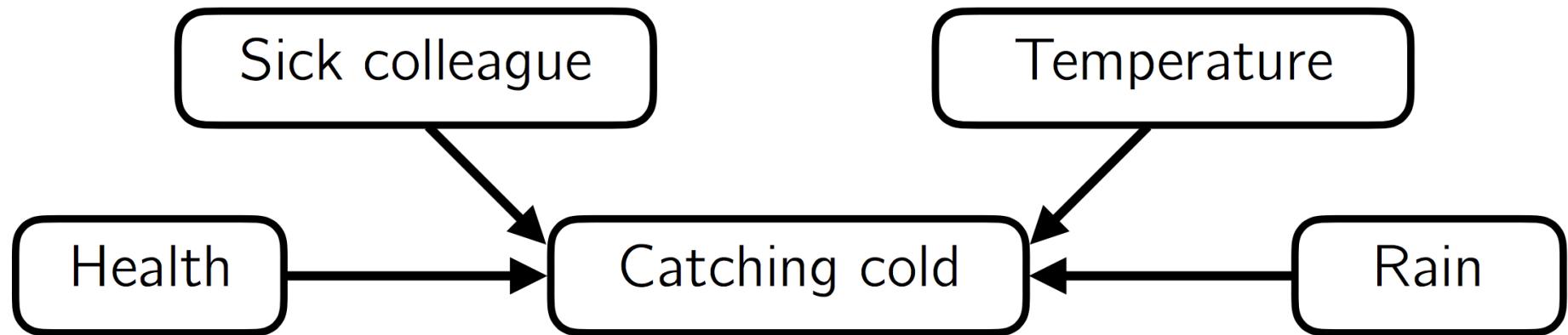


Independent causes of Catching cold

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

Credit: Slides from B. Leistedt

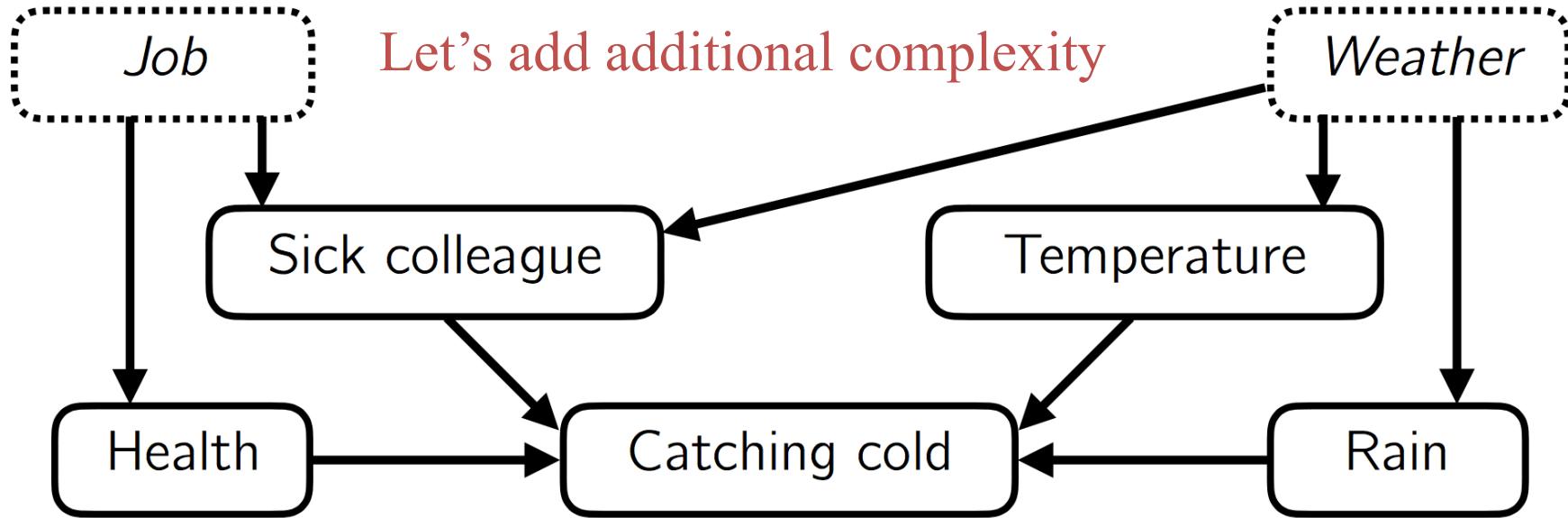
Breaking causality down into components



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

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

Credit: Slides from B. Leistedt



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

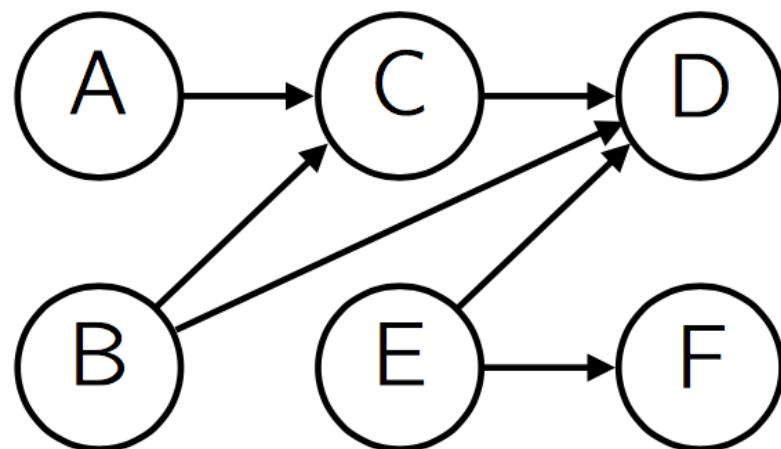
$$p(C | S, H, T, R, J, W) = p(C | S, H, T, R)$$

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

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

Example

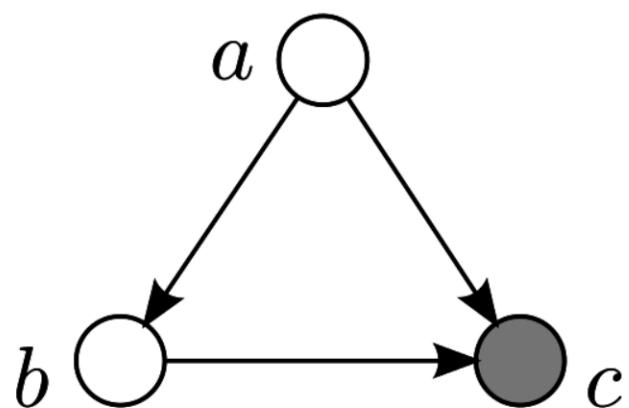
- Write down corresponding probability expressions for this graph and discuss the meaning with your neighbor



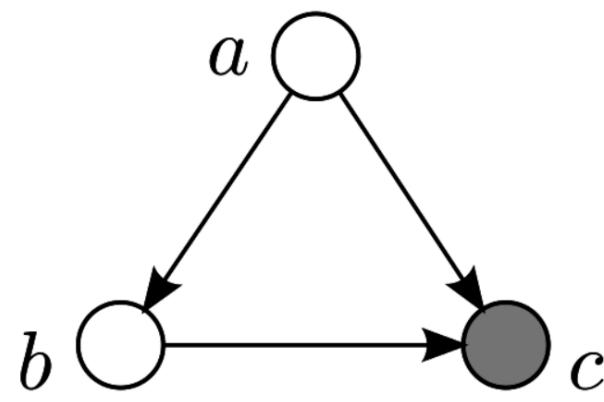
$$\begin{aligned} p(A, B, C, D, E, F) = \\ \times p(F|E) p(D|B, E, C) \\ \times p(E) p(C|A, B) p(A) p(B) \end{aligned}$$

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



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

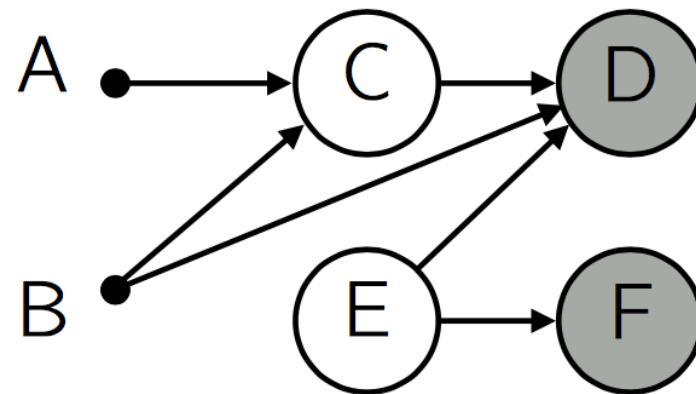


$$\underbrace{p(a, b|c)}_{\text{posterior}} \propto \underbrace{p(c|a, b)}_{\text{likelihood}} \underbrace{p(b|a)p(a)}_{\text{prior}}$$

11

Posterior Inference

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

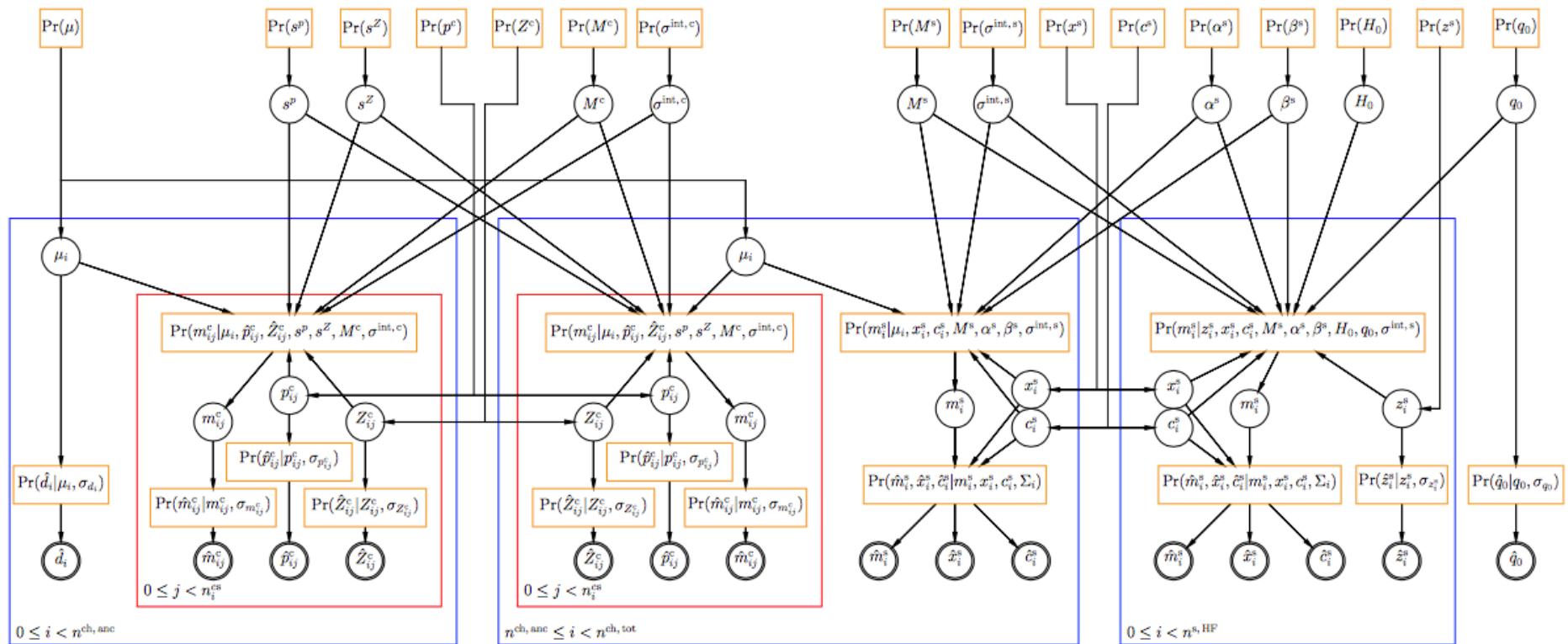


$$\begin{aligned} P(\text{Parameters} | \text{Data, Model}) &= p(C, E | D, F, A, B) \\ &= p(F|E) p(D|C, E) p(C | A, B) p(E) p(A) p(B) \end{aligned}$$

- 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)

A Big PGM (Distance Ladder)

- This is a way to organize the generative probabilistic model



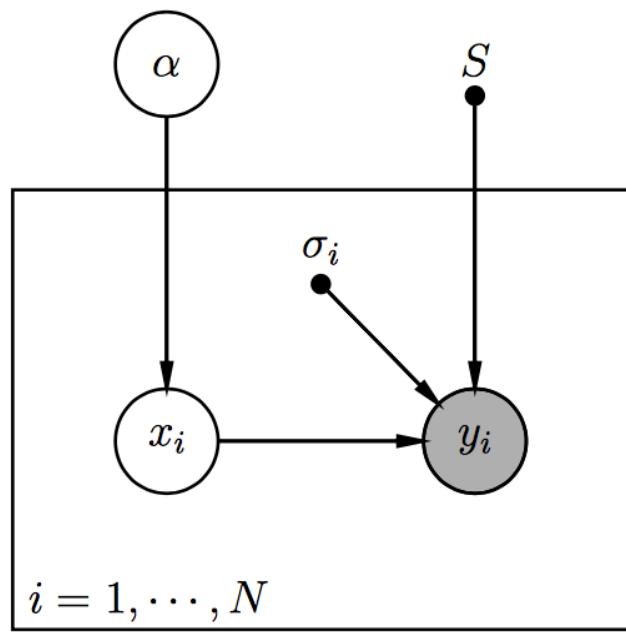
Hierarchical Bayesian Models

- 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 θ .
- d : **observable**
- Variables that are not observed are called **latent variables**: θ, x
- Variables we do not care about are called *nuisance variables*: x . We want marginalize over them to determine θ

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, ..., 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

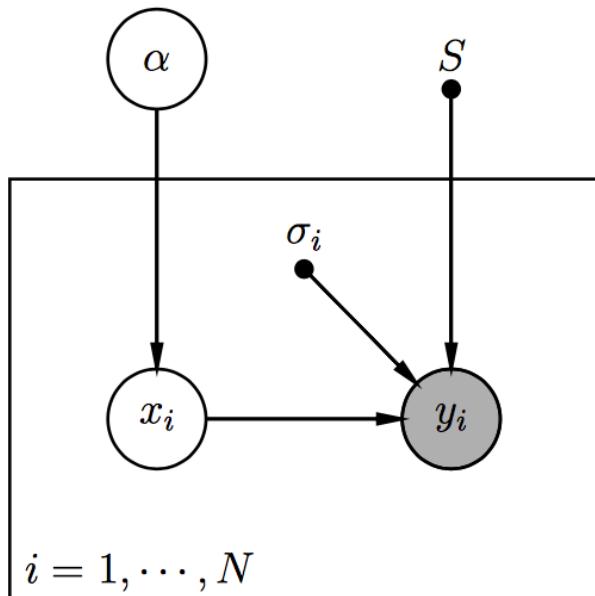


- ▶ y_i 's : measurements of the temperature in a room
- ▶ σ_i : Gaussian noise
- ▶ x_i 's : true temperature
- ▶ α : parameter parametrizing the true distribution of temperatures.
- ▶ S : some selection effect (e.g., no measurement if temperature is < 0 degrees)

Marginalization over Latent Variables

We are interested in inferring α

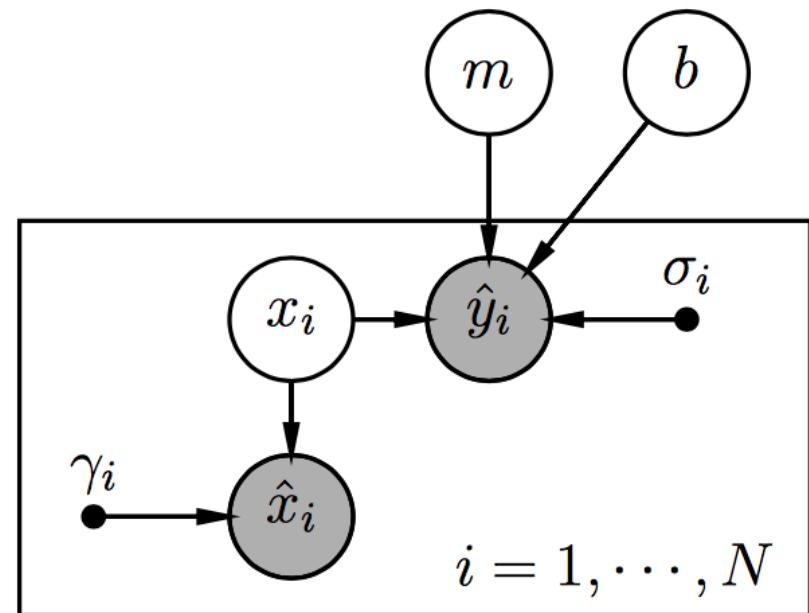
We need to **marginalize over** the latent x_i 's, numerically or analytically



$$\begin{aligned} p(\alpha | \{y_i, \sigma_i\}) \\ \propto \prod_i p(y_i | \alpha, \sigma_i) p(\alpha) \\ = \prod_i \int p(y_i, x_i | \alpha, \sigma_i) p(\alpha) dx_i \\ = \prod_i \int p(y_i | x_i, \alpha, \sigma_i) p(x_i | \alpha, \sigma_i) p(\alpha) dx_i \\ = \prod_i \int p(y_i | x_i, \sigma_i) p(x_i | \alpha) p(\alpha) dx_i \end{aligned}$$

Additional Complication: Noise in x

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

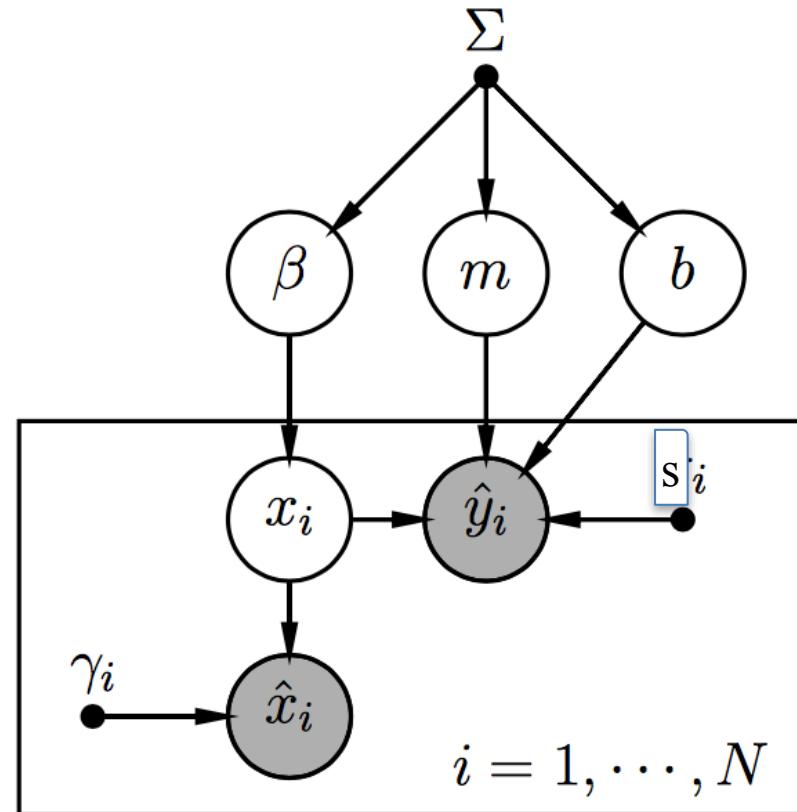


Sometimes marginalization can be done analytically

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

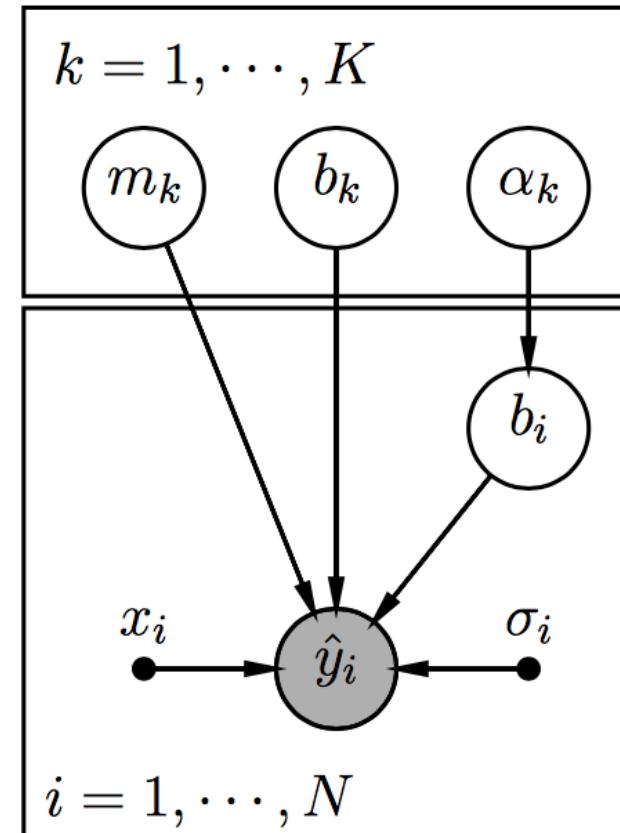
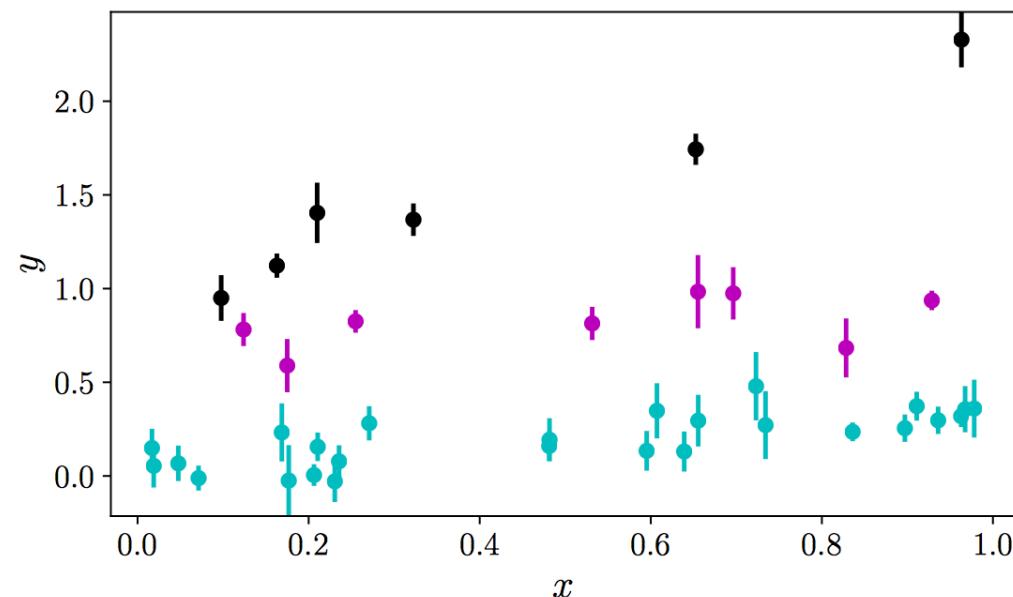
We should also put hyperpriors onto parameters

- We have hyperprior Σ
- A proper PGM should start with hyperpriors and end with observables



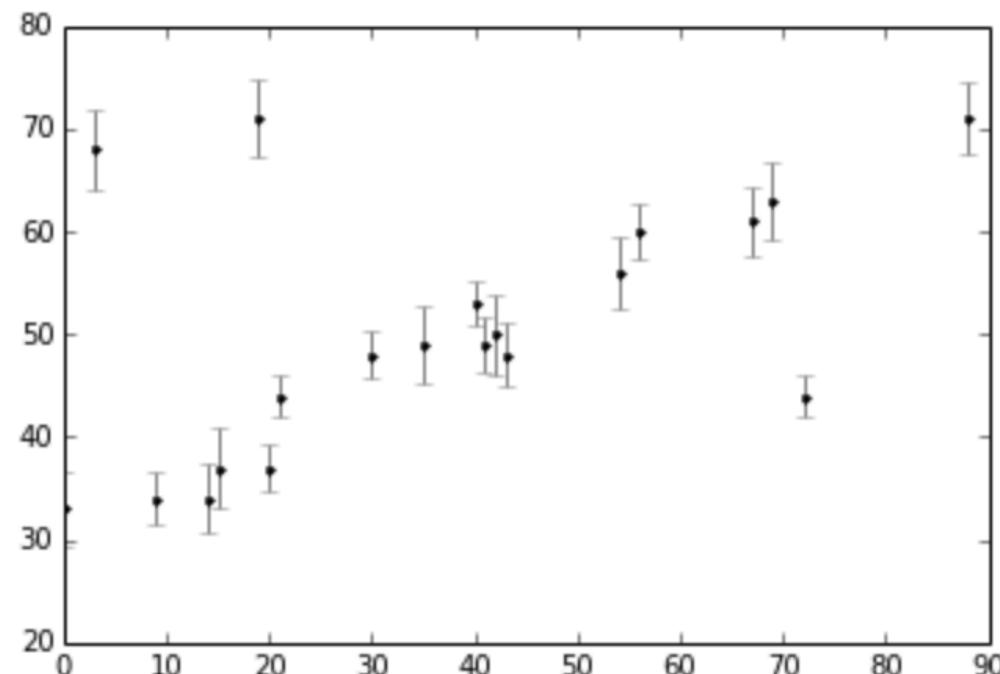
Another Extension: Mixture Models

- 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

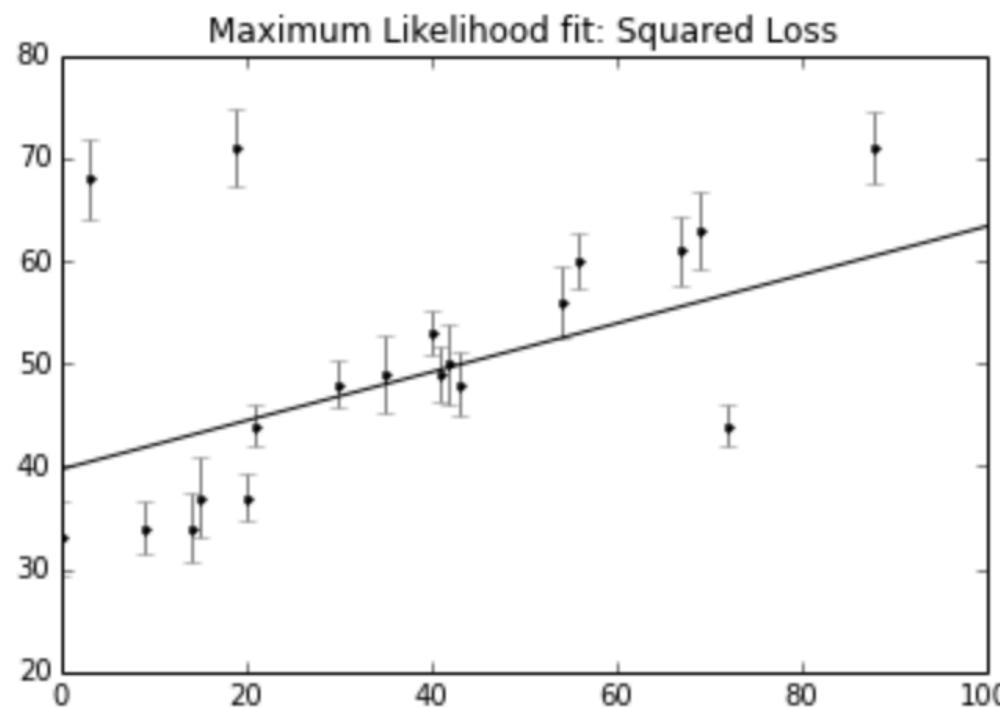
- Suppose we have data that can be fit to a linear regression, apart from a few outlier points
- It is always better to understand the underlying generative model of outliers
- But suppose we just want to identify them



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 (we will discuss more formally what that means in the next lecture)



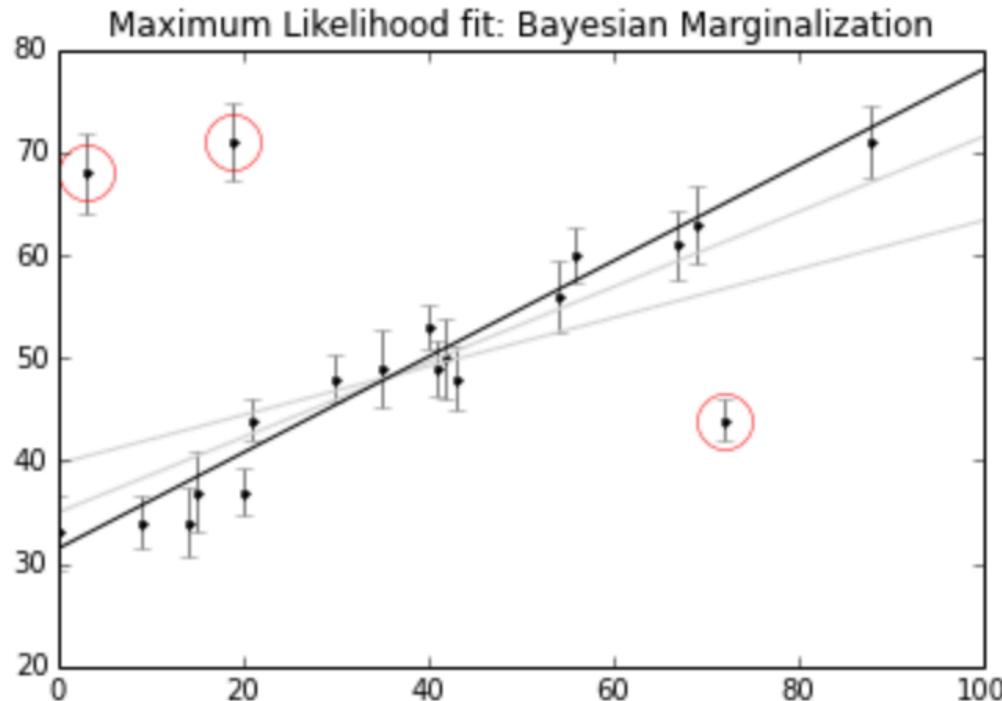
23

Let us model this as a Gaussian

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

- 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 σ_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



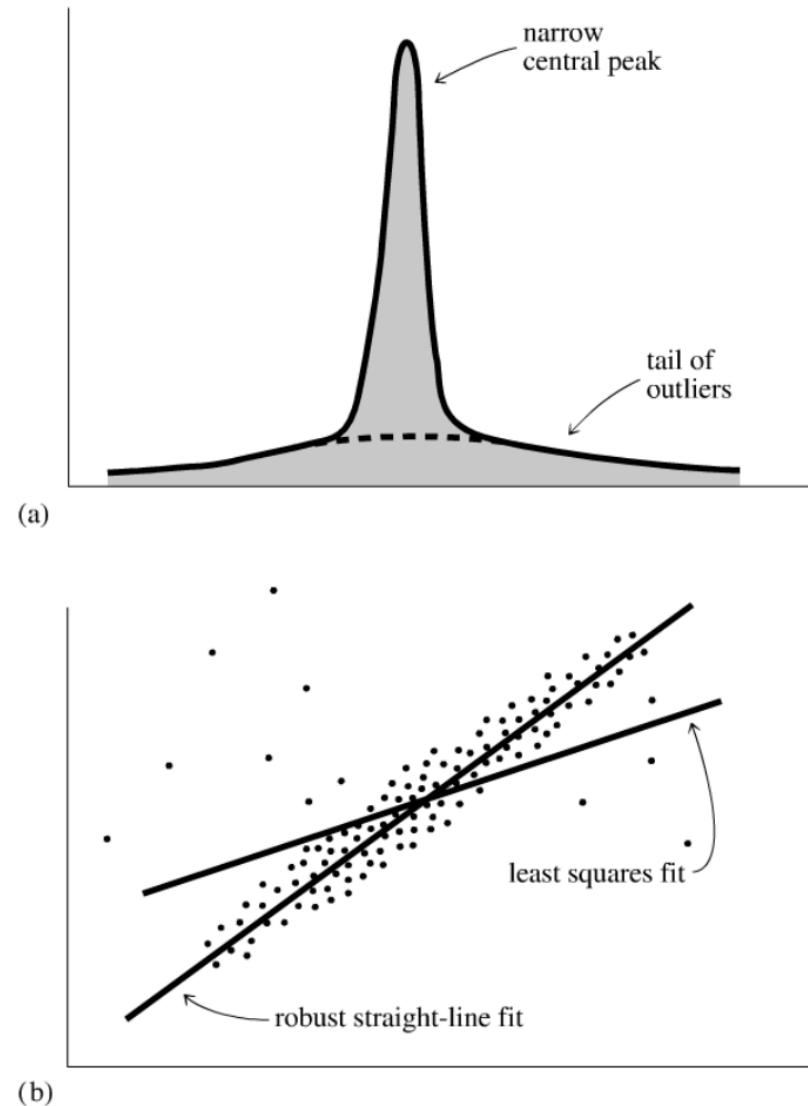
Note that this may not be what we want: outliers may be a source of information, so labeling them and discarding may destroy useful information

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) $\rho(x) = |x|$ $\psi(z) = \text{sgn}(z)$
- 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- χ^2 with n degrees of freedom
- Norms: L_p norm defined as
- L2: ridge, L1: lasso

$$\|Z\|_p = \left(\sum_{i=1}^N |Z_i|^p \right)^{1/p}$$

29

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 χ^2 and norm of parameters, with the relative contribution determining the overall level of smoothness
- We will work this out in the context of Wiener filter when we discuss Fourier methods.
- Here we want to compare L1 and L2 norms

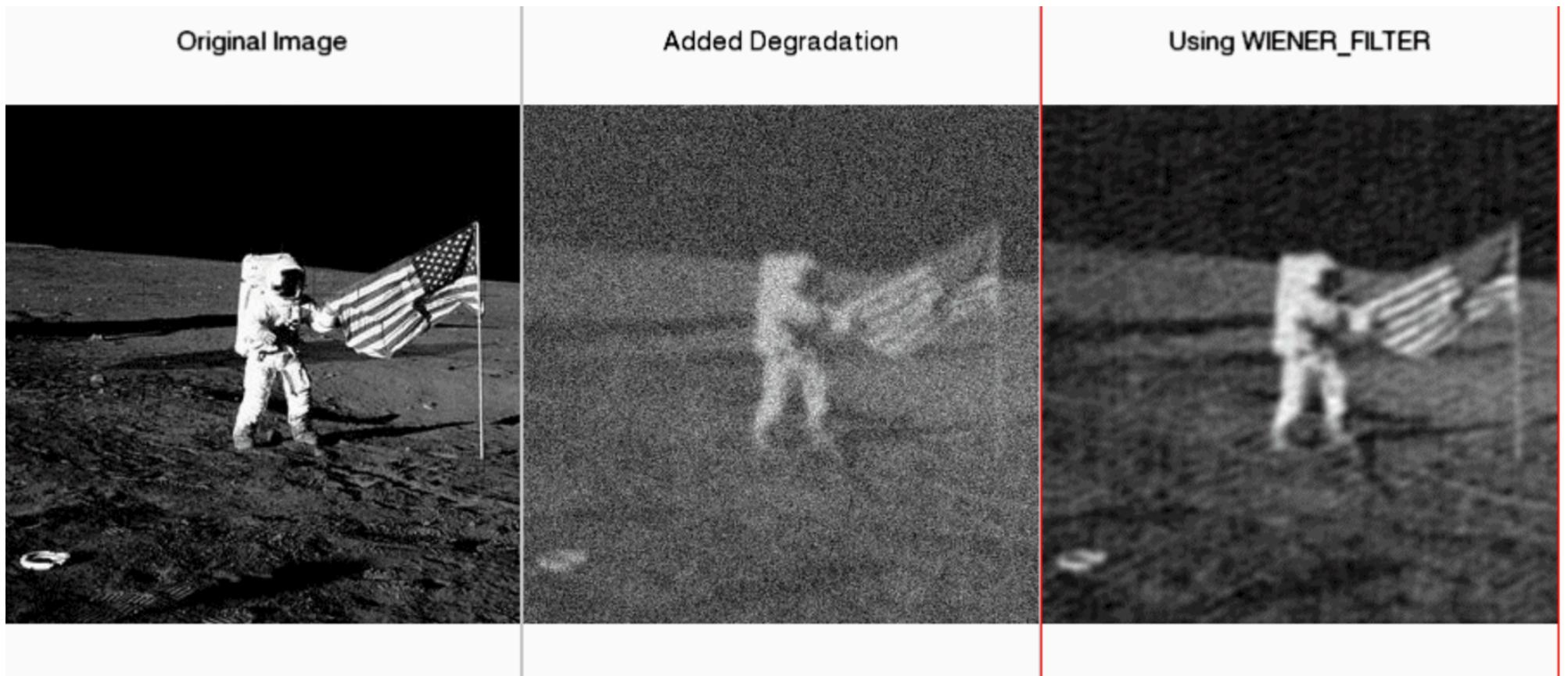
Tikhonov (ridge, L2) Regularization

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

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

- Γ 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: $A = U\Sigma V^\top$ $\hat{\mathbf{x}} = V D U^\top \mathbf{b}$ $D_{ii} = \frac{\sigma_i}{\sigma_i^2 + \alpha^2}$
- We see that regularization reduces condition number of the matrix: it regularizes it

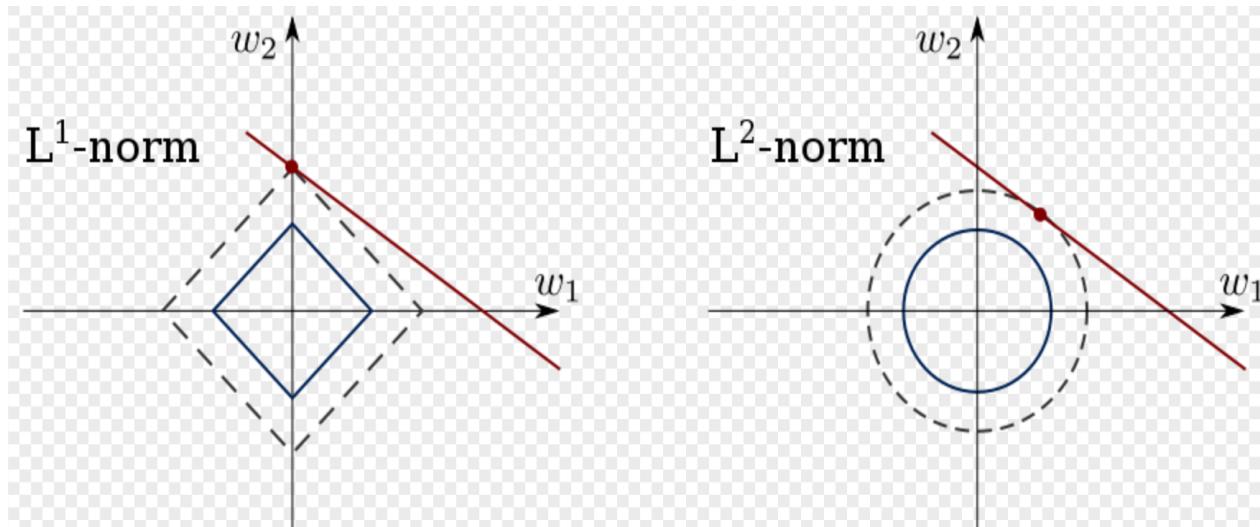
Wiener Filtering (Fourier L2)



32

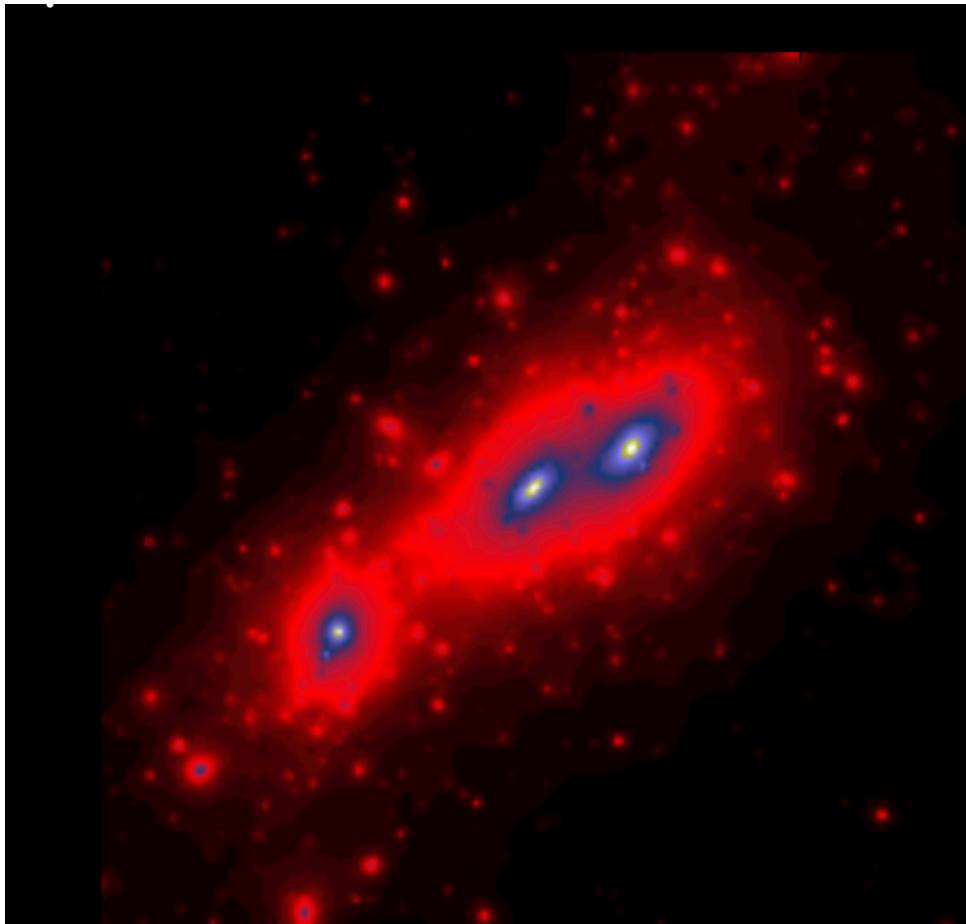
L1 vs. L2 Norm for Regularization

- 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

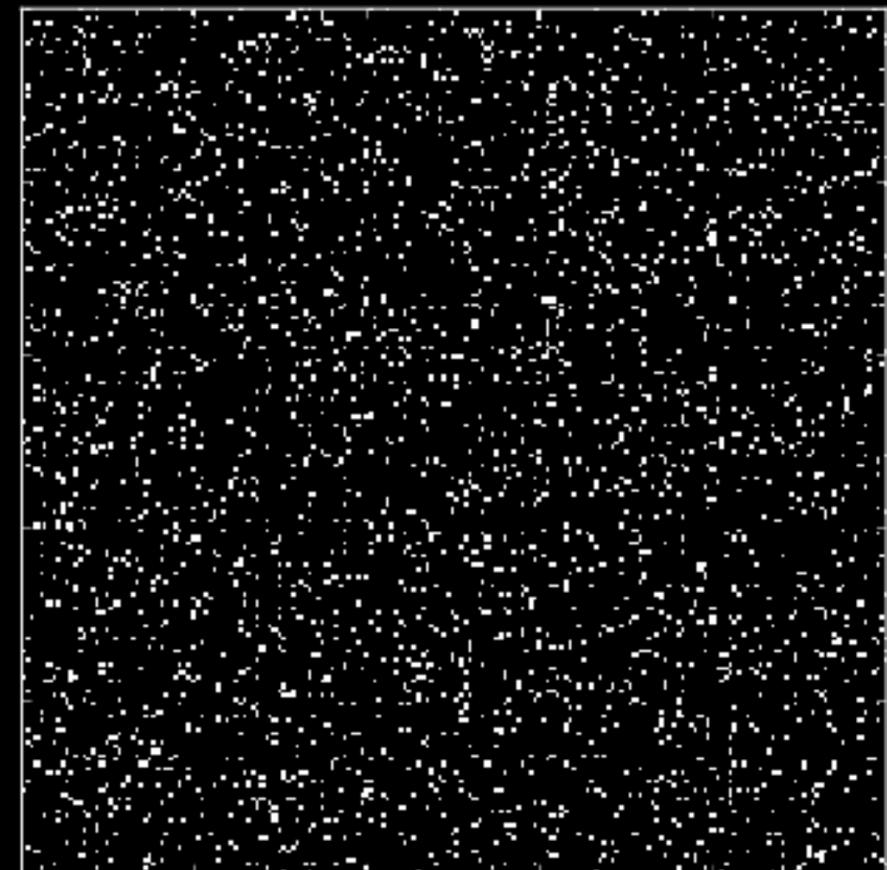


- We see that L1 norm is minimized at $w_1 = 0$: L1 norm enforces sparseness, L2 does not
- Bayesian view: Laplace distribution is sharply peaked at 0
- LASSO: can both regularize and reduce dimensionality (shrinkage)

Example: Image sampled at discrete points



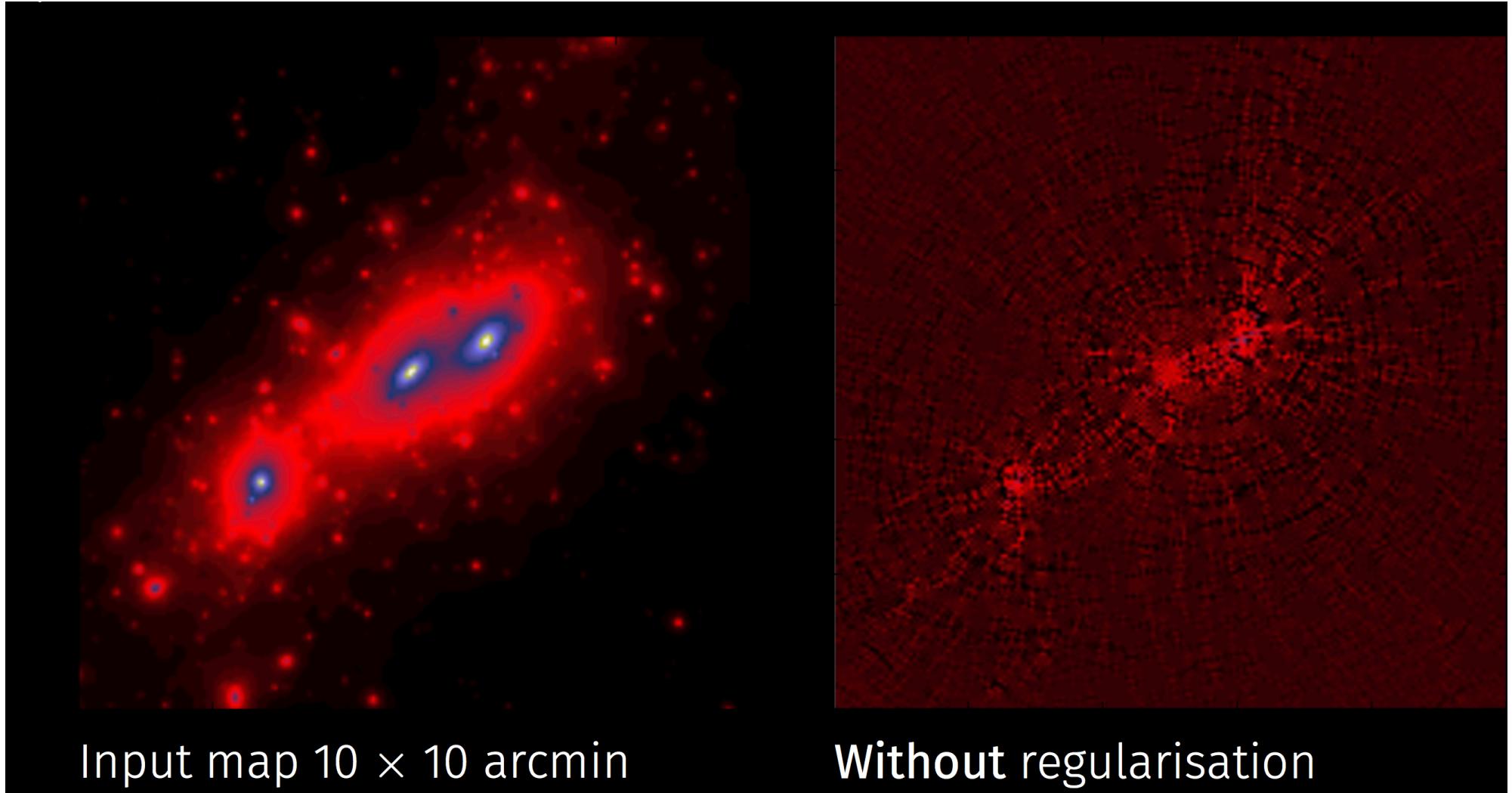
Input map 10×10 arcmin



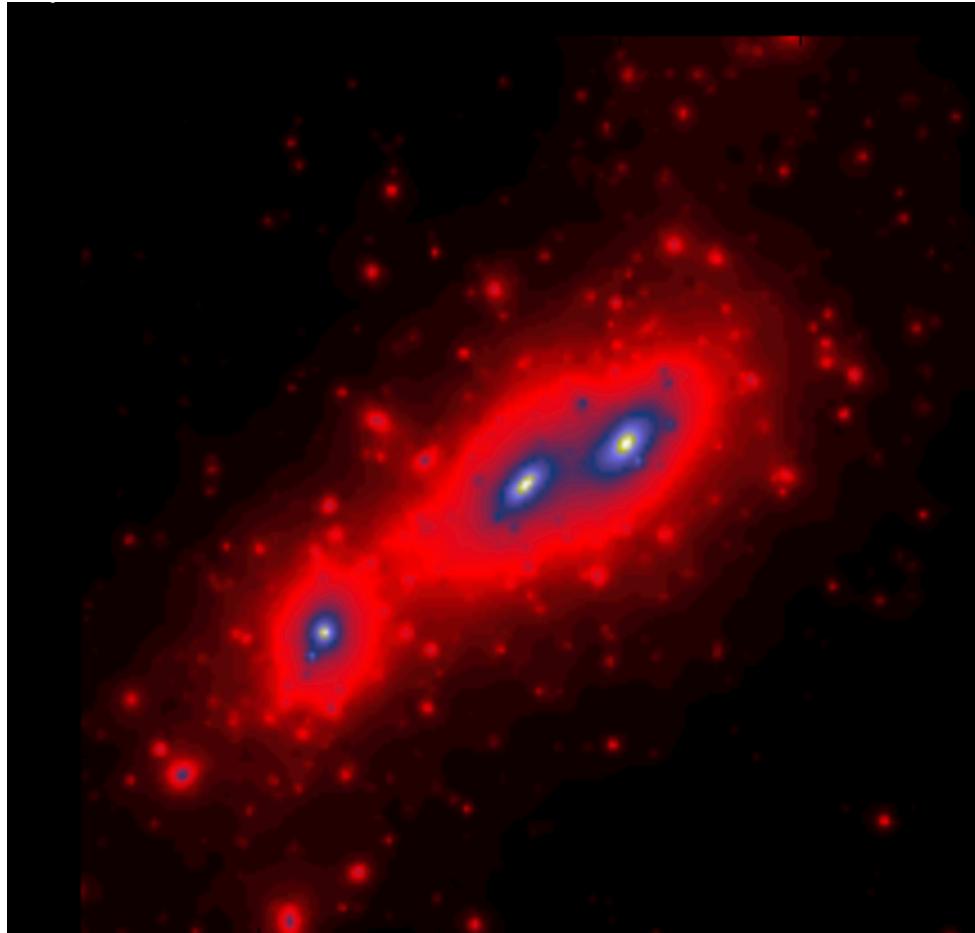
Source distribution

Credit: F. Lanusse **34**

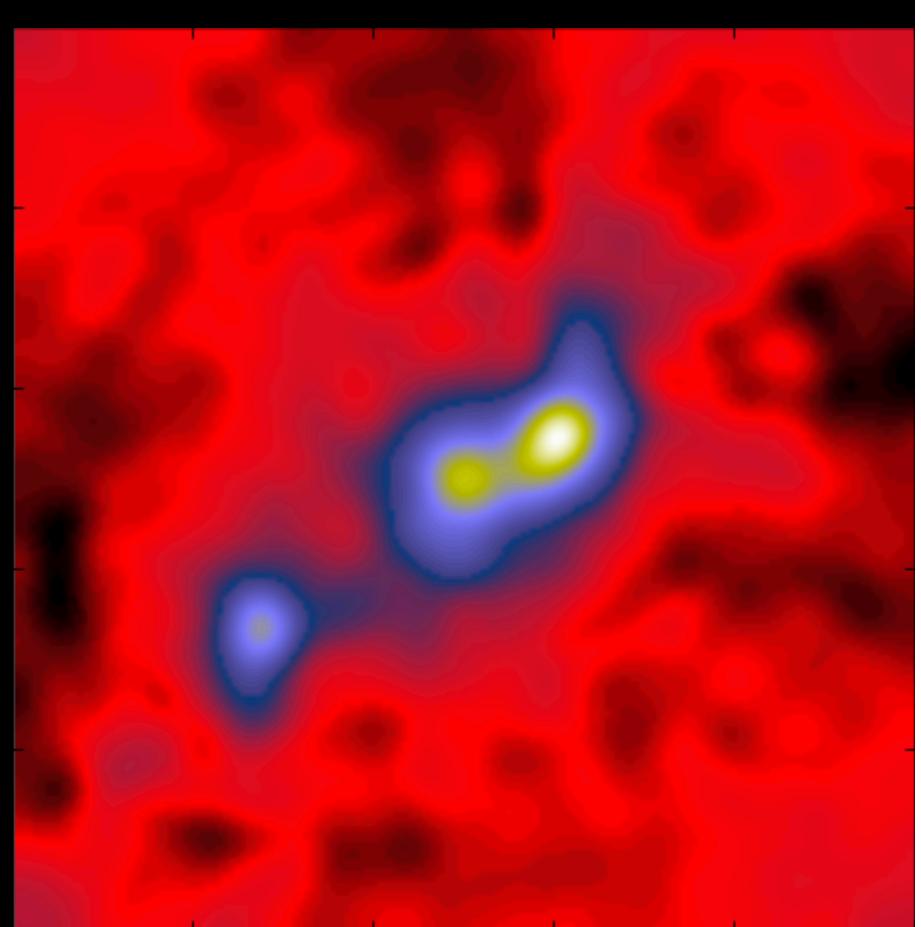
No Regularization Reconstruction



L2 Norm Regularization

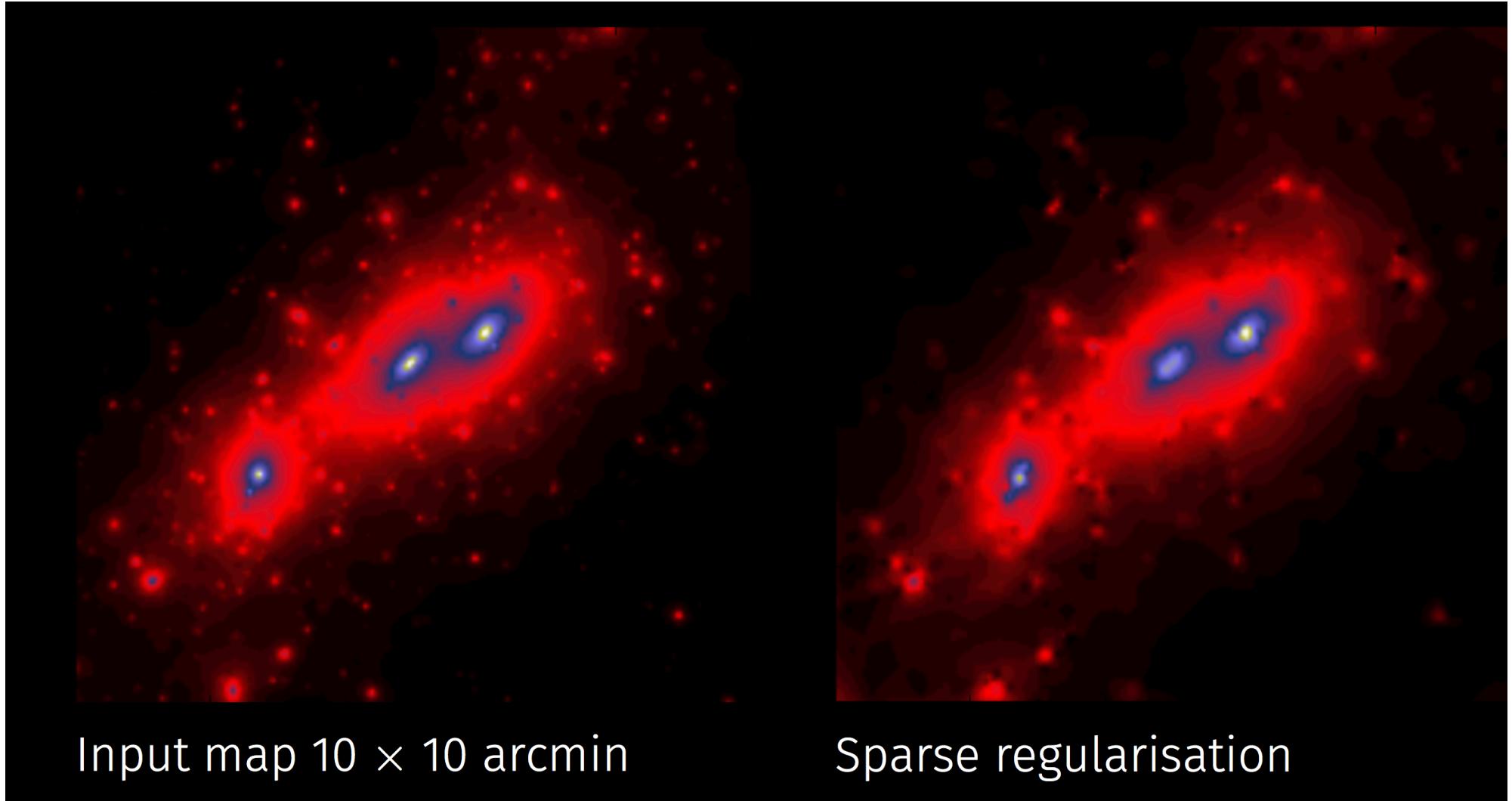


Input map 10×10 arcmin



Binning + Gaussian Smoothing

L1 Norm Regularization



Posterior for Mixture Models

$$\begin{aligned} & p(\{\alpha_k, m_k, b_k\} | \{\hat{y}_i, \sigma_i, x_i\}) \\ & \propto \prod_i p(\hat{y}_i | \{\alpha_k, m_k, b_k\}, x_i, \sigma_i) p(\{\alpha_k, m_k, b_k\}) \\ & = \prod_i \sum_{a_i} p(\hat{y}_i, a_i | \{\alpha_k, m_k, b_k\}, x_i, \sigma_i) p(\{\alpha_k, m_k, b_k\}) \\ & = \prod_i \sum_{a_i} p(\hat{y}_i | \{\alpha_k, m_k, b_k\}, a_i, x_i, \sigma_i) p(a_i | \{\alpha_k, m_k, b_k\}) p(\{\alpha_k, m_k, b_k\}) \\ & = \prod_i \sum_{a_i} p(\hat{y}_i | m_{a_i}, b_{a_i}, x_i, \sigma_i) p(a_i | \{\alpha_k\}) p(\{\alpha_k, m_k, b_k\}) \\ & = \prod_i p(\{\alpha_k, m_k, b_k\}) \sum_{a_i} \alpha_k \mathcal{N}(\hat{y}_i - m_{a_i} x_i - b_{a_i}; \sigma_i^2) \end{aligned}$$

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:

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

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

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

Here, (α, β) are the regression coefficients, ϵ is the intrinsic random scatter about the regression, x_{err} is the measurement error in x , and y_{err} is the measurement error in y . ϵ is assumed to be normally-distributed with mean zero and variance σ^2 . x_{err} and y_{err} are assumed to be normally-distributed with means equal to zero, variances x_{sig}^2 and y_{sig}^2 , respectively, and covariance x_{cov} . The distribution of x_i is modeled as a mixture of normals, with group proportions π_i , means μ_i , and variances τ_i^2 . The following graphical model illustrates, well..., the model...

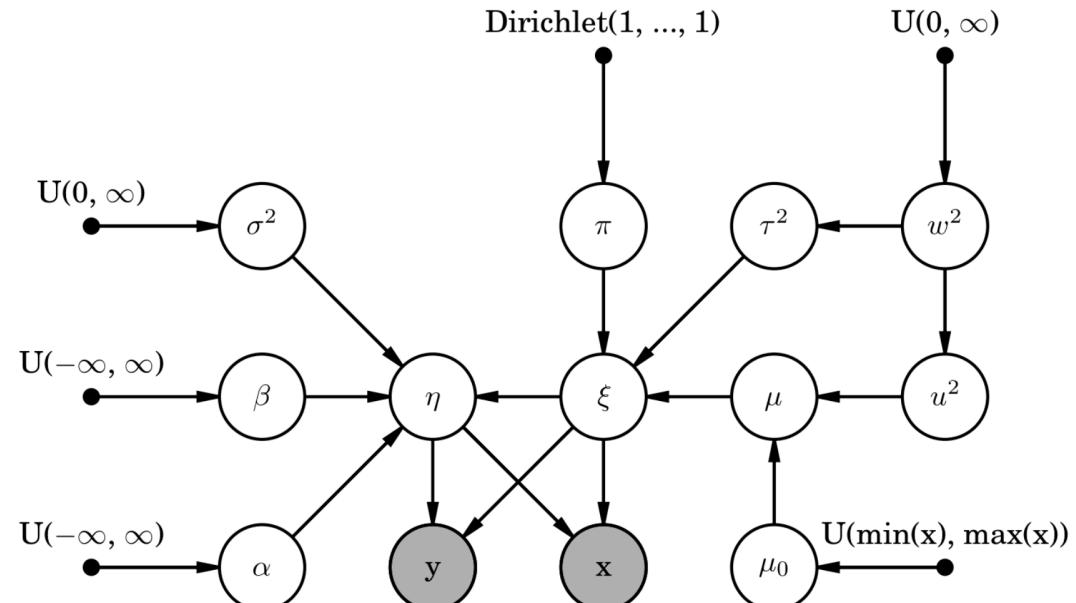
$U(x,y)$: uniform between x and y

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
- 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 general 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

Literature

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