

Probabilistic & Unsupervised Learning

Beyond linear Gaussian and Mixture models

Yee Whye Teh

`ywteh@gatsby.ucl.ac.uk`

**Gatsby Computational Neuroscience Unit, and
MSc in Intelligent Systems, Dept Computer Science
University College London**

Term 1, Autumn 2007

Models We've Learned About So Far

- Factor analysis, principle components analysis, Probabilistic PCA.
- Linear regression, Gaussian processes.
- Mixture of Gaussians, mixture of experts.
- Hidden Markov models, linear Gaussian state space models.

Models consisting of various combinations of:

- Linear Gaussian,
- mixture,
- dynamical,

See Roweis & Ghahramani (1999) A Unifying Review of Linear Gaussian Models.

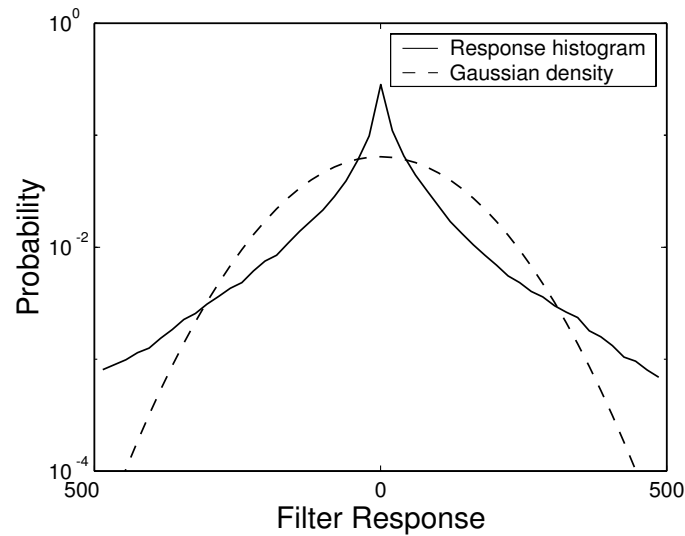
There is a need to go beyond such models. In this lecture we'll learn about

- hierarchical models,
- distributed models,
- Nonlinear models,
- Non-Gaussian models.

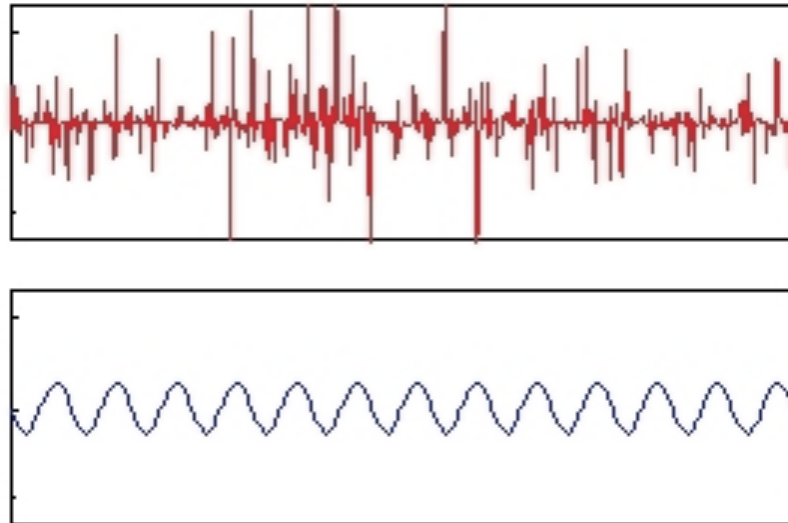
and various combinations thereof.

Why We Need ... Nonlinear/Non-Gaussian Models

... most of the world is not linear nor Gaussian...

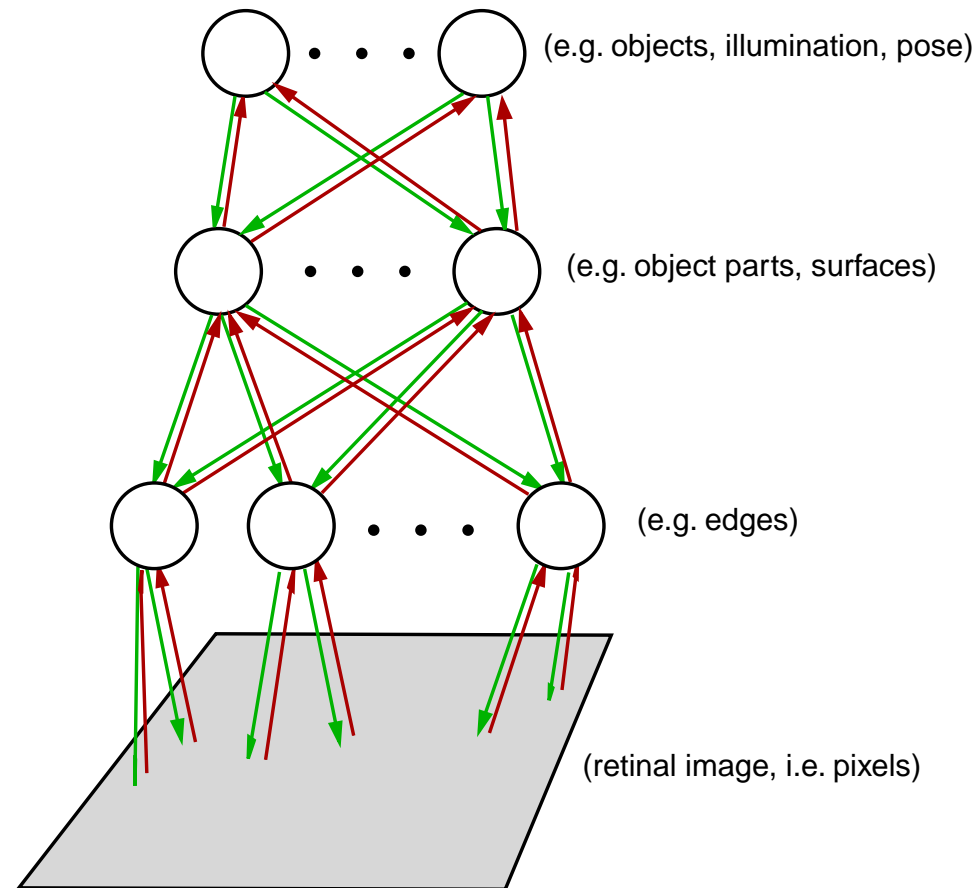


... most interesting structure we would like to learn about is not either ...



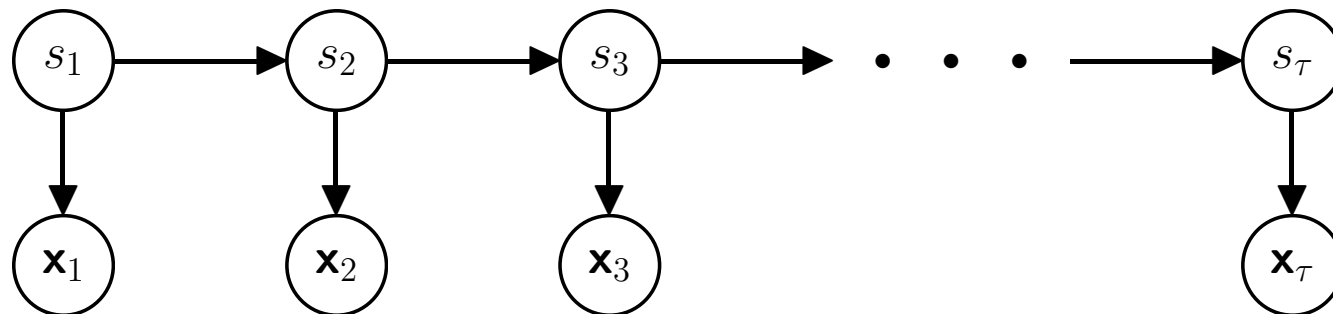
Why We Need ... Hierarchical Models

Many generative processes can be naturally described at different levels of detail.



Biology seems to have developed hierarchical representations.

Why We Need ... Distributed Models



Consider a hidden Markov model. To capture N bits of information about the history of the sequence, an HMM *requires* $K = 2^N$ states!

In a **distributed representation** each data point is represented by a vector of (discrete or continuous) attributes. Some attributes might be **latent**.

For example, you could cluster an electorate into Labour, Tory, Lib-Dem and Undecided, but this is not a distributed representation since each person is described by a single 4-valued discrete variable. A distributed representation might be: (Tory, Single, Black, Female, 18-35 years old, City-dweller, Liberal, Procedural). We might use such a representation to model voting preferences.

These attributes resemble **factors**, but may be discrete (and non-Gaussian), and may outnumber the observed dimensions (say voting preference). Such distributed representations can be exponentially more efficient than clustering.

More Complex Unsupervised Learning Methods

- Nonlinear dimensionality reduction methods
 - Independent components analysis (ICA)
 - Hierarchical clustering
 - Boltzmann machines
 - Sigmoid belief networks
 - Latent Dirichlet allocation
 - Gaussian process latent variable models
-
- Hierarchical HMMs
 - Factorial HMMs
 - Dynamic Bayesian networks
 - Nonlinear dynamical systems

Nonlinear Dimensionality Reduction

There are many ways of generalising PCA and FA models to deal with data which lies on a nonlinear manifold:

- Principal curves
- Autoencoders
- Generative topographic mappings (GTM) and Kohonen self-organising maps (SOM)
- Density networks
- Stochastic Neighbour Embedding
- Multi-dimensional scaling (MDS)
- Isomap: <http://web.mit.edu/cocosci/isomap/isomap.html>
- Locally linear embedding (LLE): <http://www.cs.toronto.edu/~roweis/lle/>
- Gaussian Process Latent Variable Models (GPLVM)

Unfortunately, we don't have time to cover these methods in the course... except for GPLVM.

Hierarchical Clustering

(See Duda and Hart, 1973)

Data $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$

Initialise number of clusters $c = n$

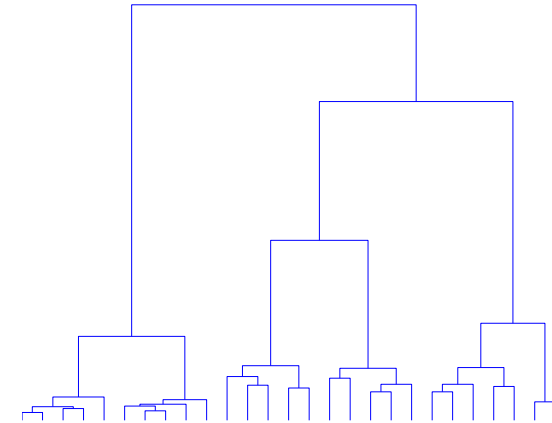
Initialise $\mathcal{D}_i = \{\mathbf{x}^{(i)}\}$ for $i = 1, \dots, c$

while $c > 1$ **do**

 Find nearest pair of clusters \mathcal{D}_i and \mathcal{D}_j

 Merge $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \mathcal{D}_j$, Delete \mathcal{D}_j , $c \leftarrow c - 1$

end while



Distance Measures:

$$d_{\min}(\mathcal{D}_i, \mathcal{D}_j) = \min_{\mathbf{x} \in \mathcal{D}_i, \mathbf{x}' \in \mathcal{D}_j} \|\mathbf{x} - \mathbf{x}'\|$$

single-linkage

$$d_{\max}(\mathcal{D}_i, \mathcal{D}_j) = \max_{\mathbf{x} \in \mathcal{D}_i, \mathbf{x}' \in \mathcal{D}_j} \|\mathbf{x} - \mathbf{x}'\|$$

complete-linkage

$$d_{\text{avg}}(\mathcal{D}_i, \mathcal{D}_j) = \frac{1}{n_i n_j} \sum_{\mathbf{x} \in \mathcal{D}_i} \sum_{\mathbf{x}' \in \mathcal{D}_j} \|\mathbf{x} - \mathbf{x}'\|$$

average-linkage

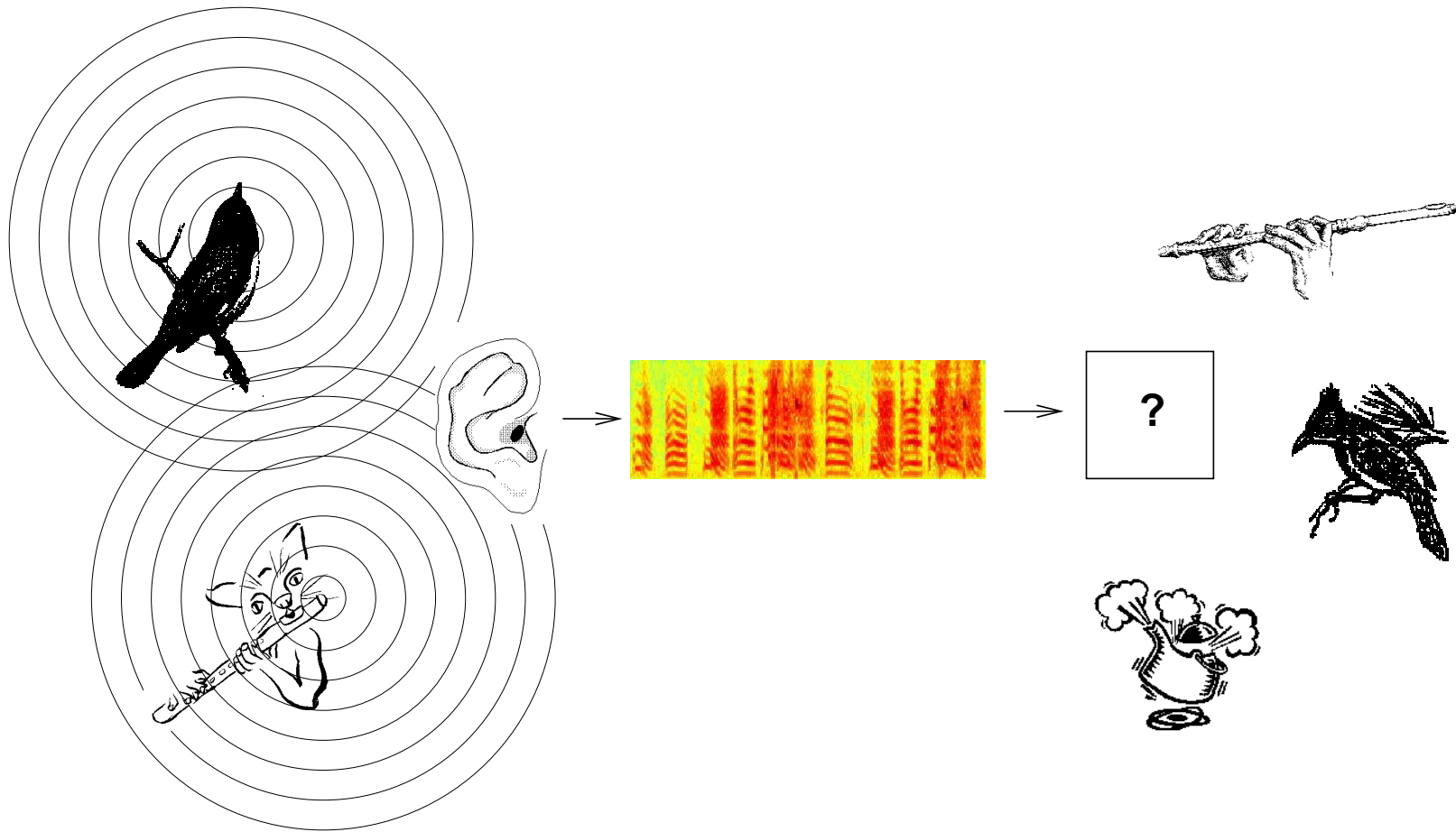
$$d_{\text{mean}}(\mathcal{D}_i, \mathcal{D}_j) = \|\mathbf{m}_i - \mathbf{m}_j\|$$

mean-linkage

Hierarchical clustering is very widely used, e.g. in bioinformatics, because it is often natural to think of data points at multiple level of granularity, or as having been generated by an evolutionary process

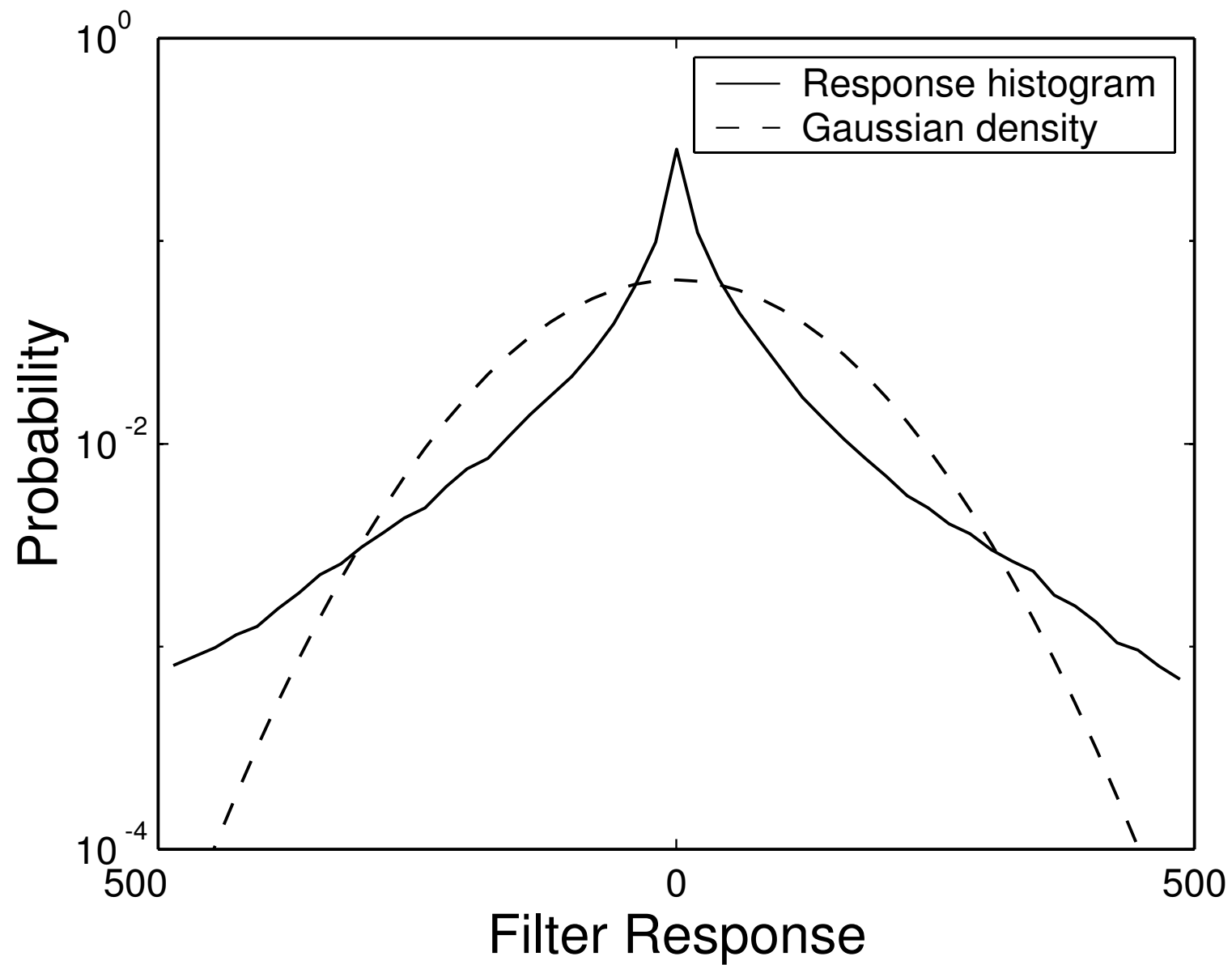
Blind Source Separation

Aka the cocktail party problem.

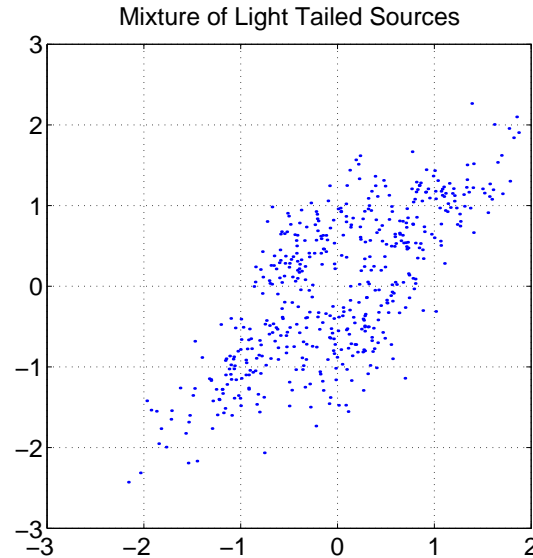
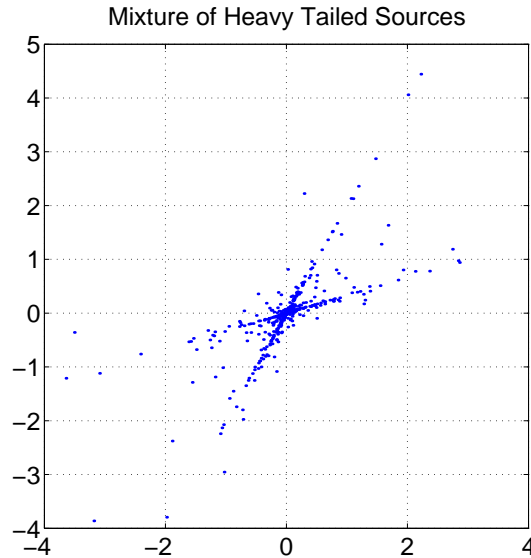


- Given auditory signals (from one or more receivers), recover the different sources of sounds.
- Independent components analysis: assumes that sources are independent, and are non-Gaussian.

Natural Scenes and Sounds



Independent Components Analysis

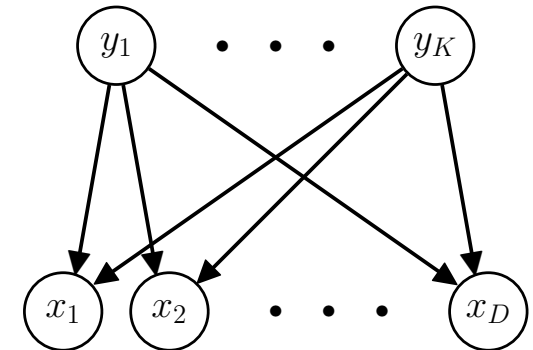


These distributions are generated by linearly combining (or **mixing**) two **non-Gaussian** sources.

- The ICA graphical model is identical to factor analysis:

$$x_d = \sum_{k=1}^K \Lambda_{dk} y_k + \epsilon_d$$

with $y_k \sim P_y$ non-Gaussian.



Differences:

- Well-posed even with $K \geq D$ (e.g., $K = D = 2$ above).
- With non-zero noise, **MAP inference** is non-linear, and the full posterior is non-Gaussian.
- This makes making **exact inference and learning** difficult for most P_y .

Square, Noiseless Causal ICA

- The special case of $K = D$, and **zero observation noise** has been studied extensively (standard **infomax** ICA, c.f. PCA):

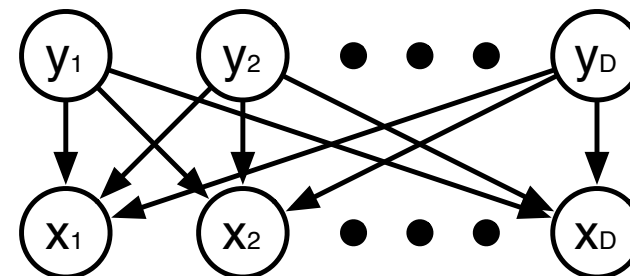
$$\mathbf{x} = \Lambda \mathbf{y} \quad \text{which implies} \quad \mathbf{y} = W \mathbf{x} \quad \text{where} \quad W = \Lambda^{-1}$$

where \mathbf{y} are the independent components (factors), \mathbf{x} are the observations, and W is the unmixing matrix.

- The likelihood can be written in terms of W :

$$P(\mathbf{x}|W) = |W| \prod_k P_y(\underbrace{[W\mathbf{x}]_k}_{y_k})$$

where p_y is marginal probability distribution of factors.



- The likelihood can be obtained by transforming the density of \mathbf{y} to that of \mathbf{x} . If $F : \mathbf{y} \mapsto \mathbf{x}$ is a differentiable bijection, and if $d\mathbf{y}$ is a small neighbourhood around \mathbf{y} , then

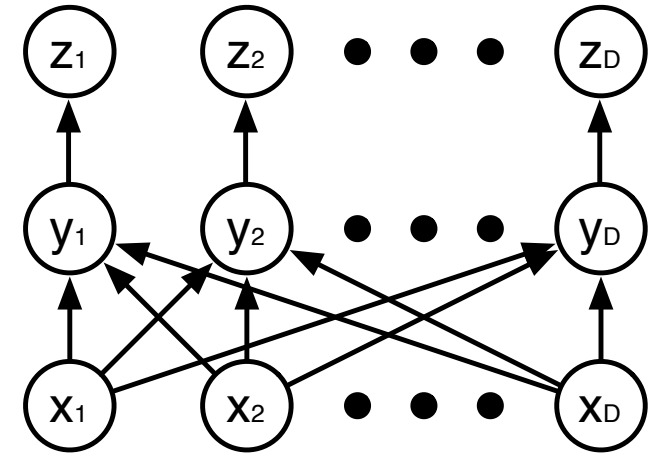
$$P_x(\mathbf{x})d\mathbf{x} = P_y(\mathbf{y})d\mathbf{y} = P_y(F^{-1}(\mathbf{x})) \left| \frac{d\mathbf{y}}{d\mathbf{x}} \right| d\mathbf{x} = P_y(F^{-1}(\mathbf{x})) |\nabla F^{-1}| d\mathbf{x}$$

Infomax ICA

- Consider a feedforward model:

$$y_i = W_i \mathbf{x} \quad z_i = f_i(y_i)$$

with a monotonic squashing function $f_i(-\infty) = 0$,
 $f_i(+\infty) = 1$.



- Infomax find filtering weights W maximizing the **information** carried by \mathbf{z} about \mathbf{x} :

$$\operatorname{argmax}_W I(\mathbf{x}; \mathbf{z}) = \operatorname{argmax}_W H(\mathbf{z}) - H(\mathbf{z}|\mathbf{x}) = \operatorname{argmax}_W H(\mathbf{z})$$

Thus we just have to maximize entropy of \mathbf{z} : make it as uniform as possible on $[0, 1]$ (note squashing function).

- But if data were generated from a square noiseless causal ICA then best we can do is if

$$z_i = f_i(y_i) = \text{cdf}_i(y_i) \quad \text{and} \quad W = \Lambda^{-1}$$

Infomax ICA \Leftrightarrow square noiseless causal ICA.

- Another view: **redundancy reduction** in the representation \mathbf{z} of the data \mathbf{x} .

$$\operatorname{argmax}_W H(\mathbf{z}) = \operatorname{argmax}_W \sum_i H(z_i) - I(z_1, \dots, z_D)$$

See: <http://www.cnl.salk.edu/~tony/ica.html> (a bit out-of-date). MacKay (1996), Pearlmutter and Parra 1996, Cardoso 1997 for equivalence, Teh et al (2003) for an energy-based view.

Learning in ICA

- Log likelihood of data:

$$\log P(\mathbf{x}) = \log |W| + \sum_i \log P_y(W_i \mathbf{x})$$

- Learning by gradient ascent:

$$\nabla W = W^{-T} + g(\mathbf{y}) \mathbf{x}^T \qquad g(y) = \frac{\partial \log P_y(y)}{\partial y}$$

- Better approach: natural gradient

$$\nabla W = W + g(\mathbf{y}) \mathbf{y}^T W$$

(see MacKay 1996).

- Note: we can't use EM in the square noiseless causal ICA model. Why?

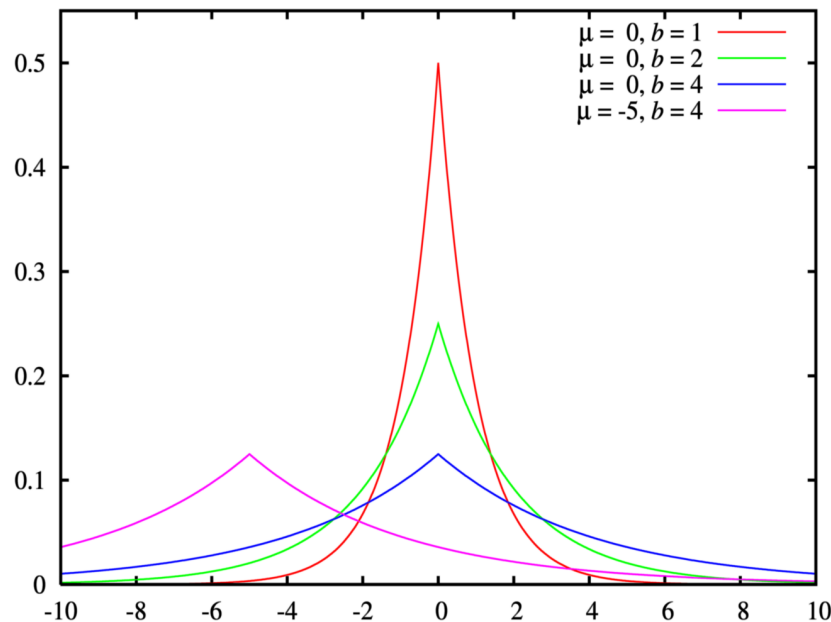
Kurtosis

The **kurtosis** (or excess kurtosis) measures how “peaky” or “heavy-tailed” a distribution is.

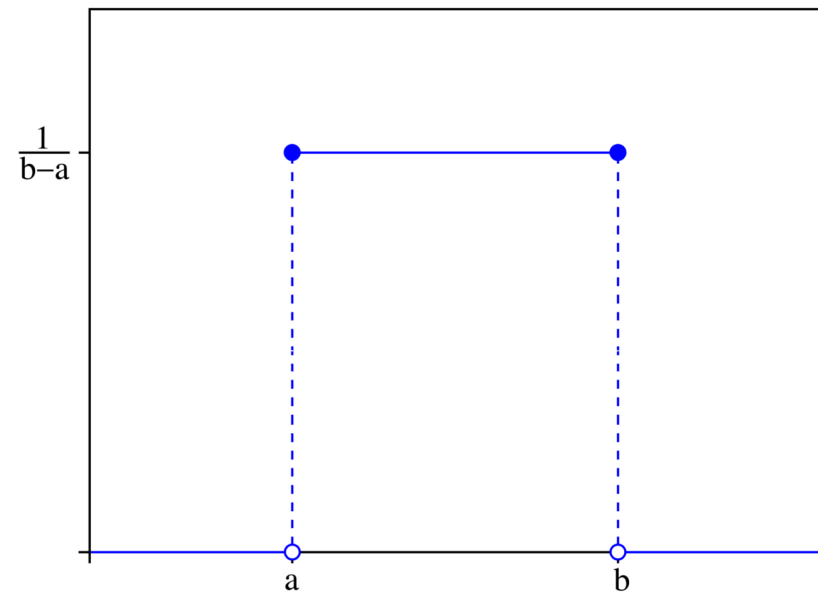
$$K = \frac{E((x - \mu)^4)}{E((x - \mu)^2)^2} - 3$$

where $\mu = E(x)$ is the mean of x .

Gaussian distributions have zero kurtosis.



Heavy tailed distributions have positive kurtosis (leptokurtic).



Light tailed distributions have negative kurtosis (platykurtic).

Some ICA algorithms are essentially **kurtosis pursuit** approaches. Possibly fewer assumptions about generating distributions.

ICA and BSS

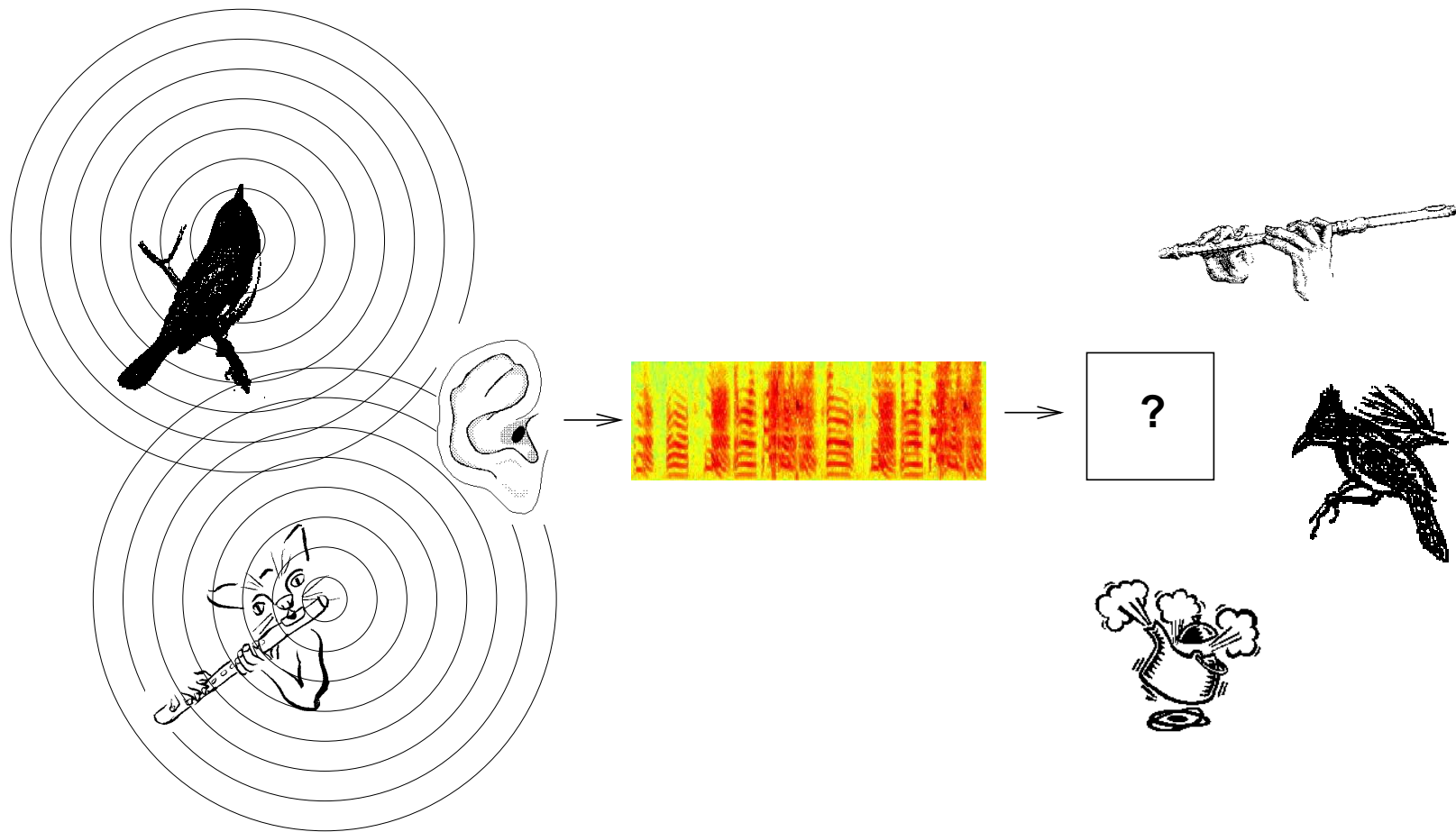
Applications:

- Separating auditory sources
- Analysis of EEG data
- Analysis of functional MRI data
- Natural scene analysis
- ...

Extensions:

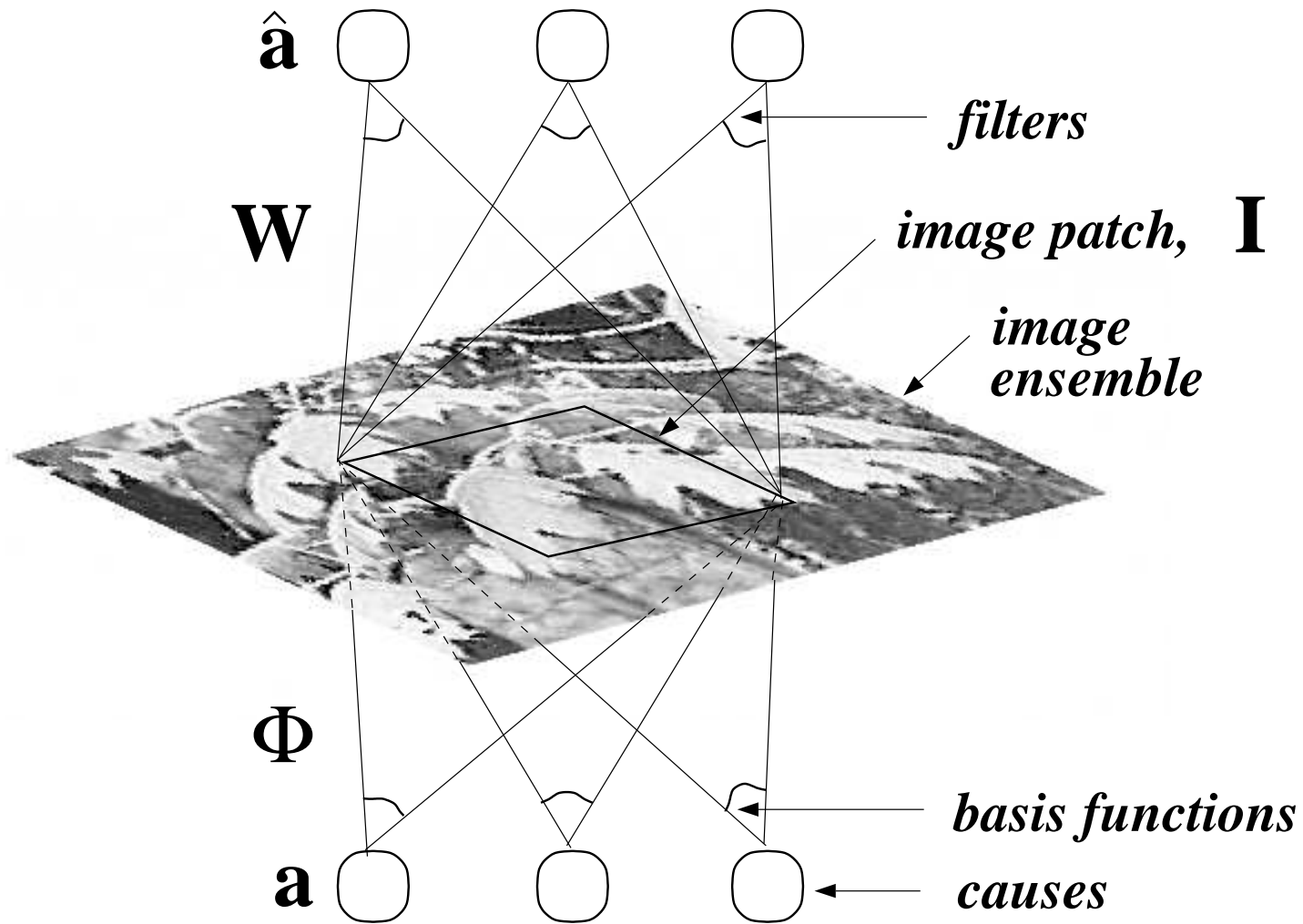
- Non-zero output noise – approximate posteriors and learning.
- Undercomplete ($K < D$) or overcomplete ($K > D$).
- Learning prior distributions (on \mathbf{y}).
- Dynamical hidden models (on \mathbf{y}).
- Learning number of sources.
- Time-varying mixing matrix.
- Nonparametric (kernel) ICA.
- ...

Blind Source Separation

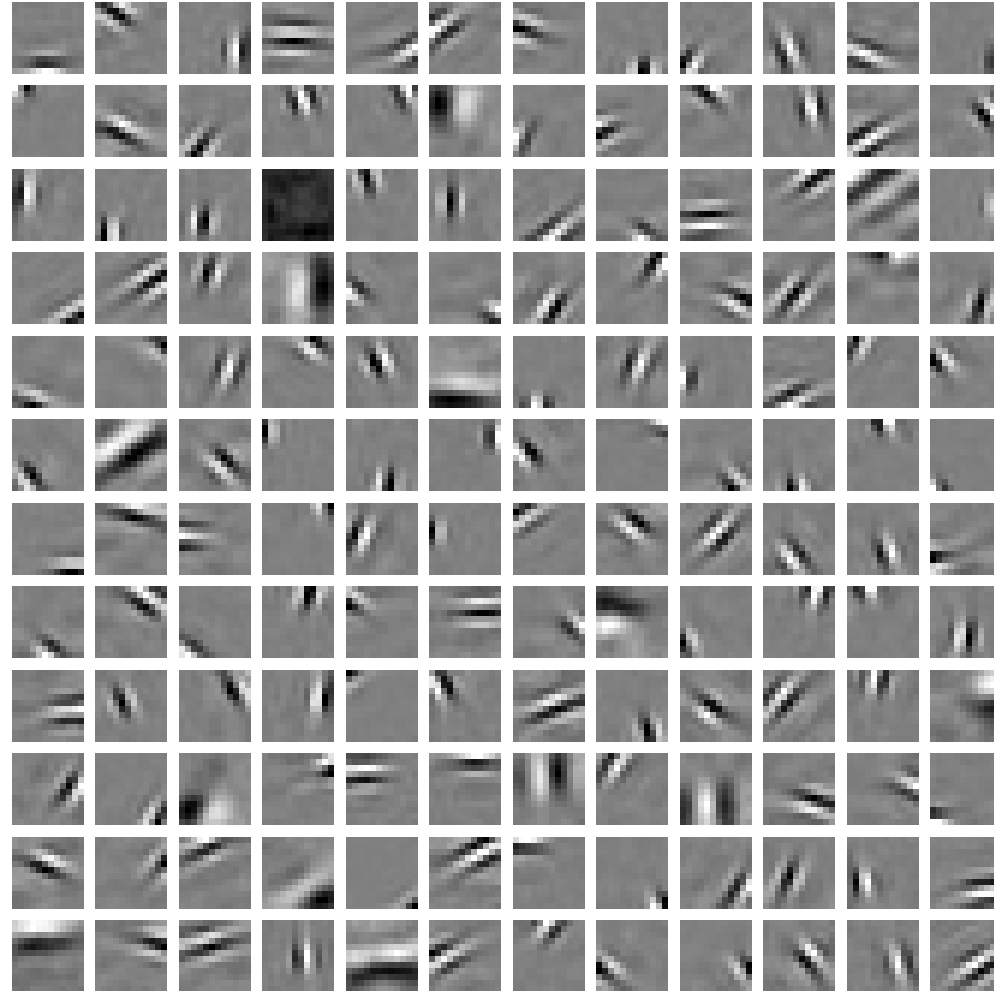


- ICA solution to blind source separation assumes no dependence across time; still works fine much of the time.
- Many algorithms: DCA, SOBI, JADE, ...

Images



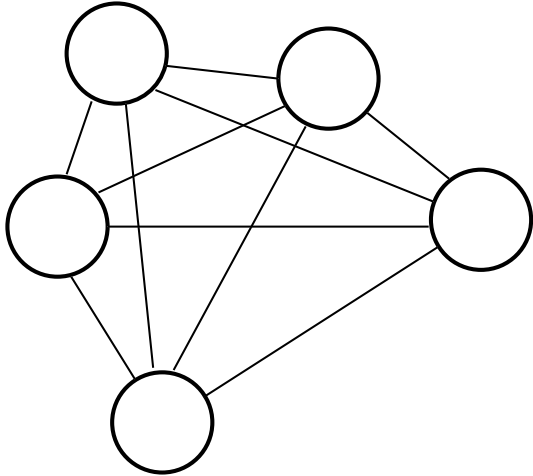
Natural Scenes



Olshausen & Field (1996). Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images.

Nature **381**:607-609.

Boltzmann Machines



Undirected graphical model (i.e. a Markov network) over a vector of binary variables $s_i \in \{0, 1\}$. Some variables may be **hidden**, some may be **visible** (observed).

$$P(\mathbf{s}|W, \mathbf{b}) = \frac{1}{Z} \exp \left\{ \sum_{ij} W_{ij} s_i s_j - \sum_i b_i s_i \right\}$$

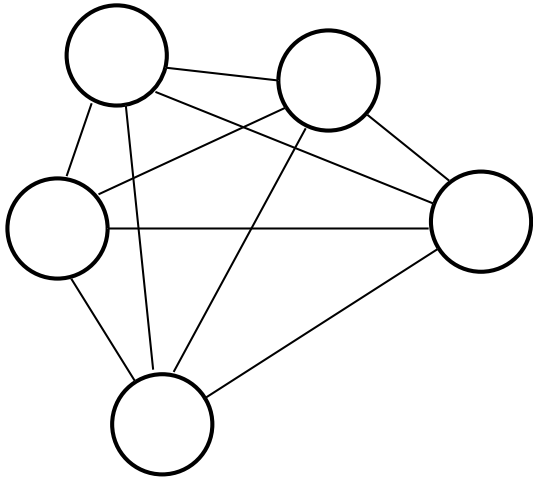
where Z is the normalization constant (partition function).

Learning algorithm: a gradient version of EM

- E step involves computing averages w.r.t. $P(\mathbf{s}^H | \mathbf{s}^V, W, \mathbf{b})$ (“clamped phase”). This could be done either exactly or (more usually) approximately using Gibbs sampling or loopy BP.
- The M step requires gradients w.r.t. Z , which can be computed by averages w.r.t. $P(\mathbf{s}|W, \mathbf{b})$ (“unclamped phase”).

$$\nabla W_{ij} = \langle s_i s_j \rangle_c - \langle s_i s_j \rangle_u$$

Learning in Boltzmann Machines



$$\log P(\mathbf{s}^V \mathbf{s}^H | W, \mathbf{b}) = \sum_{ij} W_{ij} s_i s_j - \sum_i b_i s_i - \log Z$$

with $Z = \sum_{\mathbf{s}} e^{\sum_{ij} W_{ij} s_i s_j - \sum_i b_i s_i}$

Generalised (gradient M-step) EM requires parameter step

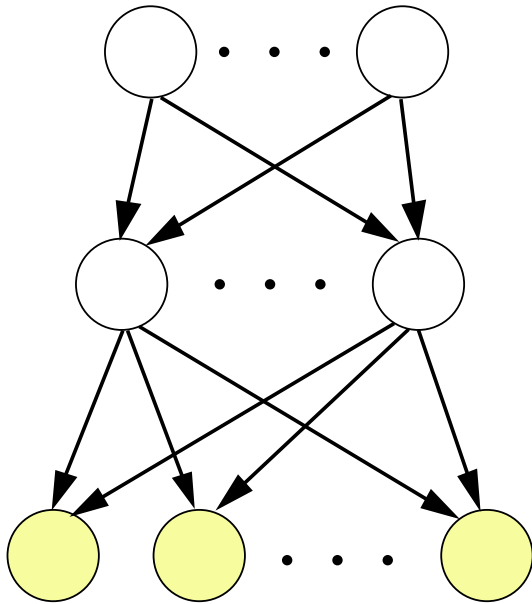
$$\Delta W_{ij} \propto \frac{\partial}{\partial W_{ij}} \langle \log P(\mathbf{s}^V \mathbf{s}^H | W, \mathbf{b}) \rangle_{P(\mathbf{s}^H | \mathbf{s}^V)}$$

Write $\langle \rangle_c$ (**clamped**) for expectations under $P(\mathbf{s} | \mathbf{s}^V)$ (with delta function $P(\mathbf{s}^V | \mathbf{s}^V)$). Then

$$\begin{aligned} \nabla W_{ij} &= \frac{\partial}{\partial W_{ij}} \left[\sum_{ij} W_{ij} \langle s_i s_j \rangle_c - \sum_i b_i \langle s_i \rangle_c - \log Z \right] \\ &= \langle s_i s_j \rangle_c - \frac{\partial}{\partial W_{ij}} \log Z \\ &= \langle s_i s_j \rangle_c - \frac{1}{Z} \frac{\partial}{\partial W_{ij}} \sum_{\mathbf{s}} e^{\sum_{ij} W_{ij} s_i s_j - \sum_i b_i s_i} \\ &= \langle s_i s_j \rangle_c - \sum_{\mathbf{s}} \frac{1}{Z} e^{\sum_{ij} W_{ij} s_i s_j - \sum_i b_i s_i} s_i s_j \\ &= \langle s_i s_j \rangle_c - \sum_{\mathbf{s}} P(\mathbf{s} | W, \mathbf{b}) s_i s_j = \langle s_i s_j \rangle_c - \langle s_i s_j \rangle_u \end{aligned}$$

with $\langle \rangle_u$ (**unclamped**) an expectation under the current joint distribution.

Sigmoid Belief Networks



Directed graphical model (i.e. a Bayesian network) over a vector of binary variables $s_i \in \{0, 1\}$.

$$P(\mathbf{s}|W, \mathbf{b}) = \prod_i P(s_i | \{s_j\}_{j < i}, W, \mathbf{b})$$

$$P(s_i = 1 | \{s_j\}_{j < i}, W, \mathbf{b}) = \frac{1}{1 + \exp\{-\sum_{j < i} W_{ij}s_j - b_i\}}$$

A probabilistic version of sigmoid multilayer perceptrons (“neural networks”).

Learning algorithm: a gradient version of EM

- E step involves computing averages w.r.t. $P(\mathbf{s}_H | \mathbf{s}_V, W, \mathbf{b})$. This could be done either exactly or approximately using Gibbs sampling or mean field approximations.
- Unlike Boltzmann machines, there is no partition function, so no need for an unclamped phase in the M step.

Topic Modelling

Topic modelling: given a corpus of documents, find the “topics” discussed by the documents in the corpus.

Example: abstracts of papers from the Proceedings of the National Academy of Sciences (PNAS).

Global climate change and mammalian species diversity in U.S. national parks

National parks and bioreserves are key conservation tools used to protect species and their habitats within the confines of fixed political boundaries. This inflexibility may be their “Achilles’ heel” as conservation tools in the face of emerging global-scale environmental problems such as climate change. Global climate change, brought about by rising levels of greenhouse gases, threatens to alter the geographic distribution of many habitats and their component species....

The influence of large-scale wind power on global climate

Large-scale use of wind power can alter local and global climate by extracting kinetic energy and altering turbulent transport in the atmospheric boundary layer. We report climate-model simulations that address the possible climatic impacts of wind power at regional to global scales by using two general circulation models and several parameterizations of the interaction of wind turbines with the boundary layer....

Twentieth century climate change: Evidence from small glaciers

The relation between changes in modern glaciers, not including the ice sheets of Greenland and Antarctica, and their climatic environment is investigated to shed light on paleoglacier evidence of past climate change and for projecting the effects of future climate warming on cold regions of the world. Loss of glacier volume has been more or less continuous since the 19th century, but it is not a simple adjustment to the end of an “anomalous” Little Ice Age....

Topic Modelling

Example topics discovered from PNAS abstracts (each topic represented in terms of the top 5 most common words in that topic).

217 INSECT MYB PHEROMONE LENS LARVAE	274 SPECIES PHYLOGENETIC EVOLUTION EVOLUTIONARY SEQUENCES	126 GENE VECTOR VECTORS EXPRESSION TRANSFER	63 STRUCTURE ANGSTROM CRYSTAL RESIDUES STRUCTURES	200 FOLDING NATIVE PROTEIN STATE ENERGY	209 NUCLEAR NUCLEUS LOCALIZATION CYTOPLASM EXPORT
42 NEURAL DEVELOPMENT DORSAL EMBRYOS VENTRAL	2 SPECIES GLOBAL CLIMATE CO2 WATER	280 SPECIES SELECTION EVOLUTION GENETIC POPULATIONS	15 CHROMOSOME REGION CHROMOSOMES KB MAP	64 CELLS CELL ANTIGEN LYMPHOCYTES CD4	102 TUMOR CANCER TUMORS HUMAN CELLS
112 HOST BACTERIAL BACTERIA STRAINS SALMONELLA	210 SYNAPTIC NEURONS POSTSYNAPTIC HIPPOCAMPAL SYNAPSES	201 RESISTANCE RESISTANT DRUG DRUGS SENSITIVE	165 CHANNEL CHANNELS VOLTAGE CURRENT CURRENTS	142 PLANTS PLANT ARABIDOPSIS TOBACCO LEAVES	222 CORTEX BRAIN SUBJECTS TASK AREAS
39 THEORY TIME SPACE GIVEN PROBLEM	105 HAIR MECHANICAL MB SENSORY EAR	221 LARGE SCALE DENSITY OBSERVED OBSERVATIONS	270 TIME SPECTROSCOPY NMR SPECTRA TRANSFER	55 FORCE SURFACE MOLECULES SOLUTION SURFACES	114 POPULATION POPULATIONS GENETIC DIVERSITY ISOLATES
		109 RESEARCH NEW INFORMATION UNDERSTANDING PAPER	120 AGE OLD AGING LIFE YOUNG		

Recap: Beta Distributions

Remember the Bayesian coin toss example.

$$P(H|q) = q$$

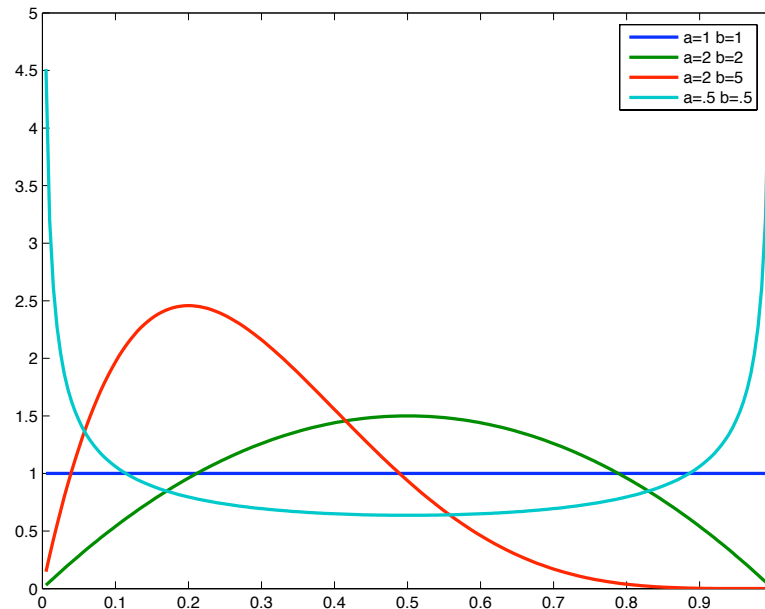
$$P(T|q) = 1 - q$$

The probability of a sequence of coin tosses is:

$$P(HHTT \dots HT|q) = q^{\text{\#heads}}(1 - q)^{\text{\#tails}}$$

A conjugate prior for q is the Beta distribution:

$$P(q) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} q^{a-1} (1-q)^{b-1} \quad a, b \geq 0$$



Dirichlet Distributions

Imagine a Bayesian dice throwing example.

$$P(1|\mathbf{q}) = q_1 \quad P(2|\mathbf{q}) = q_2 \quad P(3|\mathbf{q}) = q_3 \quad P(4|\mathbf{q}) = q_4 \quad P(5|\mathbf{q}) = q_5 \quad P(6|\mathbf{q}) = q_6$$

with $q_i \geq 0$, $\sum_i q_i = 1$. The probability of a sequence of dice throws is:

$$P(34156 \cdots 12|\mathbf{q}) = \prod_{i=1}^6 q_i^{\# \text{ face } i}$$

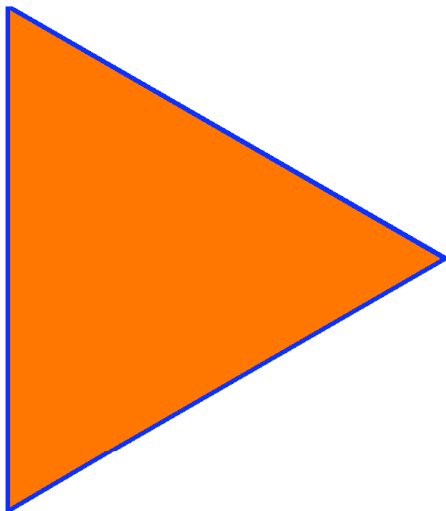
A conjugate prior for \mathbf{q} is the Dirichlet distribution:

$$P(\mathbf{q}) = \frac{\Gamma(\sum_i a_i)}{\prod_i \Gamma(a_i)} \prod_i q_i^{a_i-1}$$

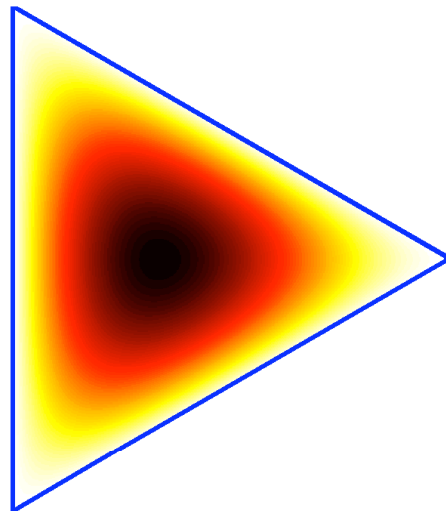
$$q_i \geq 0, \sum_i q_i = 1$$

$$a_i \geq 0$$

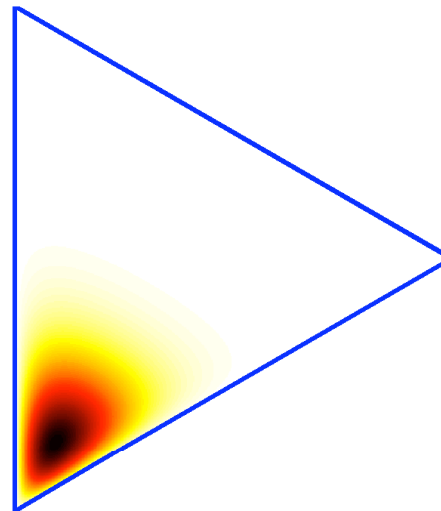
Dirichlet(1,1,1)



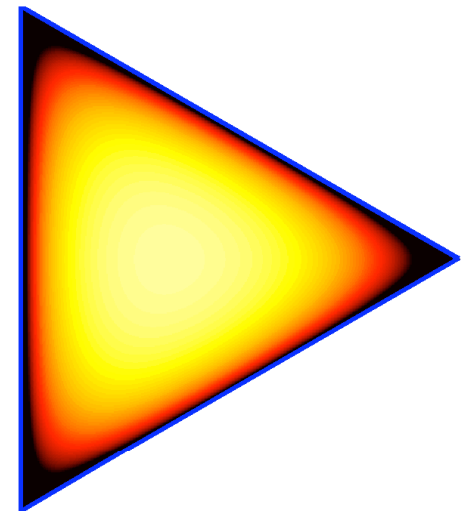
Dirichlet(2,2,2)



Dirichlet(2,10,2)



Dirichlet(0.8,0.8,0.8)



Latent Dirichlet Allocation

Each document is a sequence of words, we model it using a mixture model by ignoring the sequential nature—“bag-of-words” assumption.

- For each document d :

Place a Dirichlet prior on the mixing proportions θ_d ,

$$\theta_d \sim \text{Dir}(\alpha, \dots, \alpha)$$

- For each word i in document d :

Pick a topic,

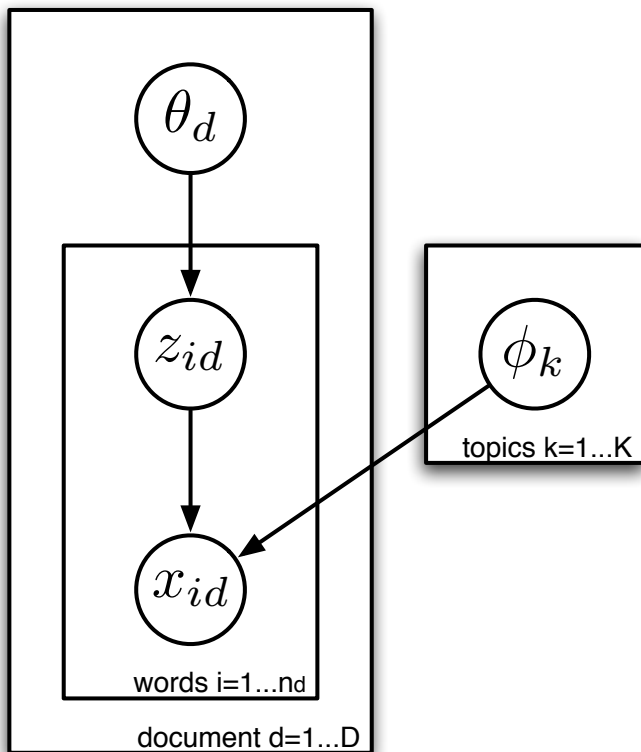
$$z_{id} \sim \text{Discrete}(\theta_d)$$

Pick a word given topic z_{id} ,

$$x_{id} \sim \text{Discrete}(\phi_{z_{id}})$$

- Also place a prior over the topic parameters,

$$\phi_k \sim \text{Dir}(\beta, \dots, \beta)$$



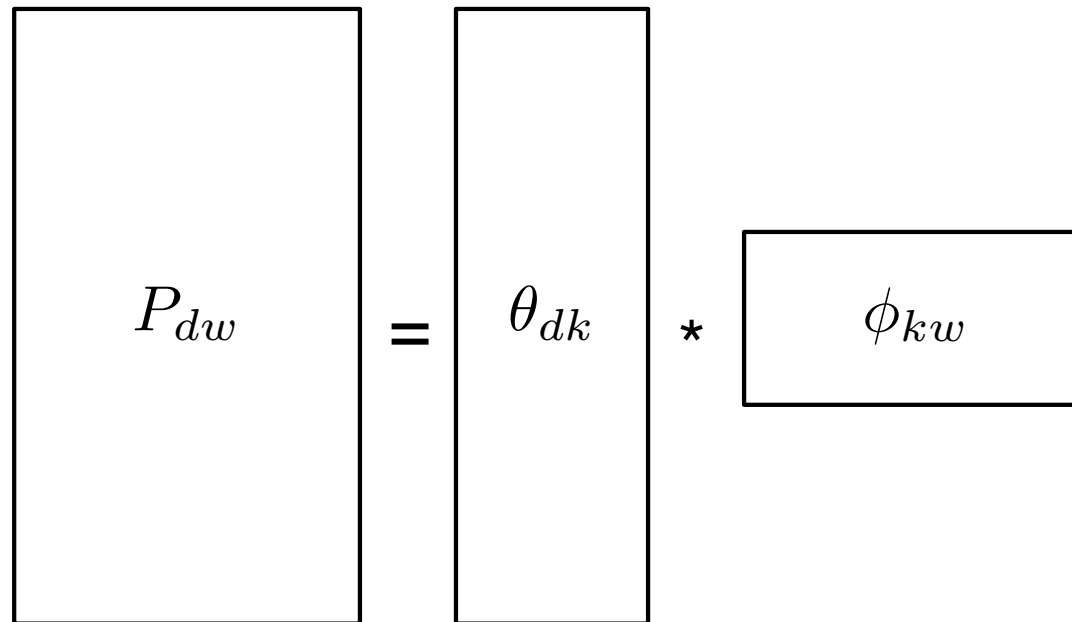
Multiple mixture models, sharing the same set of components (topics).

Latent Dirichlet Allocation as Matrix Decomposition

Let N_{dw} be the number of times word w appears in document d , and P_{dw} is the probability of word w appearing in document d .

$$p(N|P) = \prod_{dw} P_{dw}^{N_{dw}} \quad \text{likelihood term}$$

$$P_{dw} = \sum_k p(\text{pick topic } k) p(\text{pick word } w | k) = \sum_{k=1}^K \theta_{dk} \phi_{kw}$$



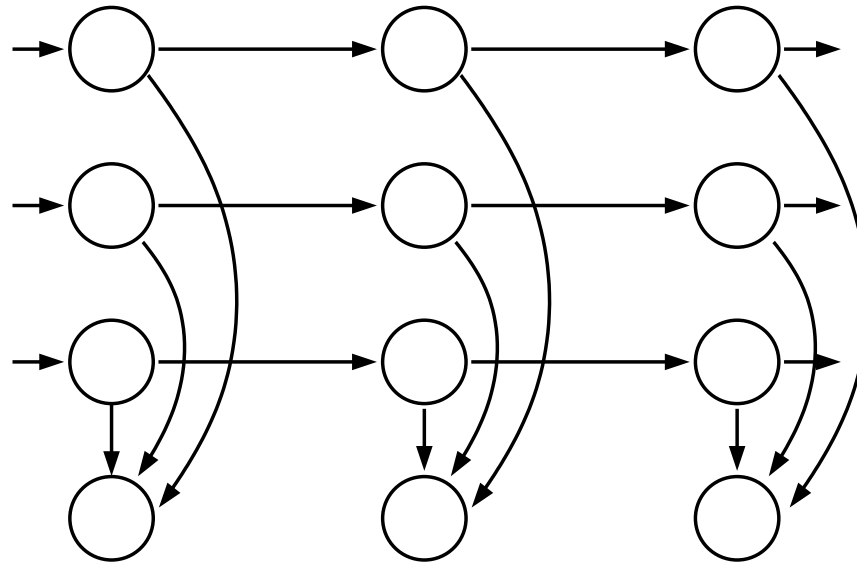
A diagram illustrating the matrix decomposition of the word-document probability matrix P_{dw} . It consists of three rectangular boxes arranged horizontally. The first box on the left is tall and contains the symbol P_{dw} . To its right is an equals sign. The second box is also tall and contains the symbol θ_{dk} . To its right is an asterisk symbol $*$. The third box is shorter and wider, containing the symbol ϕ_{kw} . This visualizes the equation $P_{dw} = \theta_{dk} * \phi_{kw}$.

This decomposition is similar to PCA and factor analysis, except not Gaussian.

Latent Dirichlet Allocation

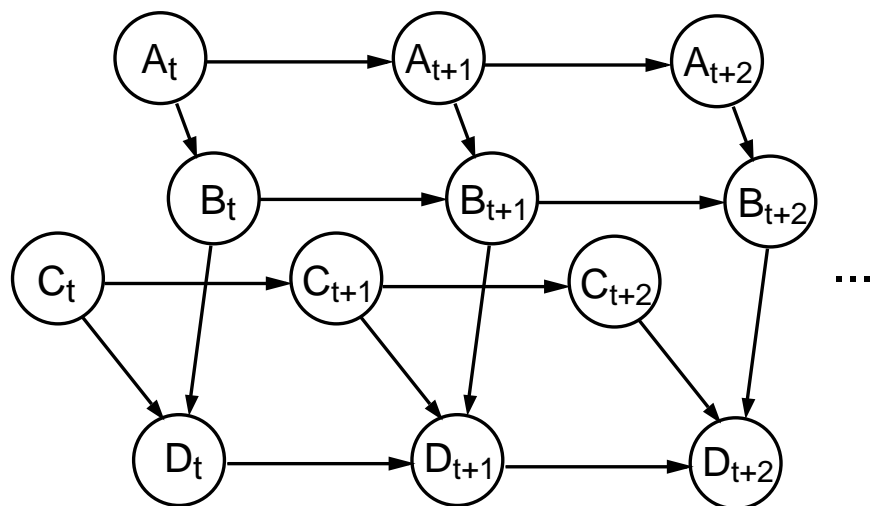
- Exact inference in latent Dirichlet allocation is intractable, and typically either variational or Markov chain Monte Carlo approximations are deployed.
- Latent Dirichlet allocation is an example of a [mixed membership model](#) from statistics.
- Latent Dirichlet allocation has also been applied to computer vision, social network modelling, natural language processing...
- Generalizations:
 - Relaxing the bag-of-words assumption (e.g. a Markov model).
 - Modelling changes in topics through time.
 - Modelling correlations among occurrences of topics.
 - Modelling authors, recipients, multiple corpora.
 - Cross modal interactions (images and tags).
 - Nonparametric generalizations.

Factorial Hidden Markov Models



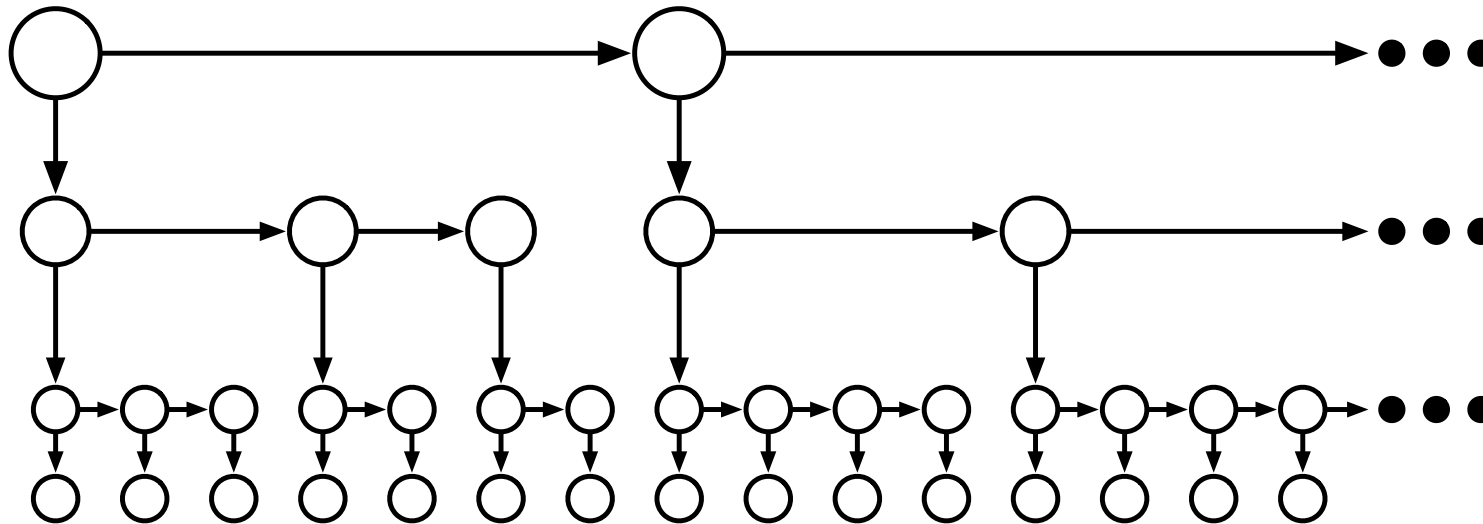
- These are hidden Markov models with many state variables (i.e. a distributed representation of the state).
- Each state variable evolves independently.
- The state can capture many more bits of information about the sequence (linear in the number of state variables).
- E step is usually intractable (due to explaining away in latent states).

Dynamic Bayesian Networks



- Like factorial HMMs but with structured dependencies among latent states.

Hierarchical Hidden Markov Models



- High level HMMs “emit” low level HMMs, recursively.
- Examples: speech recognition (words emit phonemes, phonemes actual audio signals), action recognition (playing football, running, dribbling, kicking, microactions).
- Factorial HMMs, hierarchical HMMs and dynamic Bayesian networks can be reparametrized using straight HMM, but exponentially larger state space.

Gaussian Process Latent Variable Models

Recap: probabilistic PCA

$$\begin{aligned}\mathbf{y}_i | \mathbf{x}_i, \Lambda &\sim \mathcal{N}(\Lambda \mathbf{x}_i, \beta^{-1} I) \\ \mathbf{x}_i &\sim \mathcal{N}(0, I)\end{aligned}$$

Usually: compute posterior over $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top$, maximizing likelihood over Λ .

Suppose we know the values of the latent X , then we can integrate out Λ (c.f. linear regression), giving a conditional probability of $Y = [\mathbf{y}_1 \dots \mathbf{y}_N]^\top$:

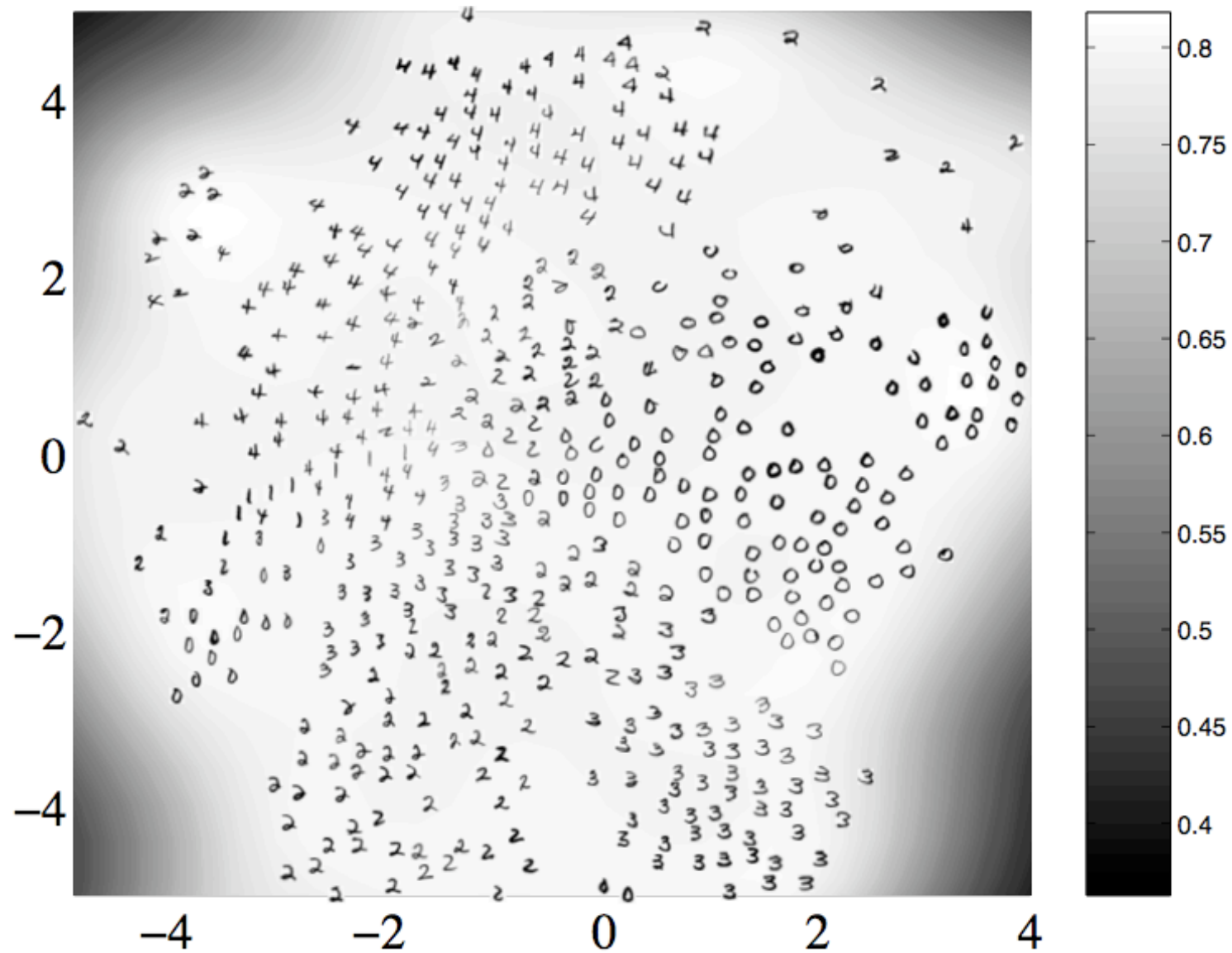
$$\begin{aligned}\Lambda &\sim \mathcal{N}(0, \alpha^{-1} I) \\ p(Y|X) &\sim |2\pi K|^{-\frac{D}{2}} \exp\left(-\frac{1}{2} \text{Tr}[K^{-1} Y Y^\top]\right) \quad K = \alpha X X^\top + \beta I\end{aligned}$$

This is just D independent Gaussian processes, one for each dimension of Y ! Each Gaussian process describes a mapping from latent space \mathbf{x} to one dimension of \mathbf{y} .

Replacing the linear kernel with nonlinear kernels gives nonlinear mappings—nonlinear dimensionality reduction.

But now dependence on X is complicated—instead of computing a posterior over X we now maximize (the likelihood) over it (along with the hyperparameters too).

Gaussian Process Latent Variable Models



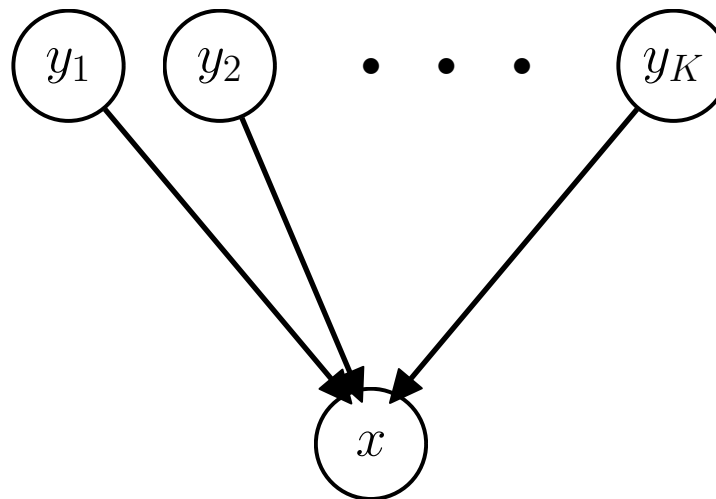
Some video demos...

Intractability

For many probabilistic models of interest, exact inference is not computationally feasible. This occurs for three (main) reasons:

- Distributions may have complicated forms (e.g. non-linearities in generative model).
- “Explaining away” causes coupling from observations
Observing the value of a child induces dependencies amongst its parents.
- Even with simple models, being Bayesian and computing the full posterior over both latent variables and parameters

There is often strong coupling between latent variables and parameters.

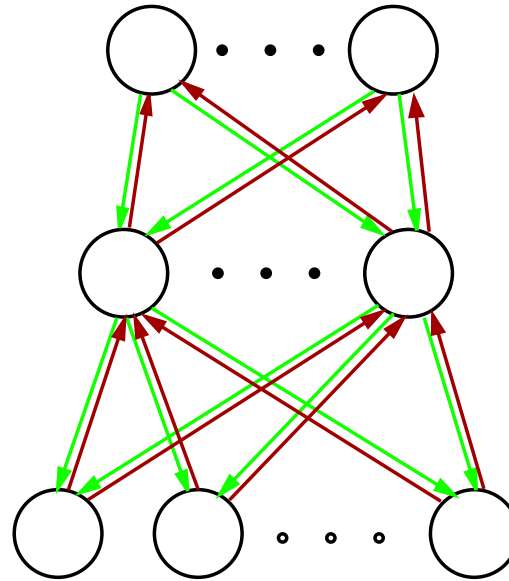


We can still work with such models by using *approximate inference* techniques to estimate the latent variables.

Approximate Inference

- **Linearisation**: Approximate nonlinearities by Taylor series expansion about a point (e.g. the approximate mean or mode of the hidden variable distribution). Linear approximations are particularly useful since Gaussian distributions are closed under linear transformations (e.g., EKF). Also Laplace's approximation.
- **Monte Carlo Sampling**: Approximate posterior distribution over unobserved variables by a set of random samples. We often need **Markov chain Monte carlo** or **sequential Monte Carlo** methods to sample from difficult distributions.
- **Variational Methods**: Approximate the hidden variable posterior $p(H)$ with a tractable form $q(H)$, such that $\mathbf{KL}[q||p]$ is minimised. This gives a lower bound on the likelihood that can be maximised with respect to the parameters of $q(H)$.
- **Local Message Passing Methods**: Approximate the hidden variable posterior $p(H)$ with a tractable form $q(H)$ or with a set of locally consistent tractable forms by other means (loopy belief propagation, expectation propagation).
- **Recognition Models**: Approximate the hidden variable posterior distribution using an explicit *bottom-up* recognition model/network.

Recognition Models



- A model is trained in a supervised way to recover the hidden causes (latent variables) from the observations.
- This may take the form of explicit recognition network (e.g. Helmholtz machine, deep belief networks) which mirrors the generative network. (tractability at the cost of restricted approximating distribution)
- Inference is done in a single *bottom-up* pass (no iteration required).

End Notes

- Independent Component Analysis. Hyvarinen, Karhunen and Oja. John Wiley and Sons, 2001.
- A Learning Algorithm for Boltzmann Machines. Ackley, Hinton and Sejnowski. Cognitive Science 1985.
- Connectionist Learning of Belief Networks. Neal. Artificial Intelligence 1992.
- Latent Dirichlet Allocation. Blei, Ng and Jordan. Journal of Machine Learning Research 2003.
- Gaussian Process Latent Variable Models. Neil Lawrence. Advances in Neural Information Processing Systems 2004.
- GP-LVM in graphics homepage. <http://grail.cs.washington.edu/projects/styleik/>

