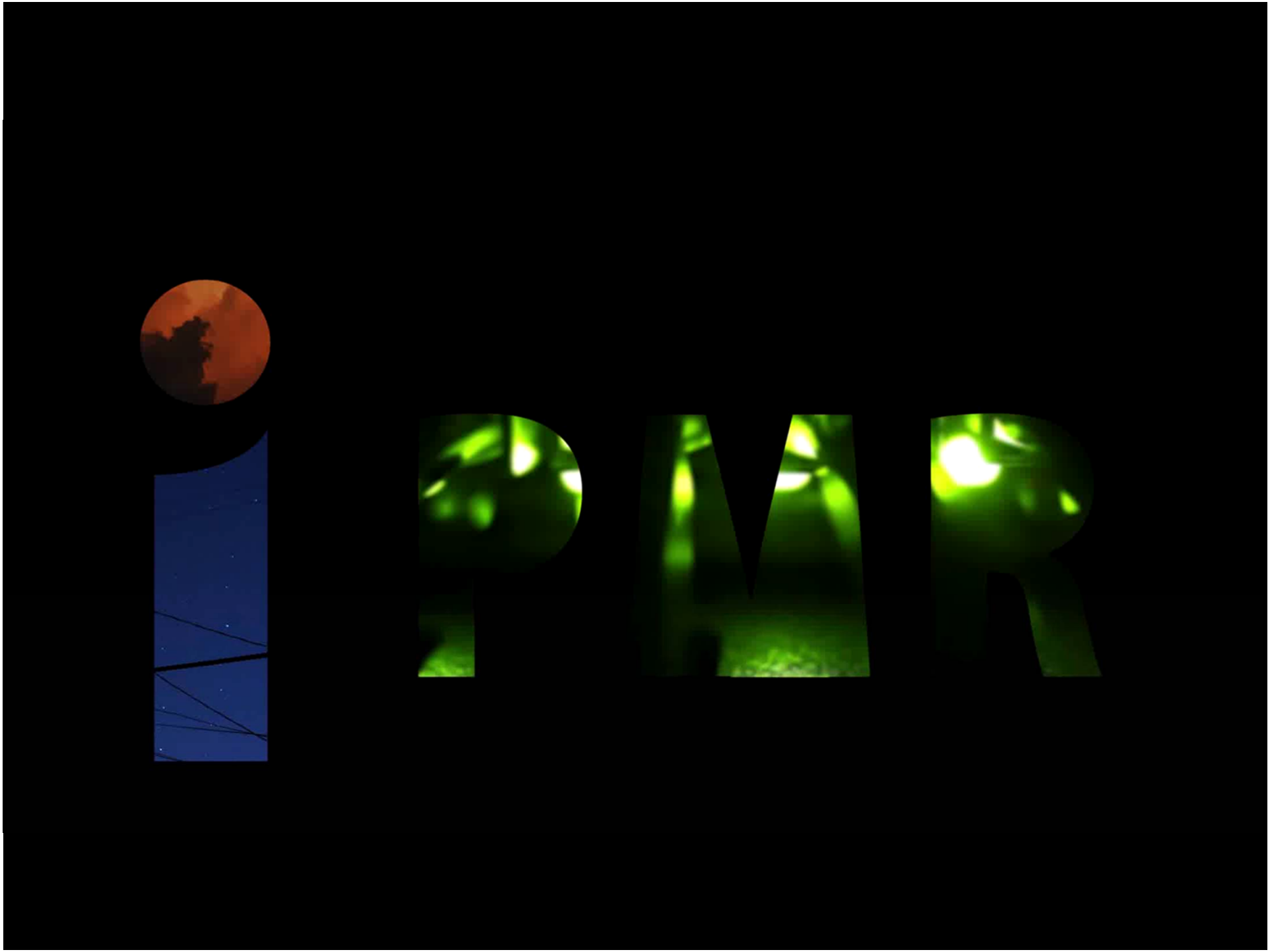

iPMR

Factor Graphs

- Remember we often write our models in the form

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h}))$$

- Here we use *v* for visible and *h* for latent (hidden) variables, at least some of which we care about.

- We found we could use the elimination algorithm to do inference in models of this form.
- This involved passing messages using the structure in *E*
- Worked well in trees.

- But for other network structures it got complicated quickly:
- e.g. eliminating a node causes a joint message to all the nodes it connects to (causing a joint factor).

- Use an approximation scheme.

Reminder: Divergences

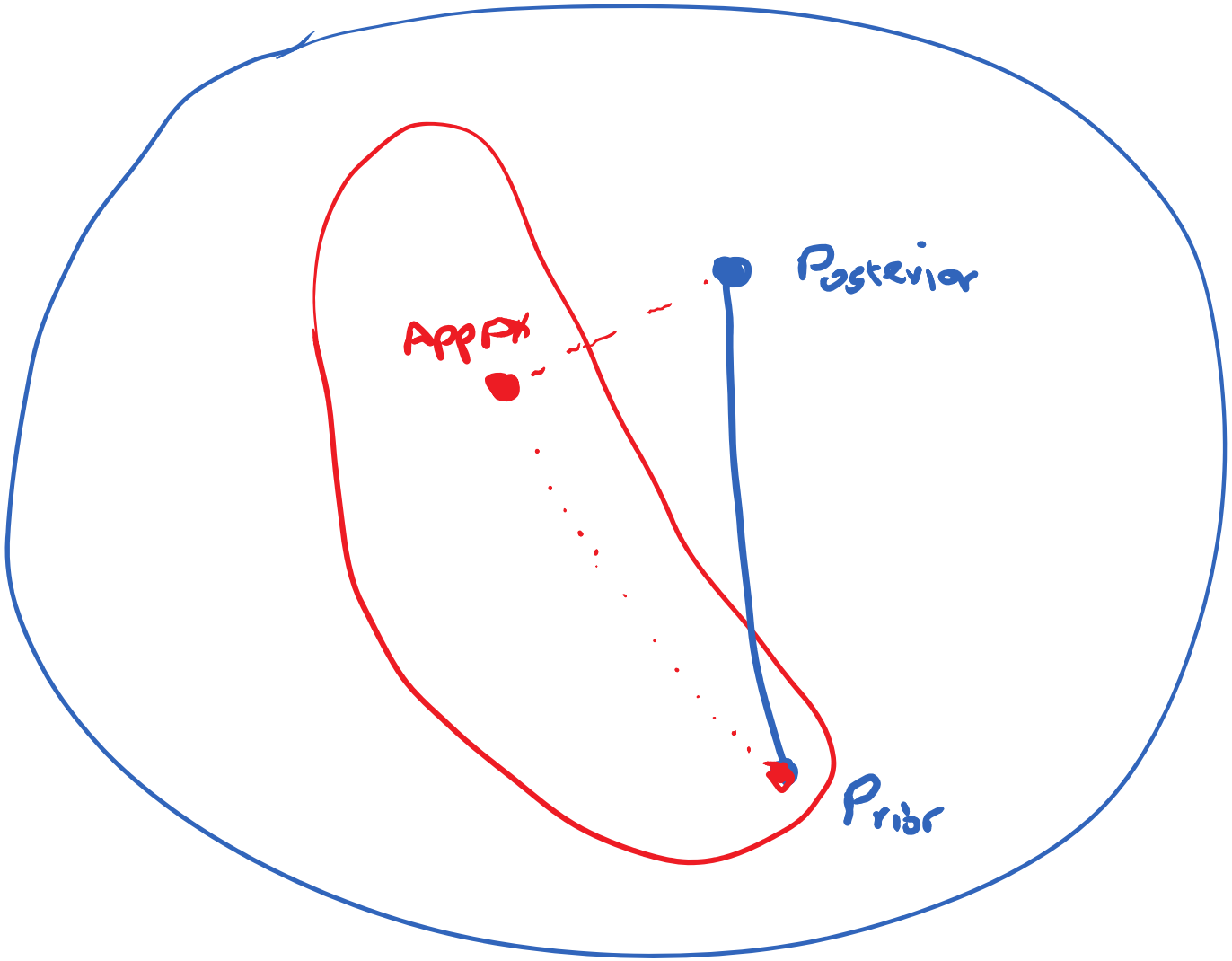
- Divergences measure a cost of using a wrong distribution instead of a correct one.
- E.g. KL(Q||P) – measures the coding cost of coding using the distribution P instead of the true distribution Q.

$$KL(Q(.)||P(.)) = \int d\mathbf{x}\, Q(\mathbf{x}) \log \frac{Q(\mathbf{x})}{P(\mathbf{x})}$$

- Sometimes by constraining the set of distributions we allow, and the ensuring the divergence to the required distribution is low, we can do the computation.

Note: placement of dx is for convenience, all logs are natural logs throughout

Picture



The Free Energy

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h})), \qquad P(\mathbf{h}|\mathbf{v}) = \frac{1}{ZP(\mathbf{v})}\exp(-E(\mathbf{v},\mathbf{h})).$$

The problem is computing $P(\mathbf{v})$ and then marginalising over some of the $\mathbf{h}$ to focus on variables we care about.

Consider matching some distribution $Q(\mathbf{h}|\mathbf{v})$ to the distribution we want: $P(\mathbf{h}|\mathbf{v})$. Minimizing $KL(Q||P)$ gives zero. So

$$0 = \min_Q \sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v})[\log Q(\mathbf{h}|\mathbf{v}) - \log P(\mathbf{h}|\mathbf{v})]$$

Rearranging and using the form for $P(\mathbf{v},\mathbf{h})$ we get

$$\log P(\mathbf{v}) = \max_Q \left[-\sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v})E(\mathbf{v},\mathbf{h}) - \sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v}) \log Q(\mathbf{h}|\mathbf{v}) \right] - \log Z.$$

The Free Energy

More generally

$$\log P(\mathbf{v}) = \left[-\sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v}) E(\mathbf{v},\mathbf{h}) - \sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v}) \log Q(\mathbf{h}|\mathbf{v}) \right] - \log Z + KL(Q||P).$$

The term

$$\sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v}) E(\mathbf{v},\mathbf{h}) + \sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v}) \log Q(\mathbf{h}|\mathbf{v})$$

is the called the "free energy".

Minimizing $KL(Q(\mathbf{h}|\mathbf{v})||P((\mathbf{h}|\mathbf{v}))$ also minimizes the free energy. The (negative of the) minimum of the free energy returns the log probability of the data.

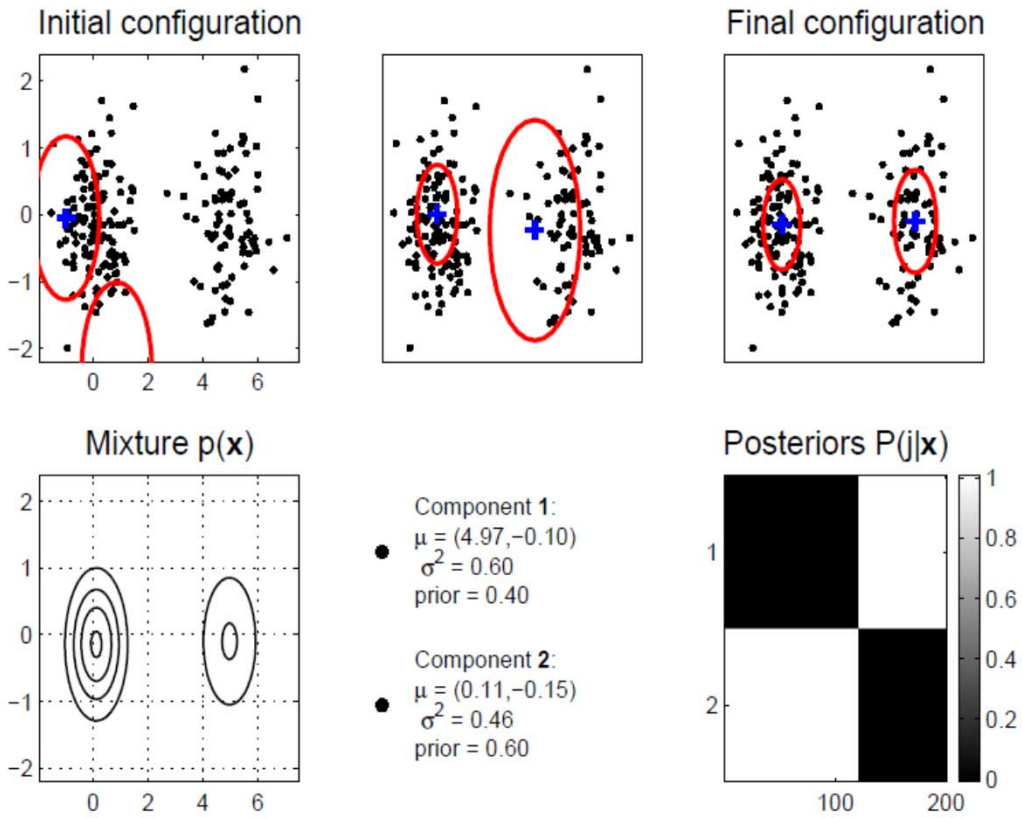
Want to minimize the free energy to get best $Q(\mathbf{h}|\mathbf{v})$. But (1) minimum might be hard to compute, (2) optimal Q might be hard to marginalise to the variables we care about. So we approximate this method by constraining the form of Q to something easy to marginalise. Result: Many different approximate free energies.

Using Free Energies

$$\log P(D|\boldsymbol{\theta}) \geq \sum_n \left[-\sum_{\mathbf{h}^n} Q^n(\mathbf{h}^n) E(\mathbf{v}^n, \mathbf{h}^n|\boldsymbol{\theta}) - \sum_{\mathbf{h}}^{n} Q^n(\mathbf{h}^n) \log Q^n(\mathbf{h}^n) \right] - \log Z.$$

- Example: mixture models, and Expectation Maximization Algorithm:
  - Fix Q, maximize RHS for θ. M-Step.
  - Fix θ, maximize RHS for Q. E-Step.
- Iteration guaranteed to locally maximize log likelihood.
- See Barber and work through EM algorithm for mixture of Gaussians.

Mixture of Gaussians



(Tipping, 1999)

Using Graphical Structure

Making the dependence of Q on $\mathbf{v}$ implicit again, we start from

$$\log P(\mathbf{v}) = \max_{Q} \left[-\sum_{\mathbf{h}} Q(\mathbf{h}) E(\mathbf{v}, \mathbf{h}) - \sum_{\mathbf{h}} Q(\mathbf{h}) \log Q(\mathbf{h}) \right] - \log Z.$$

we can use the graphical structure $-E(\mathbf{v}, \mathbf{h}) = \sum_i \phi_i(\mathbf{v}_{D_i}, \mathbf{h}_{C_i})$ (where C_i and D_i denote the hidden and visible variables in factor i) to get

$$\log P(\mathbf{v}) = \max_{Q} \left[\sum_{i} \sum_{\mathbf{h}_{C_i}} Q(\mathbf{h}_{C_i}) \phi_i(\mathbf{v}_{D_i}, \mathbf{h}_{C_i}) - \sum_{\mathbf{h}} Q(\mathbf{h}) \log Q(\mathbf{h}) \right] - \log Z.$$

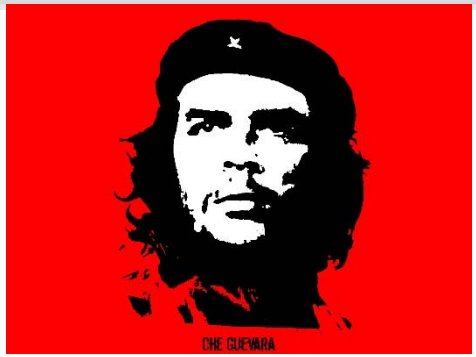
The first term (*Negative Energy*) is local - it decomposes to computations local to the graphical structure.

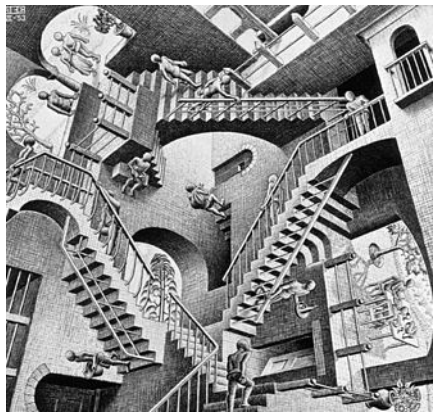
The second term (*Entropy*) is more problematic: not local.

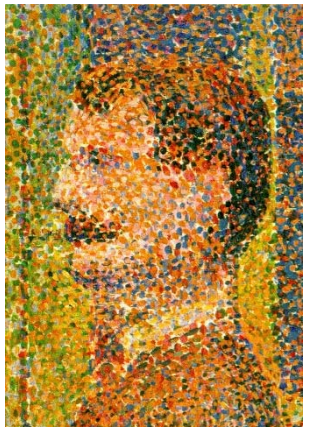
Break

Approximations

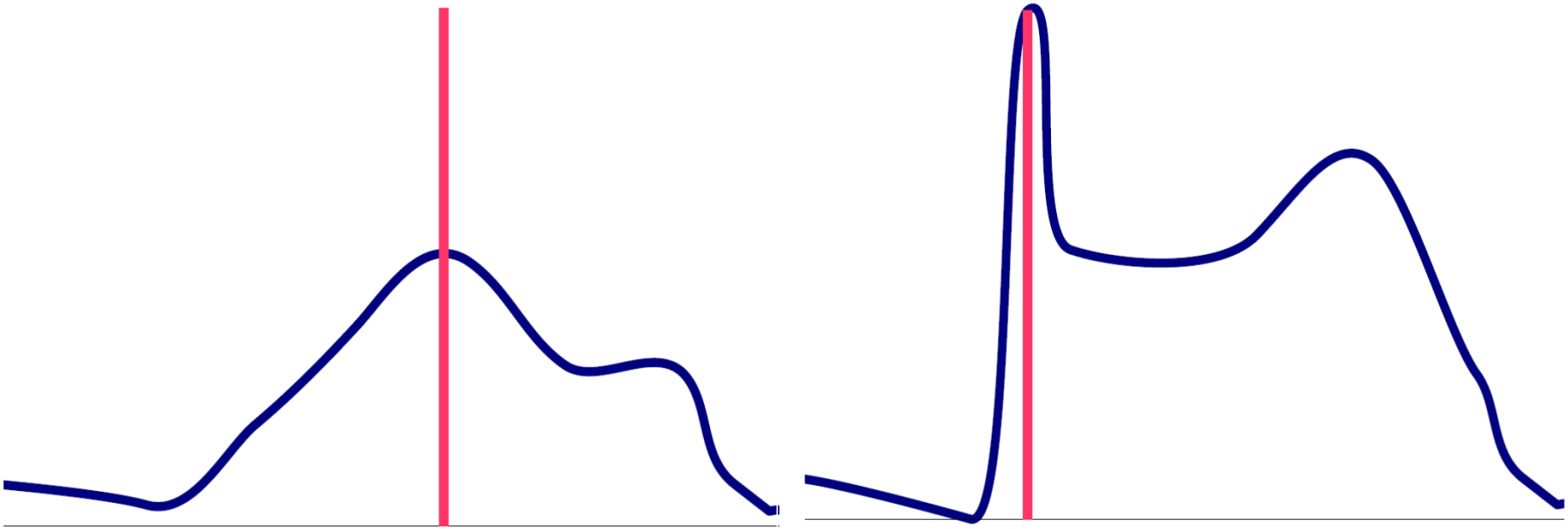
- Constrain Q to a family and optimize
  - Delta function (last lecture)
  - More general form. E.g. factorized distribution
- Unconstrain Q to impose only local consistency.
  - Loopy belief propagation
- Sample to obtain Q that is a mixture of points.
- Combine these methods.







Beyond Deltas



- Maximum posterior can fail to capture mass of distribution.

Variational Approx: Constrain Q

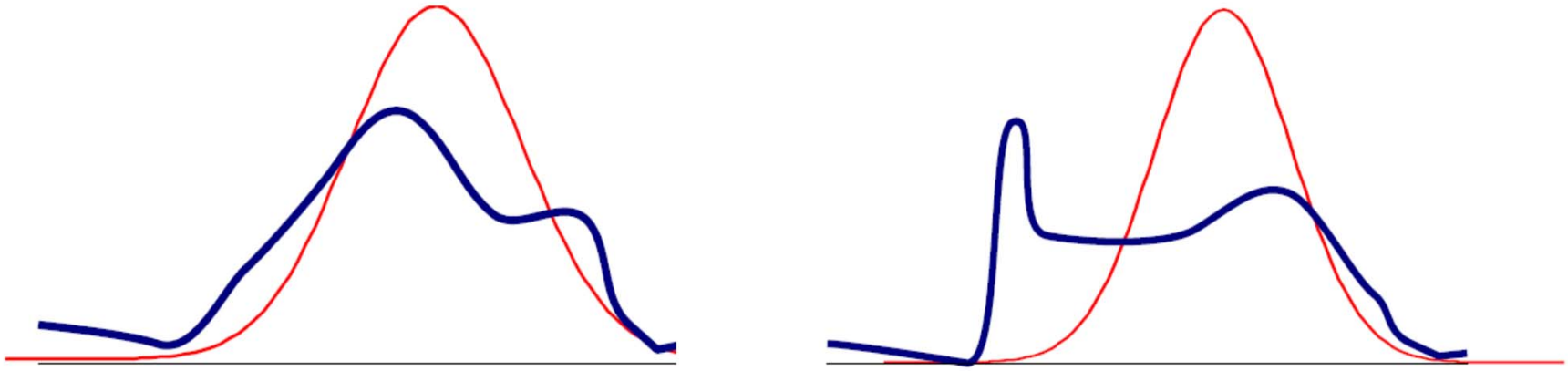
$$\log P(\mathbf{v}) = \max_Q \left[\sum_i \sum_{\mathbf{h}_{C_i}} Q(\mathbf{h}_{C_i}) \phi_i(\mathbf{v}_{D_i}, \mathbf{h}_{C_i}) - \sum_{\mathbf{h}} Q(\mathbf{h}) \log Q(\mathbf{h}) \right] - \log Z.$$

Example: Make Q to be factorised: $Q(\mathbf{h}) = \prod_j Q(h_j)$. Then both terms are local. Optimize parameters of Q:

$$\log P(\mathbf{v}) \geq \max_Q \left[\sum_i \sum_{\mathbf{h}_{C_i}} \left(\prod_{j \in C_i} Q(h_j) \right) \phi_i(\mathbf{v}_{D_i}, \mathbf{h}_{C_i}) - \sum_j \sum_{h_j} Q(h_j) \log Q(h_j) \right] - \log Z.$$

Now entropy is local too. Provides a lower bound to $\log P(\mathbf{v})$. Negative of term in square brackets is the variational free energy.

Picture



- Fits to best approximate the probability mass
- KL(Q||P): does not put mass where there is none
- Constraints on distributions for which variational approximation can be used.

Relax Global Distribution

$$\log P(\mathbf{v}) = \max_{Q} \left[\sum_i \sum_{\mathbf{h}_{C_i}} Q(\mathbf{h}_{C_i}) \phi_i(\mathbf{v}_{D_i}, \mathbf{h}_{C_i}) - \sum_{\mathbf{h}} Q(\mathbf{h}) \log Q(\mathbf{h}) \right] - \log Z.$$

Example: Make Q to be a set of local distributions Q_i.

$$\log P(\mathbf{v}) \approx \max_{\{Q_i, q_j\}} \left[\sum_i \sum_{\mathbf{h}_{C_i}} Q_i(\mathbf{h}_{C_i}) \phi_i(\mathbf{v}_{D_i}, \mathbf{h}_{C_i}) - \sum_i \sum_{\mathbf{h}_{C_i}} Q_i(\mathbf{h}_{C_i}) \log Q_i(\mathbf{h}_{C_i}) + \sum_j (d_j - 1) q_j(h_j) \log q_j(h_j) \right] - \log Z.$$

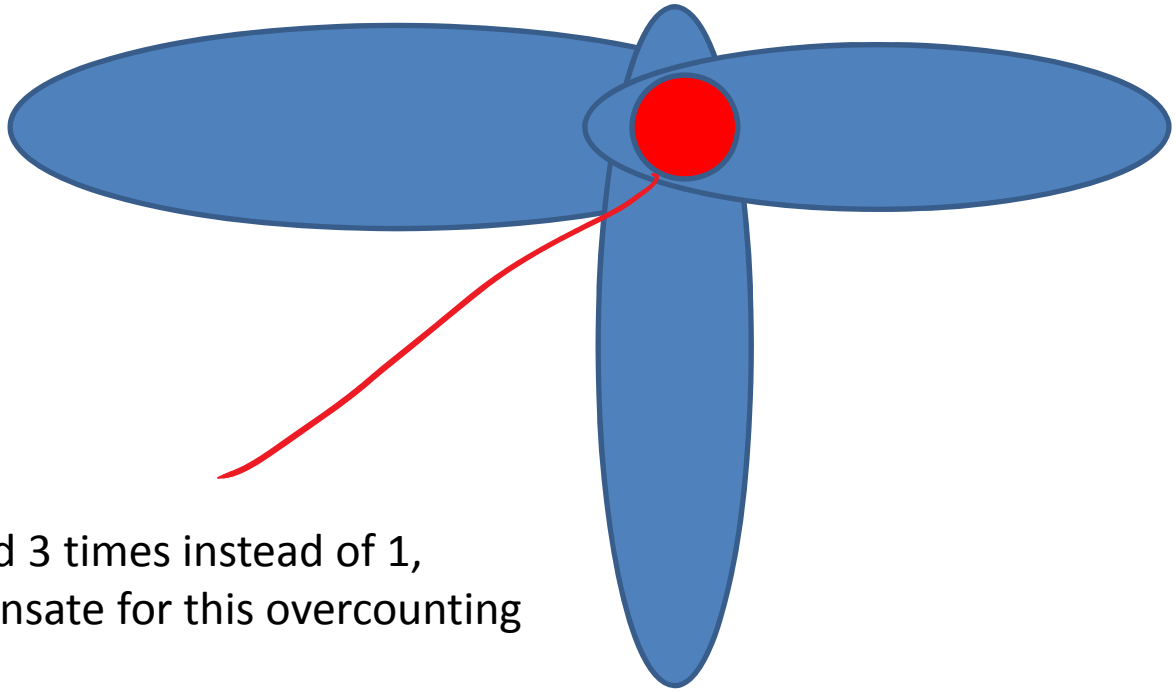
where the maximization is subject to consistency: marginals q_j match marginals of all Q_i: i.e. marginalising out all h_k, $k \neq j$ from any Q_i containing h_j gives the same distribution.

Here, d_j is the *counting number*: the number of neighbours to i in the graph.

The negative of the term in square brackets is called the Bethe Free Energy.

Note locally consistency (with marginals) but not global consistency (with some global distribution). Probabilistic equivalent of an Escher painting.

Picture



This node counted 3 times instead of 1,
so have to compensate for this overcounting
of the marginal.

Result

- Belief propagation in loopy graphs.
  - If it converges…
  - …it converges to a local minimum of the Bethe Free Energy (Yedidia et al 2000).
- Running belief propagation in loopy graphs does approximate inference.
- Belief propagation only suitable for certain distributions (e.g. discrete, Gaussian), where messages stay tractable.
- Don’t forget: methods for inference are also methods for learning.

Expectation Propagation

- What if messages are not tractable?
- Project *locally* to a tractable family.
- Idea behind Expectation Propagation.
- Minka 2001.

Our Journey

Graphical Models

Learning Probabilistic Models

- Introduction to Learning
- Learning exponential family models and Bayesian Set example.
- Approximate Learning and Maximum Posterior/Likelihood
- More Approximate Methods with Free Energies
- Sampling, and hybrid methods: stochastic optimization and stochastic variational methods.