Probabilistic learning and Boltzmann machines

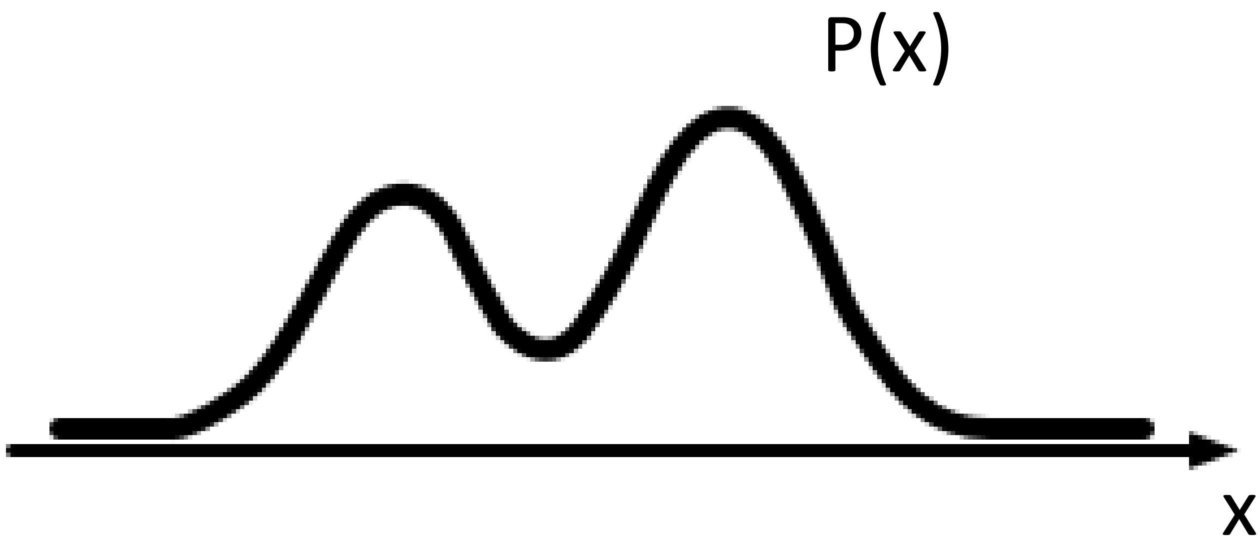
Pietro Berkes, Brandeis University

Probabilistic learning

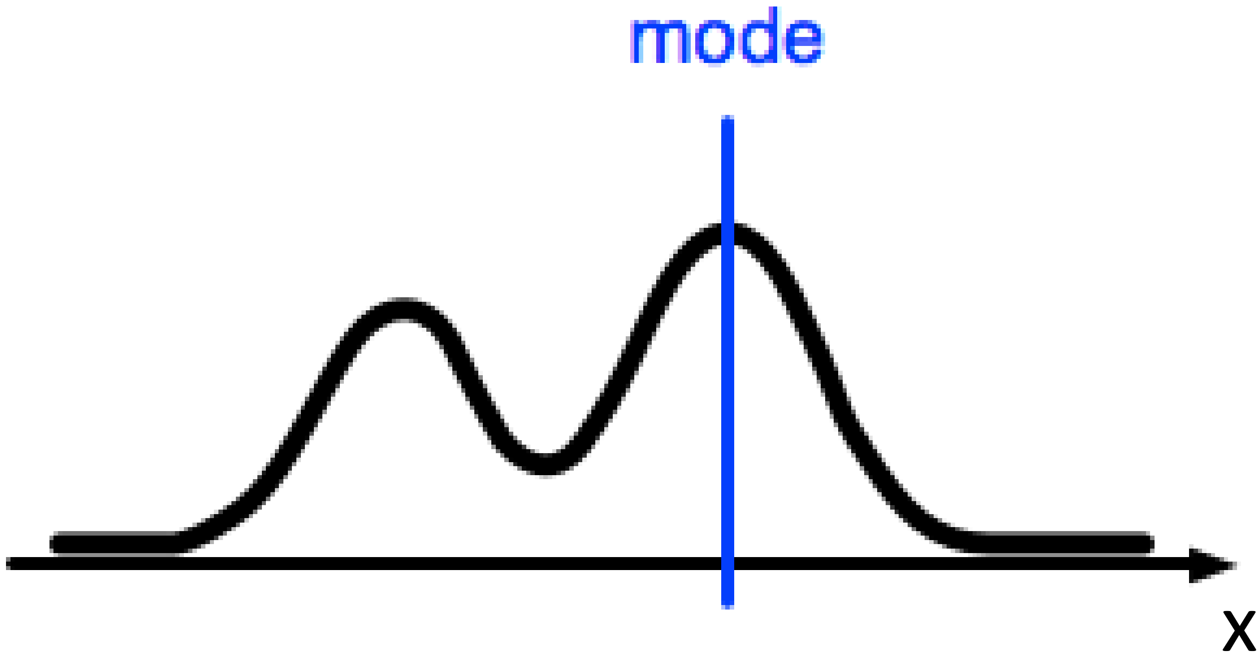
- Problems with classical learning:
  - No uncertainty (consequences for learning)
  - Overfitting: also a consequence of determinism

- Probabilistic approach (aka statistical approach): keep a probability distribution over outcomes, and ideally also over parameters

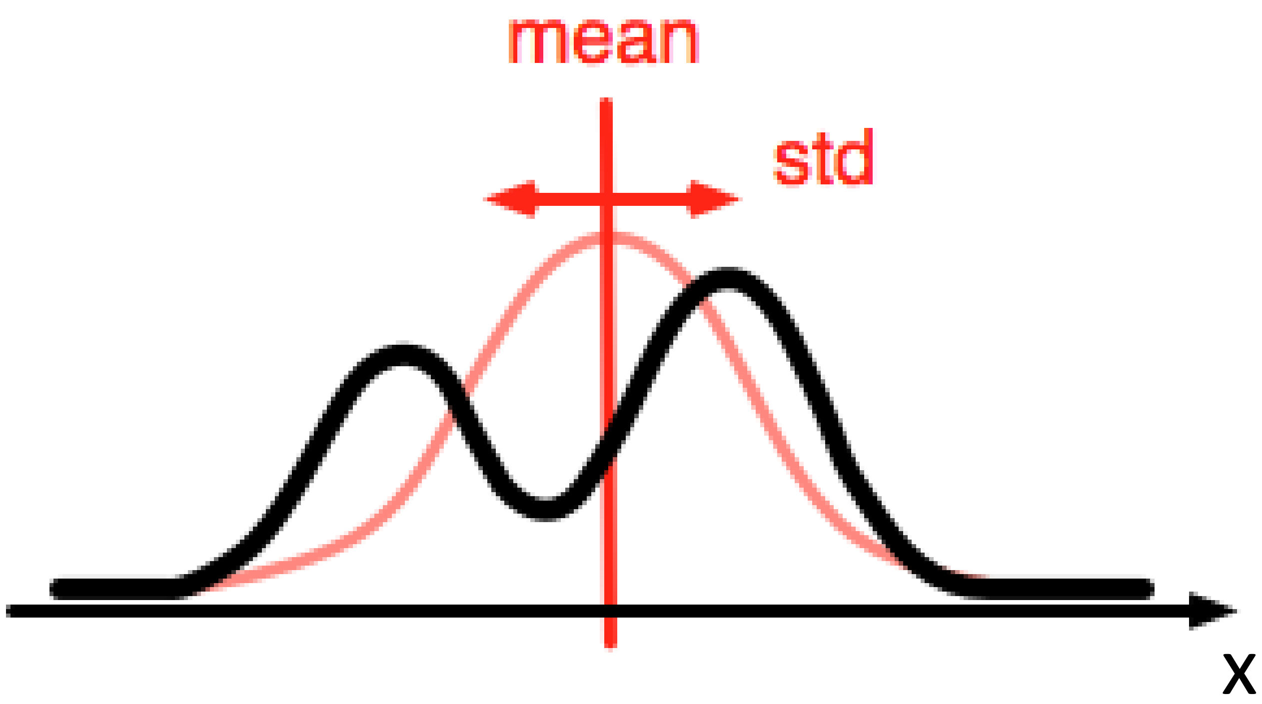
Representing probability distributions



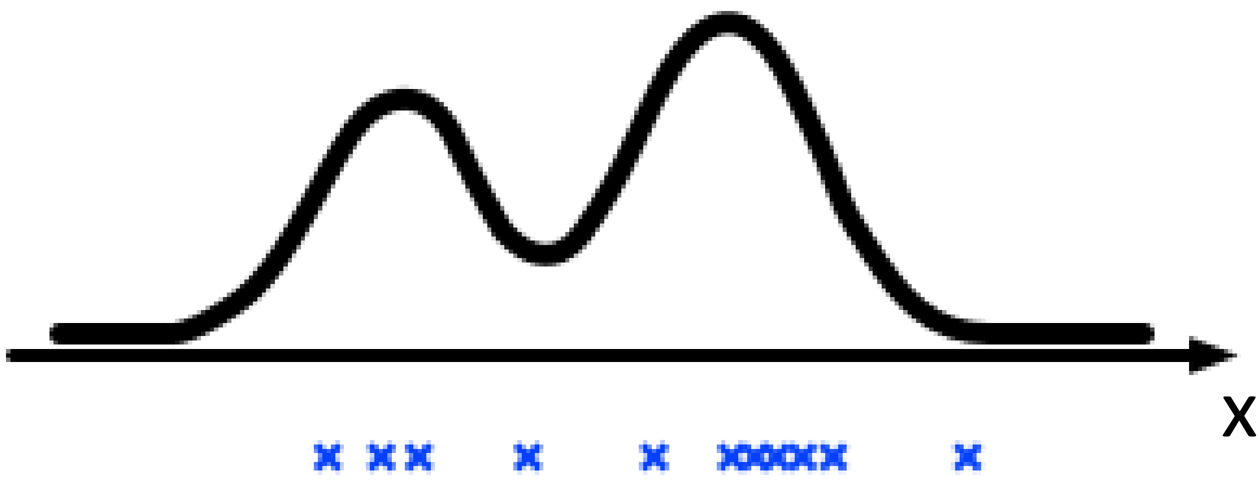
Maximum A Posteriori (MAP)



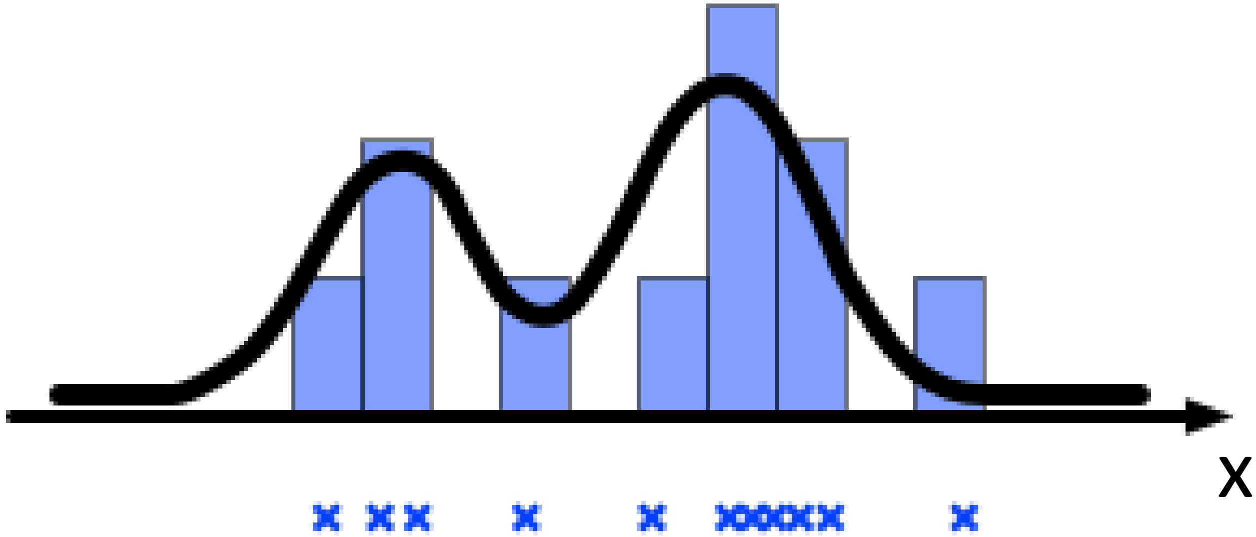
Laplace approximation



Sampling



Sampling



Representing probability distributions

- Parametric
  - The whole distribution is represented at once by a small number of parameters
  - Learning and inference are a complex function of the parameters

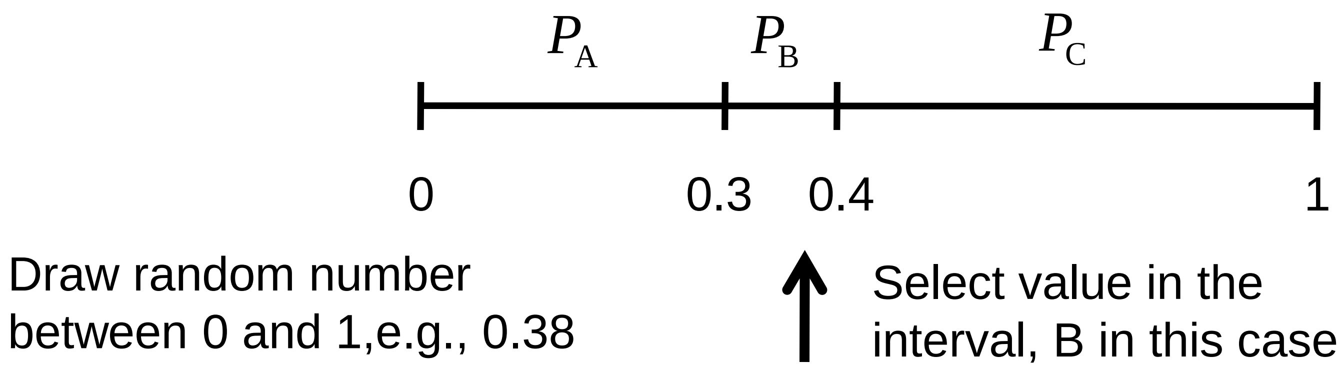
- Sampling
  - The distribution is represented only implicitly
  - Need to collect a number of samples to get an idea of mean and uncertainty
  - Learning is easy, easily allows to compute complex functions of the variables

How to sample

- Simple distributions (1D): use random number generator
- Complex distributions (high-dim): MCMC
  - Iteratively construct new samples from old ones
  - There are **many** different strategies
  - If they satisfy certain basic conditions, it is guarantees that if you collect enough samples you'll get a representation of the whole joint distribution, no matter how complex
- Important for today's class:
  - Sampling from discrete distributions
  - Gibbs sampling

Sampling from discrete distributions

- For example:
  x = A, B, or C with probability
  P_A=0.3, P_B=0.1, and P_C=0.6

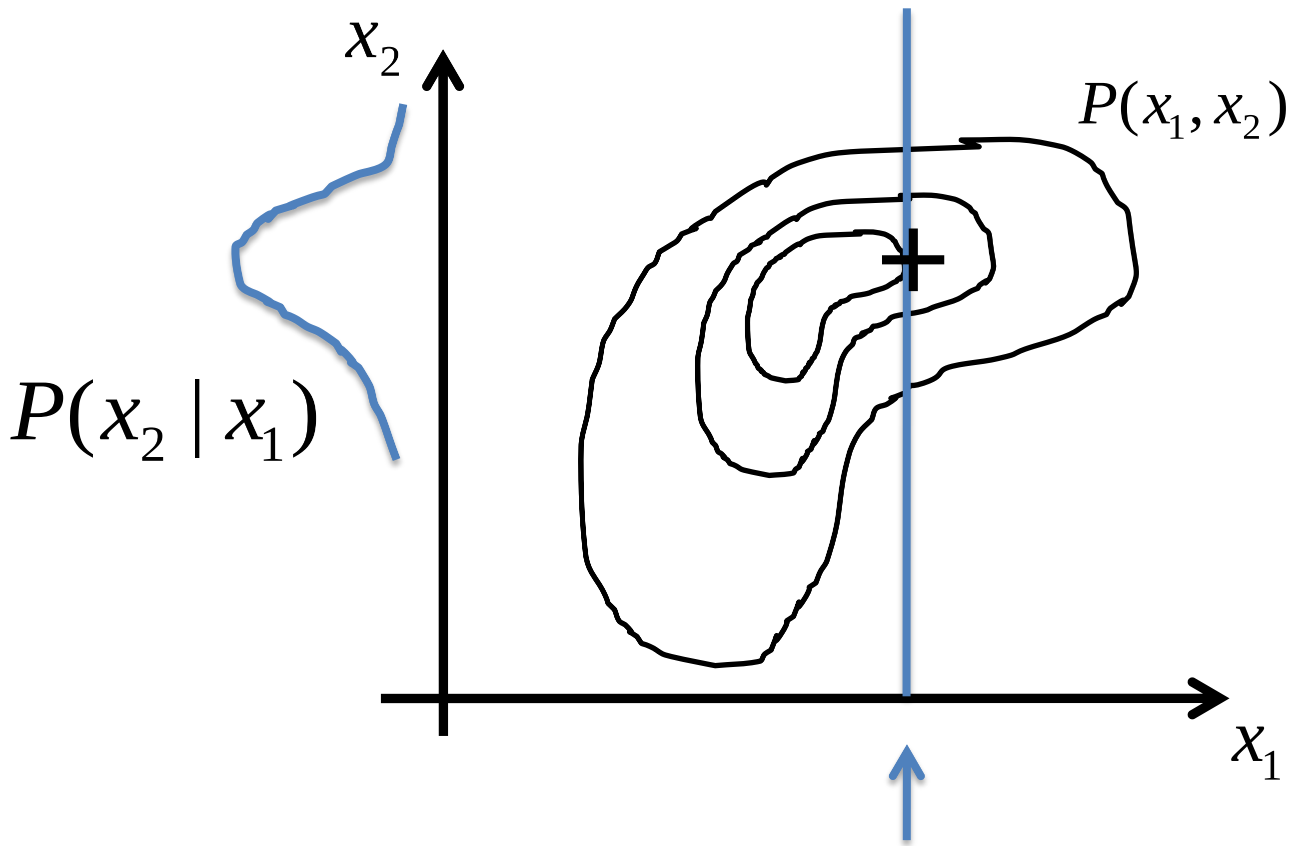


Gibbs sampling

- Goal: sample from $P(x_1, x_2, \ldots, x_N)$
- Idea: sample one variable at the time given the state of the others: $P(x_i \mid x_j, i \neq j)$

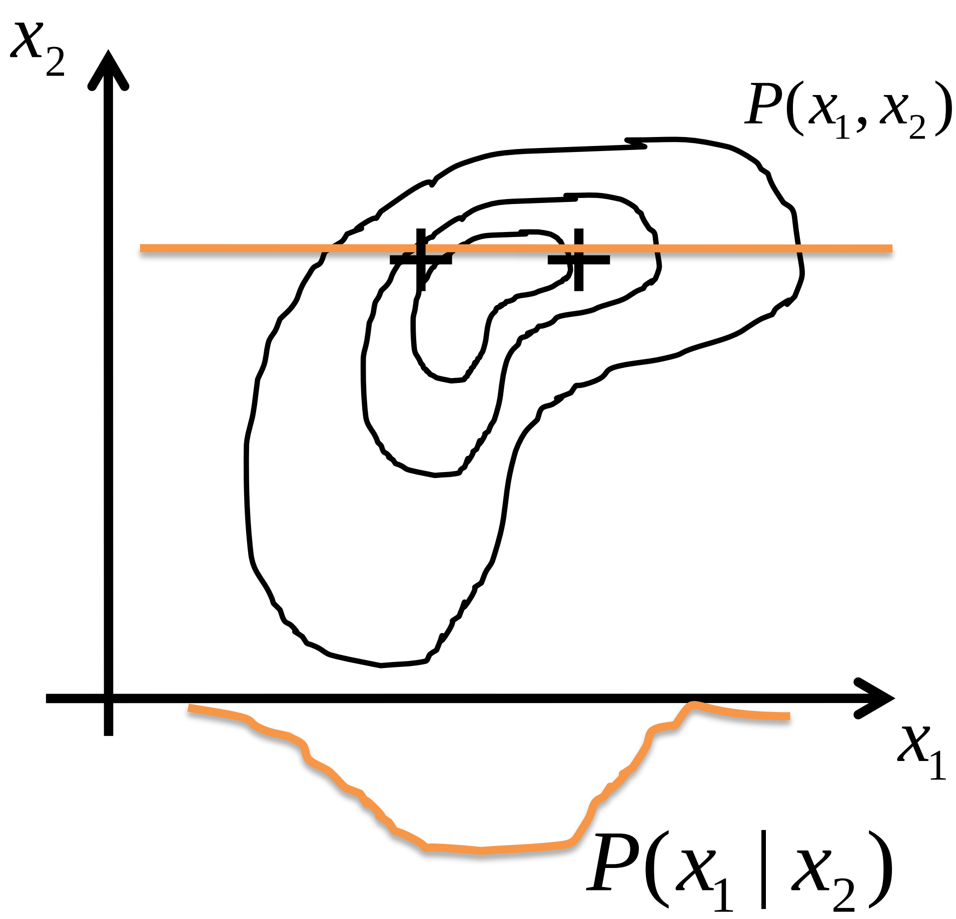
Gibbs sampling

Example: sample from $P(x_1, x_2)$



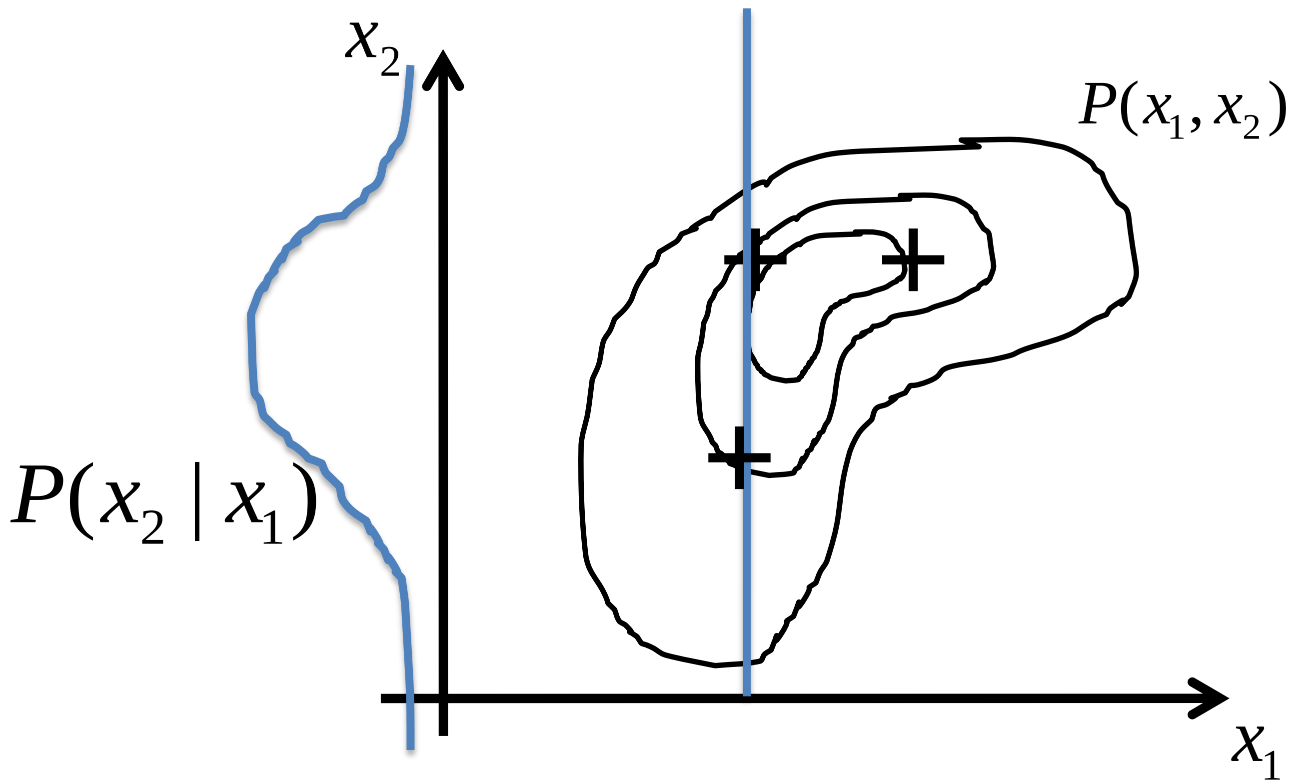
Gibbs sampling

Example: sample from $P(x_1, x_2)$



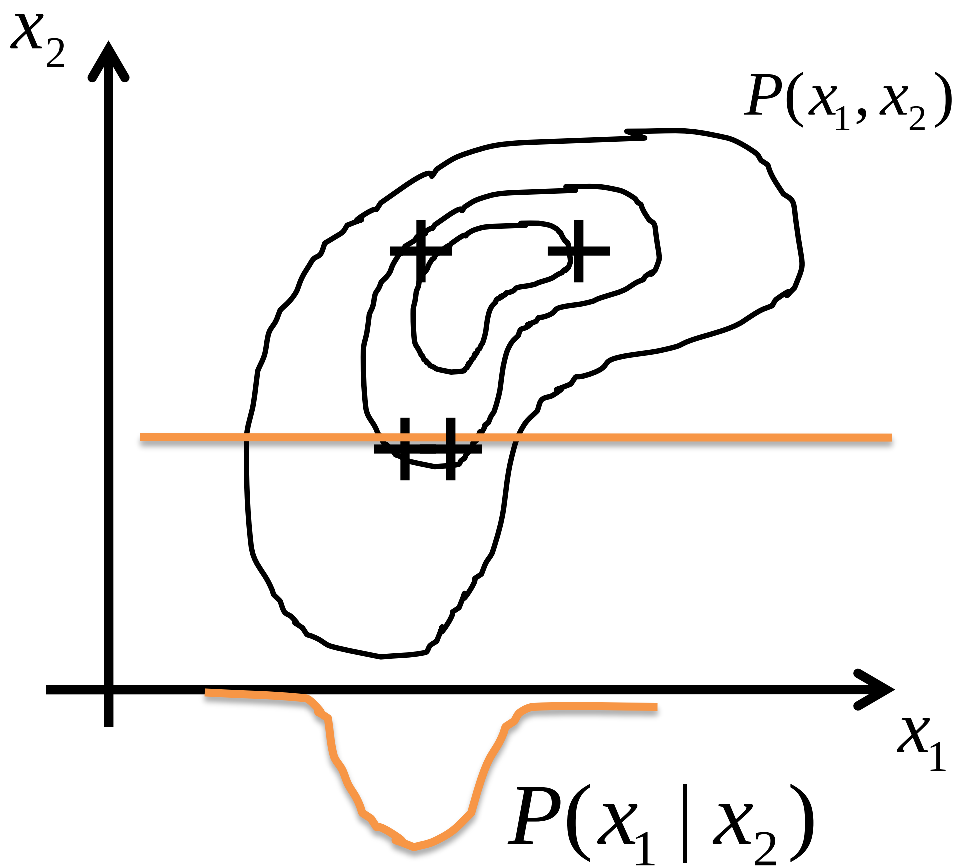
Gibbs sampling

Example: sample from $P(x_1, x_2)$



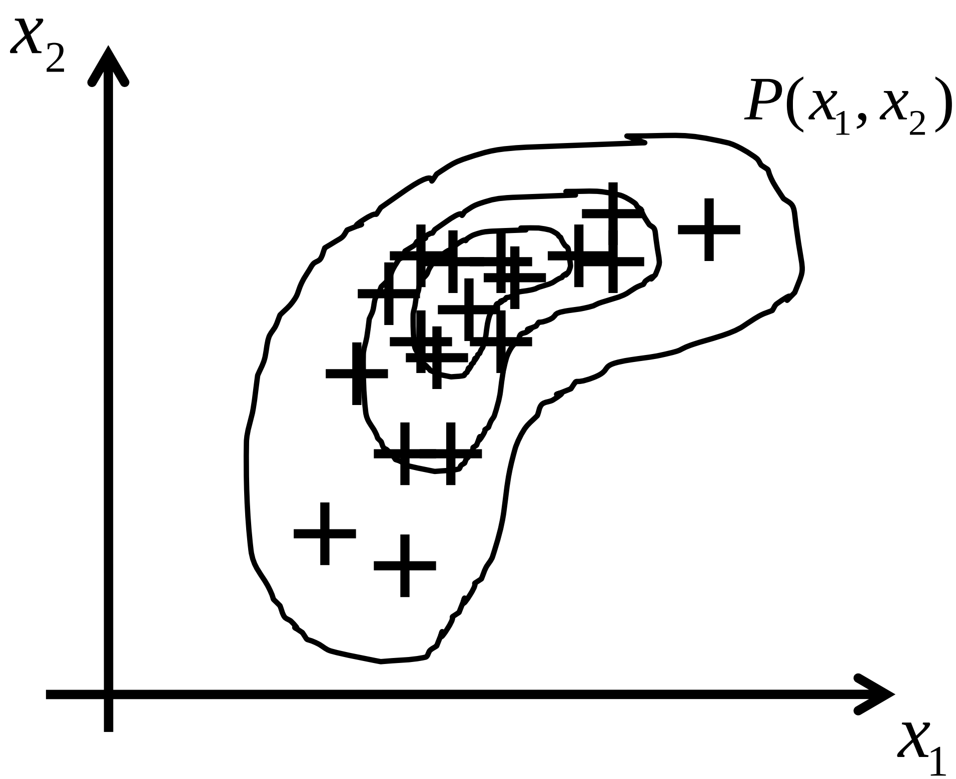
Gibbs sampling

Example: sample from $P(x_1, x_2)$



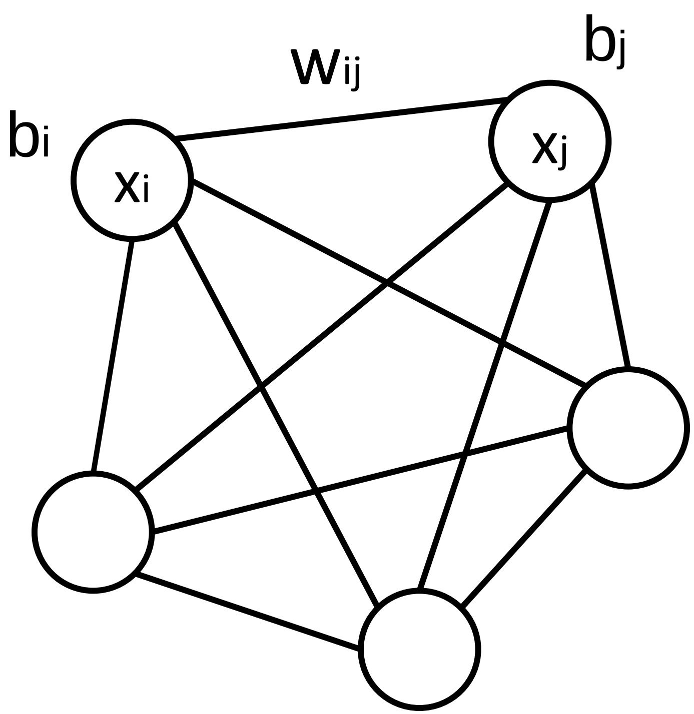
Gibbs sampling

Example: sample from $P(x_1, x_2)$



Boltzmann machines

- The probabilistic equivalent of the Hopfield network (1983)



- x_i = 0 or 1
  (binary neural activity)

- $w_{ii} = 0$ (no self-connections)
- $w_{ij} = w_{ji}$ (symmetric, bidirectional connections)

- b_i: bias term (threshold)

Activity rule

- For each neuron *i*:

  1) compute activation $$a_i = \sum_j w_{ij} x_j$$

Activity rule

- For each neuron *i*:

1) compute activation $$a_i = b_i + \sum_j w_{ij} x_j$$

2) update state of neuron as

Hopfield

$$x_i = \begin{cases} 1 & a_i \geq 0 \\ 0 & a_i < 0 \end{cases}$$

Boltzmann

$$P(x_i = 1 | x_j) = \frac{1}{1 + e^{-a_i}}$$

[plots]

Interpretation as sampling

- This activity rule has the same form of the Gibbs sampling equations

Boltzmann

$$P(x_i = 1 \mid x_j) = \frac{1}{1+e^{-a_i}}$$

- The Boltzmann machine is sampling from the joint distribution

$$P(\mathbf{x}) = \frac{1}{Z}\exp\left(-\sum_i b_i x_i - \sum_{i<j} w_{ij} x_i x_j\right)$$

- Refines what we understand with “model of the data: it models the probability distribution over possible activity patterns in the network

Learning

- We'd like to adapt the model parameters such that the probability distribution captured by the model, $P_{model}(x)$, is as close as possible to the distribution of the observed data, $P_{data}(x)$
- … a few equations later:

$$\Delta w_{ij} = \eta \cdot \left(\left\langle x_i x_j \right\rangle_{\mathrm{data}} - \left\langle x_i x_j \right\rangle_{\mathrm{model}} \right)$$

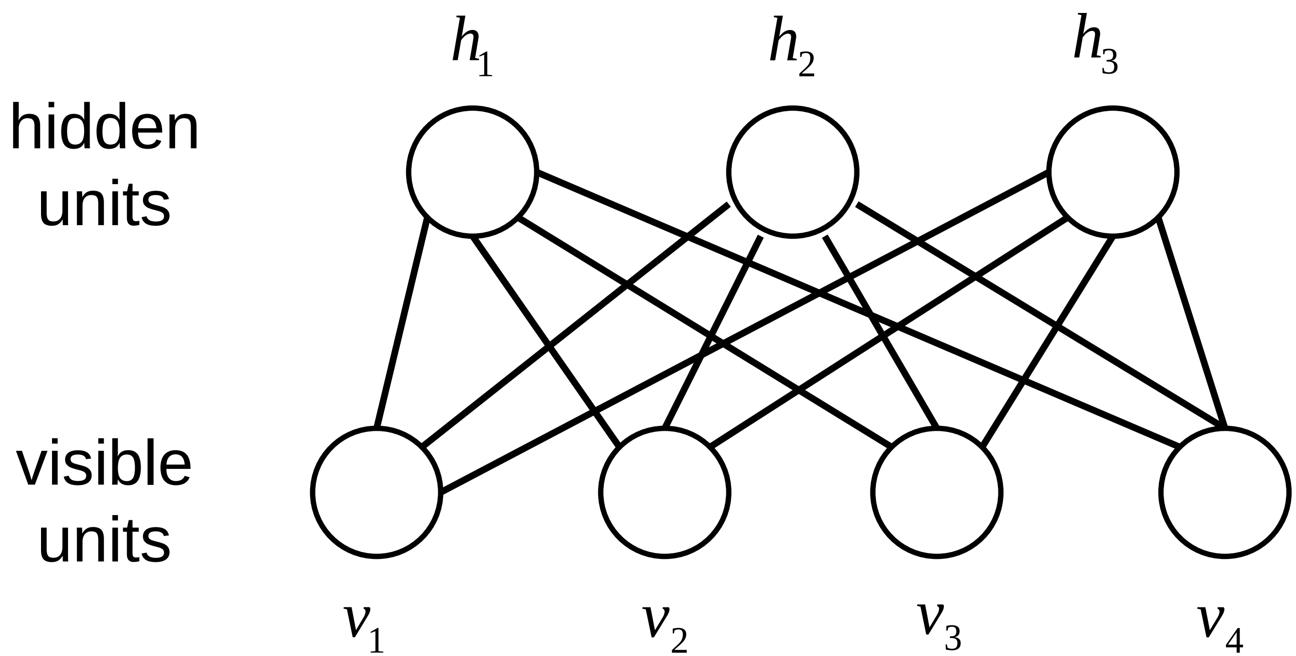
- The Boltzmann machine is matching the second-order correlations of the data distribution

Hidden units

- The relation between observed states may be due to the state of other causes, which are not observed (for example, edges in images)
- We can easily include some “hidden” units in the Boltzmann machine that correspond to these un-observed causes
- The learning rules do not change

Restricted Boltzmann Machine (RBM)

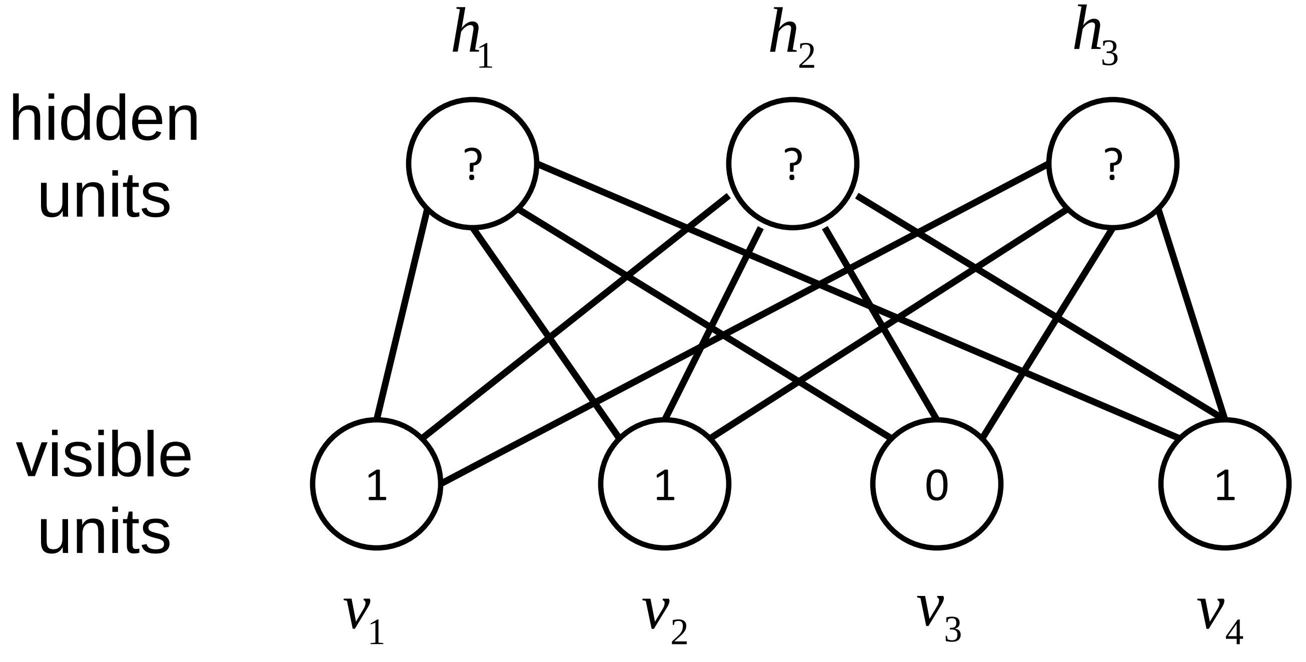
- Neurons are split in two groups: visible and hidden units
- Connectivity is only between the two groups



Inference

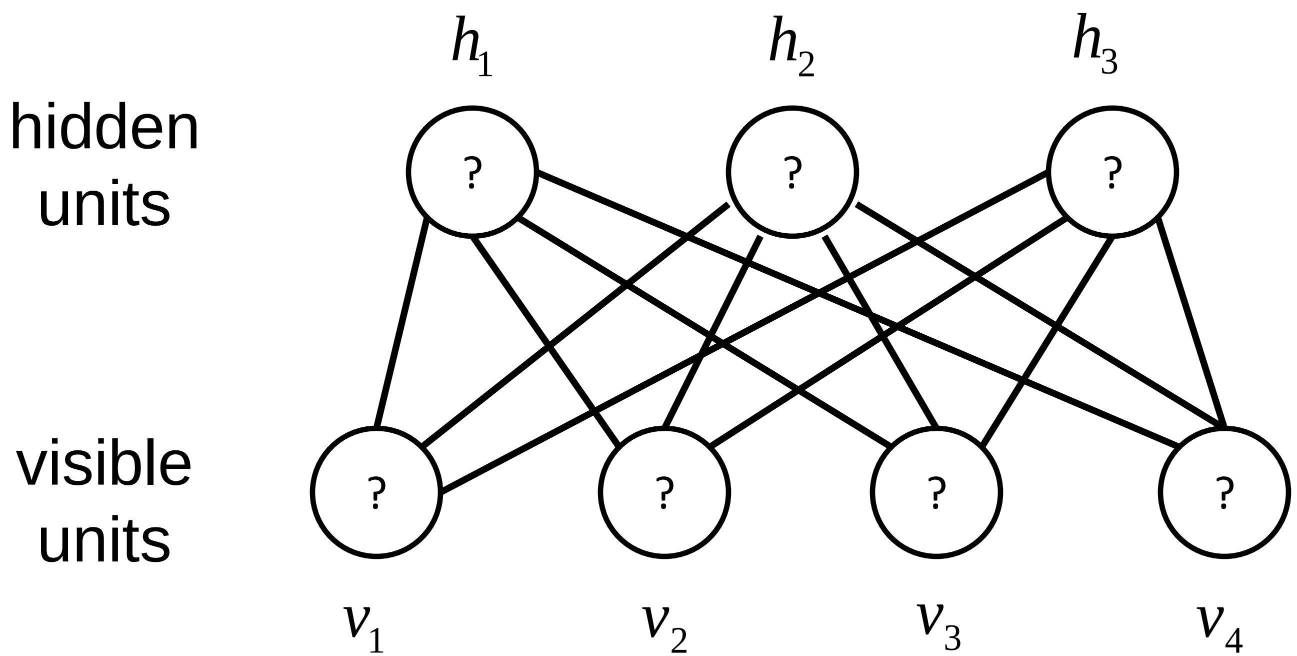
- Given an input activity pattern, we can infer the state of the hidden causes

$$P(h_i = 1 | v_j) = \frac{1}{1 + e^{-b_i - w_{ij} v_j}}$$



Generating data

- Alternate sampling hidden and visible units
- Units in one layer are independent given a pattern in the other layer => sampling is very efficient



Hands-on!

- We’ll use RBMs to learn a model of handwritten letters