Tutorial 3

Graphical Models

How graphs help us describe probabilities, fit models, and make predictions

CSC412 / CSC2506

What we will work out

A **graphical model** uses a graph to describe the structure of a probability distribution. Nodes represent random variables.

1. How a missing connection relates to conditional independence.
2. How to fit a Gaussian model to observed data.
3. How products of weights lead to exponential families.
4. How to sum out variables without building the full probability table.
5. How hidden variables help a restricted Boltzmann machine model images.

Notation. X is a random variable and x is one of its possible values. A vector x lists values for several variables.

Conditional independence on a chain

Consider three discrete random variables with an undirected graph (its edges have no arrows):

$$X - Y - Z.$$

Assume $p(x, y, z) > 0$ for every choice of values.

Conditional independence means that once Y is known, learning Z does not change the probabilities for X:

$$\boxed{X \perp Z \mid Y \qquad \text{means} \qquad p(x \mid y, z) = p(x \mid y).}$$

The graph expresses this requirement: observing the middle node blocks the only path between the two ends.

The next slides show why this is equivalent to writing the joint probability as a product of two weights.

A distribution built from two weights

Assign a positive weight to each pair of values on an edge:

$$\psi_{XY}(x, y) > 0, \qquad \psi_{YZ}(y, z) > 0.$$

These functions are called **potentials**. Larger weights favor that combination of values. A potential need not sum to one.

$$\boxed{p(x, y, z) = \frac{\psi_{XY}(x, y)\psi_{YZ}(y, z)}{C},} \qquad C = \sum_{x,y,z} \psi_{XY}(x, y)\psi_{YZ}(y, z).$$

Dividing by the total weight C makes the probabilities sum to one. We call C a **normalizing constant**.

Writing a distribution as a product of smaller functions is a **factorization**. Here each factor involves only the variables at the ends of one edge.

Week 3 lecture, slides 7–8; tutorial, p. 1

Conditional independence gives a product

Assume $X \perp Z \mid Y$, so $p(x \mid y, z) = p(x \mid y)$.

The probability chain rule gives

$$\begin{aligned} p(x, y, z) &= p(x \mid y, z)p(y, z) \\ &= p(x \mid y)p(y, z). \end{aligned}$$

We can therefore choose

$$\psi_{XY}(x, y) = p(x \mid y), \qquad \psi_{YZ}(y, z) = p(y, z).$$

For this particular choice, $C = 1$, because the product already sums to one.

No factor needs to contain both X and Z.

The product gives conditional independence

Now assume $p(x, y, z) = C^{-1}\psi_{XY}(x, y)\psi_{YZ}(y, z)$. Then

$$\begin{aligned} p(x \mid y, z) &= \frac{p(x, y, z)}{\sum_{x'} p(x', y, z)} \\ &= \frac{\psi_{XY}(x, y)\psi_{YZ}(y, z)}{\sum_{x'} \psi_{XY}(x', y)\psi_{YZ}(y, z)} \\ &= \frac{\psi_{XY}(x, y)}{\sum_{x'} \psi_{XY}(x', y)}. \end{aligned}$$

The factor $\psi_{YZ}(y, z)$ does not depend on the summation variable x', so it cancels. The same happens to C.

The result does not depend on z. This proves $X \perp Z \mid Y$.

Hammersley–Clifford theorem

A **clique** is a set of nodes with an edge between every pair. It is **maximal** if no further node can be added while keeping this property.

Let G be an undirected graph and $\mathcal{C}$ its maximal cliques. For a discrete distribution with $p(x) > 0$ for every x, the following are equivalent:

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C) \quad \text{for some positive potentials } \psi_C$$

$$\Updownarrow$$

$$X_A \perp X_B \mid X_S \quad \text{whenever } S \text{ separates } A \text{ and } B \text{ in } G,$$

$$\text{for every choice of disjoint node sets } A, B, S.$$

Here X_A denotes the variables in A, x_C lists values in C, and $Z = \sum_x \prod_{C \in \mathcal{C}} \psi_C(x_C)$.

S **separates** A and B when every path from A to B passes through S.

This is a **Markov random field**. Edges allow dependence but do not require it.

Week 3 lecture, slides 6–8

Gaussian data and the precision matrix

Suppose n independent observations $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ come from the same multivariate Gaussian distribution:

$$X \sim \mathcal{N}(0, \Sigma).$$

The mean is known to be zero. The **covariance matrix** Σ records variances on its diagonal and how pairs of variables vary together elsewhere.

Assume Σ is **positive definite**: $u^\top \Sigma u > 0$ for every nonzero vector u. This means every direction has positive variance.

$$\boxed{K = \Sigma^{-1}}$$

The inverse covariance K is called the **precision matrix**. For a Gaussian, its zero entries tell us which pairs are independent given all the other variables.

Week 3 lecture, slides 20–23; tutorial, pp. 1–2

The Gaussian density in terms of K

The Gaussian probability density is

$$p(x) = \frac{1}{(2\pi)^{d/2}(\det \Sigma)^{1/2}} \exp\left(-\frac{1}{2}x^\top \Sigma^{-1} x\right).$$

Here $\det \Sigma$ denotes the determinant. Using $K = \Sigma^{-1}$ and $\det \Sigma = 1/\det K$,

$$p(x; K) = (2\pi)^{-d/2}(\det K)^{1/2} \exp\left(-\frac{1}{2}x^\top K x\right).$$

Taking natural logarithms gives

$$\boxed{\log p(x; K) = -\frac{d}{2}\log(2\pi) + \frac{1}{2}\log\det K - \frac{1}{2}x^\top K x.}$$

The notation $p(x; K)$ makes the model parameter K explicit.

Missing edges in a Gaussian model

Because K is symmetric, the two copies of each off-diagonal term combine:

$$-\frac{1}{2}x^\top Kx = -\frac{1}{2}\sum_i K_{ii}x_i^2 - \sum_{i<j} K_{ij}x_i x_j.$$

Exponentiating turns these sums into factors:

$$p(x;K) \propto \prod_i e^{-\frac{1}{2}K_{ii}x_i^2} \prod_{i<j} e^{-K_{ij}x_i x_j}.$$

The symbol $\propto$ means equality up to a normalizing constant. If $K_{ij} = 0$, its pair factor is 1.

$$\boxed{K_{ij} = 0 \quad \Longleftrightarrow \quad X_i \perp X_j \mid X_{\text{all other variables}}, \quad i \neq j.}$$

This equivalence holds for the Gaussian model with positive definite covariance. Thus a missing edge imposes $K_{ij} = 0$.

Week 3 lecture, slides 22–23

Combining the observations

The **log-likelihood** is the sum of the observed log-densities, viewed as a function of K. We fit K by maximizing it:

$$\ell(K) = \sum_{r=1}^{n} \log p(x^{(r)}; K) = C + \frac{n}{2} \log\det K - \frac{1}{2} \sum_{r=1}^{n} (x^{(r)})^\top K x^{(r)}.$$

Here C contains terms that do not depend on K. Average the matrices $x^{(r)}(x^{(r)})^\top$:

$$S_n = \frac{1}{n} \sum_{r=1}^{n} x^{(r)} (x^{(r)})^\top.$$

The trace $\operatorname{tr}(M)$ is the sum of a matrix's diagonal entries. Since $x^\top K x = \operatorname{tr}(K x x^\top)$,

$$\sum_{r} (x^{(r)})^\top K x^{(r)} = n \operatorname{tr}(K S_n), \qquad \boxed{\ell(K) = C + \frac{n}{2}\big[\log\det K - \operatorname{tr}(K S_n)\big].}$$

We average about the known zero mean, so no sample mean is subtracted. All information needed to fit K is in S_n.

Week 3 tutorial, pp. 1–2

The best fit when every pair is allowed

We use two matrix derivative identities. The **gradient** ∇_K collects the derivatives with respect to the entries of K:

$$\nabla_K \log\det K = K^{-1}, \qquad \nabla_K \operatorname{tr}(KS_n) = S_n.$$

Therefore,

$$\nabla_K \ell(K) = \frac{n}{2}(K^{-1} - S_n).$$

Setting this to zero gives

$$K^{-1} = S_n, \qquad \boxed{\widehat{K} = S_n^{-1}, \quad \widehat{\Sigma} = S_n.}$$

This requires S_n to have an inverse. The log-likelihood curves downward along every line through positive definite matrices, so this is the unique maximum.

If S_n has no inverse: there is no finite maximizing positive definite K in this model.

Fitting a chosen Gaussian graph

Assume S_n is positive definite. Fit the same objective while fixing each missing-edge entry to zero:

$$\max_{\substack{K \succ 0 \\ K_{ij}=0 \text{ for missing edges } \{i,j\}}} \big[\log\det K - \operatorname{tr}(KS_n) \big].$$

The notation $K \succ 0$ means that K is positive definite.

For the chain $X_1 - X_2 - X_3$, the allowed matrix has the form

$$K = \begin{pmatrix} K_{11} & K_{12} & 0 \\ K_{12} & K_{22} & K_{23} \\ 0 & K_{23} & K_{33} \end{pmatrix} \succ 0, \qquad X_1 \perp X_3 \mid X_2.$$

The graph fixes which entries must be zero. We estimate the remaining entries together. Setting entries of S_n^{-1} to zero afterwards generally does not solve this optimization.

Week 3 lecture, slide 23; tutorial likelihood, pp. 1–2

A binary model on three nodes

Return to the chain $X_1 - X_2 - X_3$, now with $X_i \in \{0, 1\}$.

$$p(x_1, x_2, x_3) = \frac{\psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)}{Z}.$$

Each positive potential gives a weight to a pair of values. The normalizing constant Z adds the product over all eight possible triples.

Taking logs changes the product into a sum:

$$\log p(x) = \log \psi_{12}(x_1, x_2) + \log \psi_{23}(x_2, x_3) - \log Z.$$

Each log-potential is a table with four entries. We will express its value as a weighted sum of simple indicator functions.

That will put this model into the standard form called an **exponential family**.

Week 3 lecture, slides 15–16; tutorial, pp. 2–3

The exponential-family form

An exponential family writes the probabilities as

$$p_\theta(x) = b(x)\exp\{\theta^\top T(x) - A(\theta)\}, \qquad A(\theta) = \log \sum_x b(x) e^{\theta^\top T(x)}.$$

- θ contains the **natural parameters**: the coefficients of $T(x)$.
- $T(x)$ is a vector of numbers computed from an observation.
- $b(x)$ is a fixed weight. Our binary model uses $b(x) = 1$.
- $A(\theta)$, the **log-partition function**, is the log of the total weight.

For n independent observations, the log-likelihood is

$$\ell(\theta) = \theta^\top \sum_{r=1}^{n} T(x^{(r)}) - nA(\theta) + \underbrace{\sum_{r=1}^{n} \log b(x^{(r)})}_{\text{does not depend on } \theta}.$$

The sum $\sum_r T(x^{(r)})$ keeps all the data information needed to fit θ. It is a **sufficient statistic**.

Week 3 lecture, slides 15–16

Selecting a table entry with indicators

An **indicator** equals 1 when a condition holds and 0 otherwise. For $a, b \in \{0, 1\}$, define

$$T_{12}^{ab}(x_1, x_2) = \mathbb{1}\{x_1 = a,\ x_2 = b\}, \qquad \theta_{12}^{ab} = \log \psi_{12}(a, b).$$

Order the entries as $00, 01, 10, 11$. The superscript ab labels a pair of values.

$$T_{12}(0,1) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \qquad \theta_{12} = \begin{pmatrix} \log \psi_{12}(0,0) \\ \log \psi_{12}(0,1) \\ \log \psi_{12}(1,0) \\ \log \psi_{12}(1,1) \end{pmatrix}.$$

Exactly one indicator is 1, so the dot product selects the matching entry:

$$\boxed{\theta_{12}^{\top} T_{12}(x_1, x_2) = \log \psi_{12}(x_1, x_2).}$$

Define T_{23} and θ_{23} in the same way for the other edge.

Combining the two pairs

Join the two parameter vectors and the two indicator vectors:

$$\theta = \begin{pmatrix} \theta_{12} \\ \theta_{23} \end{pmatrix}, \qquad T(x) = \begin{pmatrix} T_{12}(x_1, x_2) \\ T_{23}(x_2, x_3) \end{pmatrix}.$$

Then

$$\begin{aligned} \log p_\theta(x) &= \theta_{12}^\top T_{12}(x_1, x_2) + \theta_{23}^\top T_{23}(x_2, x_3) - \log Z(\theta) \\ &= \theta^\top T(x) - A(\theta), \end{aligned}$$

where

$$\boxed{Z(\theta) = \sum_{x \in \{0,1\}^3} e^{\theta^\top T(x)}, \qquad A(\theta) = \log Z(\theta).}$$

A quick check. If all log-potentials are zero, all eight weights equal 1. Thus $Z = 8$ and each triple has probability $1/8$.

Week 3 lecture, slides 15–16

Different parameters, the same probabilities

Exactly one indicator for an edge is active:

$$T_{12}^{00} + T_{12}^{01} + T_{12}^{10} + T_{12}^{11} = 1.$$

Add the same constant c to the four entries of θ_{12}, leaving θ_{23} unchanged. Call the resulting full vector θ'. Then

$$\theta'^{\top} T(x) = \theta^{\top} T(x) + c, \qquad Z(\theta') = e^{c} Z(\theta).$$

Every weight and its total are multiplied by the same amount:

$$\boxed{p_{\theta'}(x) = \frac{e^{\theta^{\top} T(x) + c}}{e^{c} Z(\theta)} = p_{\theta}(x).}$$

This parameter change has no effect on probabilities. The extra freedom makes the representation **non-minimal**, meaning it uses more parameters than needed.

To make a representation minimal, remove every way of changing the parameters without changing the probabilities.

Derivatives of the log-partition give averages

For our model, $A(\theta) = \log \sum_x e^{\theta^\top T(x)}$. Differentiating gives

$$\nabla A(\theta) = \frac{\sum_x T(x) e^{\theta^\top T(x)}}{Z(\theta)} = \sum_x p_\theta(x) T(x) = \boxed{\mathbb{E}_\theta[T(X)]}.$$

Here $\mathbb{E}_\theta$ means an average using the model probabilities p_θ.

An indicator's average is the probability that its condition holds. For example,

$$\frac{\partial A}{\partial \theta_{12}^{01}} = \mathbb{E}_\theta[T_{12}^{01}(X)] = P_\theta(X_1 = 0, X_2 = 1).$$

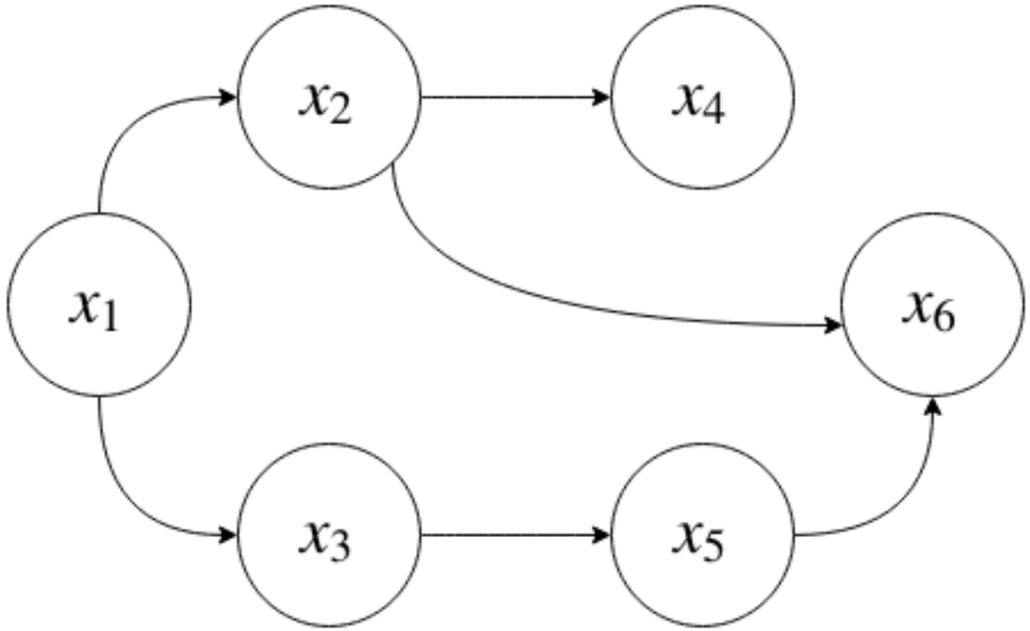
One further derivative gives

$$\nabla^2 A(\theta) = \mathrm{Cov}_\theta(T(X)).$$

The left side is the matrix of second derivatives. The right side measures how the entries of T vary together. These identities hold even when parameters have extra freedom.

Week 3 lecture, slide 16

A directed graph and a probability query



All variables are discrete.
Observed value: $X_6 = \bar{x}_6$, with $p(\bar{x}_6) > 0$.
Wanted: probabilities for X_1 after this observation:

$$p(X_1 \mid \bar{x}_6).$$

We must sum over the unobserved variables X_2, X_3, X_4, X_5.

An arrow into a node identifies a variable used in that node's conditional probability. The joint distribution is

$$p(x_1, \ldots, x_6) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1) \times p(x_4 \mid x_2)p(x_5 \mid x_3)p(x_6 \mid x_2, x_5).$$

Graph from Week 3 tutorial, p. 4

The calculation we need

First compute $g(x_1) = p(x_1, \bar{x}_6)$ for every value of x_1:

$$g(x_1) = p(x_1) \sum_{x_2, x_3, x_4, x_5} p(x_2 \mid x_1) p(x_3 \mid x_1) p(x_4 \mid x_2) \times p(x_5 \mid x_3) p(\bar{x}_6 \mid x_2, x_5).$$

Each factor is a table involving only a few variables. To remove one variable:

1. Multiply all current factors containing that variable.
2. Sum over its values to create a new, smaller table.
3. Save the new table and reuse it in later sums.

This procedure is **variable elimination**. Factors that do not contain the variable stay outside its sum.

We will use the order X_5, X_4, X_3, X_2.

Week 3 lecture, slides 25 and 30; tutorial, pp. 4–5

Four small sums

Each f below names a table we compute once and keep for the next step.

$$f_5(x_2, x_3) = \sum_{x_5} p(x_5 \mid x_3)\, p(\bar{x}_6 \mid x_2, x_5),$$

$$f_4(x_2) = \sum_{x_4} p(x_4 \mid x_2) = 1,$$

$$f_3(x_1, x_2) = \sum_{x_3} p(x_3 \mid x_1)\, f_5(x_2, x_3),$$

$$f_2(x_1) = \sum_{x_2} p(x_2 \mid x_1)\, f_3(x_1, x_2).$$

The factor for X_4 becomes 1, so it drops out of later products.

These intermediate tables do not have to sum to one over their listed variables. At the end, $g(x_1) = p(x_1) f_2(x_1)$.

The final conditional probability

Divide each value of $g(x_1)$ by their total:

$$\boxed{p(x_1 \mid \bar{x}_6) = \frac{p(x_1) f_2(x_1)}{\sum_{x_1'} p(x_1') f_2(x_1')}.}$$

Here $f_2(x_1) = p(\bar{x}_6 \mid x_1)$. This is Bayes' rule, with the probability of the observation computed by elimination.

Why saving tables helps. Suppose each variable has q values. Computing f_5 uses q^2 pairs (x_2, x_3), each with a sum of q terms. That takes work proportional to q^3.

The other steps take at most this much work. Directly listing every assignment with X_6 fixed would involve q^5 assignments.

For other graphs, the largest table we must form depends on the elimination order. These exact probability calculations can still be expensive.

Week 3 lecture, slides 29–35

Learning probabilities for images

Suppose we have N black-and-white images of digits, each 28×28 pixels. Write each image as a vector with $k = 784$ entries:

$$\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}, \qquad x^{(n)} \in \{-1, +1\}^k.$$

We now use $\{-1, +1\}$ in place of the earlier $\{0, 1\}$ encoding:

$$x_i = +1 \text{ means ink}, \qquad x_i = -1 \text{ means background}.$$

Goal. Fit a distribution $p_\theta(x)$ over complete images. Pixels in the same stroke often appear together, so the model should allow them to depend on one another.

$$\max_\theta \sum_{n=1}^{N} \log p_\theta(x^{(n)}).$$

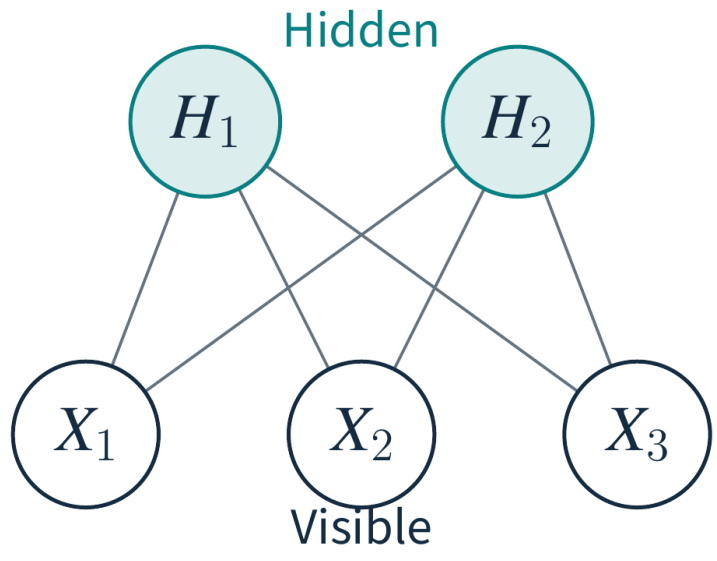
The parameters θ should make the observed images likely. We use only the pixels, without a label saying which digit each image shows.

Image example adapted from Hinton (2010), section 2

Visible and hidden variables

A **restricted Boltzmann machine (RBM)** is an undirected probability model with two groups of variables. Here each variable takes a value in $\{-1, +1\}$.

- **Visible variables** $X_1, \ldots, X_k$ are the observed pixels.
- **Hidden variables** $H_1, \ldots, H_\ell$ are not recorded in the data. They can capture shared patterns among pixels.
- **Restricted** means edges connect the two groups, with no edges within either group.



Adding the joint probabilities over all hidden vectors gives the image probability:

$$p_\theta(x) = \sum_h p_\theta(x, h).$$

Week 3 tutorial, p. 5; diagram retained from the original slides

A hidden variable can link two pixels

Consider $X_1 - H - X_2$, with values in $\{-1, +1\}$ and $\lambda > 0$:

$$p(x_1, x_2, h) \propto e^{\lambda x_1 h} e^{\lambda x_2 h}.$$

With H fixed, the two pixel factors separate. Each pixel tends to have the same sign as H, independently of the other pixel.

With H unobserved, we add its two possible contributions:

$$p(x_1, x_2) \propto e^{\lambda(x_1+x_2)} + e^{-\lambda(x_1+x_2)}.$$

Pixel values	Weight for each pair
$(+1, +1)$ or $(-1, -1)$	$e^{2\lambda} + e^{-2\lambda}$
$(+1, -1)$ or $(-1, +1)$	2

At $\lambda = 1$, the probability of matching pixels is about 0.790. Each pixel is equally likely to be $+1$ or -1, yet their values are dependent.

The RBM probability model

For k visible and ℓ hidden variables, let $\theta = (\alpha, \beta, J)$, where $\alpha \in \mathbb{R}^k$, $\beta \in \mathbb{R}^\ell$, and $J \in \mathbb{R}^{k \times \ell}$.

Assign a positive weight to each pair of vectors:

$$w_\theta(x, h) = \exp\left\{ \sum_i \alpha_i x_i + \sum_j \beta_j h_j + \sum_{i,j} J_{ij} x_i h_j \right\}.$$

The **biases** α_i and β_j favor a variable's $+1$ value when positive and its -1 value when negative. The interaction J_{ij} favors matching signs when positive and opposite signs when negative.

$$\boxed{p_\theta(x, h) = \frac{w_\theta(x, h)}{Z(\theta)},} \qquad Z(\theta) = \sum_{x,h} w_\theta(x, h).$$

The sum in Z covers every visible and hidden vector. It makes the joint probabilities sum to one.

For the next calculations, hold the parameters fixed and write w, p, Z without the subscript θ.

Week 3 tutorial, pp. 5–6

Fixing an image separates the hidden variables

The weight is a product of single-variable factors and edge factors:

$$w(x,h) = \underbrace{\prod_i e^{\alpha_i x_i} \prod_j e^{\beta_j h_j}}_{\text{one variable per factor}} \underbrace{\prod_{i,j} e^{J_{ij} x_i h_j}}_{\text{one edge per factor}}.$$

Now fix the image x. Collect the terms multiplying each h_j:

$$a_j(x) = \beta_j + \sum_i J_{ij} x_i.$$

Then

$$\boxed{w(x,h) = e^{\sum_i \alpha_i x_i} \prod_j e^{h_j a_j(x)}.}$$

The first factor is constant while h varies. Each remaining factor involves only one hidden variable. No term contains two different hidden variables.

Week 3 lecture, slide 19; tutorial, pp. 5–6

Adding up the hidden states

For a fixed image, define its total weight by adding over all hidden vectors:

$$W(x) = \sum_h w(x, h).$$

Because each h_j appears in its own factor,

$$\begin{aligned} W(x) &= e^{\sum_i \alpha_i x_i} \sum_h \prod_j e^{h_j a_j(x)} \\ &= e^{\sum_i \alpha_i x_i} \prod_j \sum_{h_j \in \{-1,+1\}} e^{h_j a_j(x)} \\ &= \boxed{e^{\sum_i \alpha_i x_i} \prod_j \left(e^{-a_j(x)} + e^{a_j(x)}\right).} \end{aligned}$$

Expanding the product selects one value for each hidden variable, covering every hidden vector exactly once.

After computing the inputs $a_j(x)$, we need only ℓ two-value sums instead of listing 2^ℓ vectors. A common shorthand is $e^{-a} + e^{a} = 2\cosh(a)$.

Hidden probabilities given an image

Divide each hidden vector's weight by their total $W(x)$. The visible-only factor and Z cancel:

$$p(h \mid x) = \frac{w(x,h)}{W(x)} = \prod_j \underbrace{\frac{e^{h_j a_j(x)}}{e^{-a_j(x)} + e^{a_j(x)}}}_{p(h_j|x)}.$$

Each factor sums to one over its own hidden variable. This proves that the hidden variables are independent given the image.

For the value $+1$,

$$\boxed{P(H_j = +1 \mid x) = \frac{e^{a_j(x)}}{e^{-a_j(x)} + e^{a_j(x)}} = \frac{1}{1 + e^{-2a_j(x)}}.}$$

The factor 2 comes from comparing the exponents $+a_j(x)$ and $-a_j(x)$.

With the usual **sigmoid** function $\sigma(t) = 1/(1 + e^{-t})$, this is $\sigma(2a_j(x))$. The official tutorial puts the factor 2 inside its definition of σ.

Week 3 tutorial, pp. 5–6

Pixel probabilities given hidden values

The calculation works the same way with h fixed. Define the input to pixel i:

$$b_i(h) = \alpha_i + \sum_j J_{ij} h_j.$$

Collect the terms for each pixel and normalize over the visible vectors:

$$w(x, h) = e^{\sum_j \beta_j h_j} \prod_i e^{x_i b_i(h)}, \qquad p(x \mid h) = \prod_i \frac{e^{x_i b_i(h)}}{e^{-b_i(h)} + e^{b_i(h)}}.$$

Thus the pixels are independent given all hidden values, with

$$\boxed{P(X_i = +1 \mid h) = \frac{1}{1 + e^{-2b_i(h)}} = \sigma(2b_i(h)).}$$

Both conditional distributions, $p(h \mid x)$ and $p(x \mid h)$, are easy to compute without knowing Z.

This independence requires the other group to be known. Summing out shared hidden variables can make the pixels dependent, as our two-pixel example showed.

Generating an image

Start with a random visible vector $x^{(0)}$. Repeat these two steps:

1. Draw each hidden variable independently using its probability given the current image.
2. Draw each pixel independently using its probability given the new hidden vector.

$$\underbrace{h^{(t)} \sim p(h \mid x^{(t)})}_{\text{new hidden values}}, \qquad \underbrace{x^{(t+1)} \sim p(x \mid h^{(t)})}_{\text{new pixel values}}.$$

Here $\sim$ means "draw a random value from this distribution." For example, a pixel with probability 0.8 of $+1$ takes $+1$ in about eight out of ten such draws.

This method is called **alternating Gibbs sampling**. After enough steps, the visible vectors approximately follow $p(x)$.

Each step is simple, but reaching the model distribution can take many steps. Nearby samples can also remain similar.

Hinton (2010), section 2

Comparing images and assigning probabilities

The distribution of images alone is

$$p(x) = \frac{W(x)}{Z}, \qquad W(x) = e^{\sum_i \alpha_i x_i} \prod_j \left(e^{-a_j(x)} + e^{a_j(x)}\right).$$

For two images under the **same fixed parameters**, the total Z cancels:

$$\boxed{\frac{p(x)}{p(x')} = \frac{W(x)}{W(x')}.}$$

So we can compare how likely two images are without computing Z.

To assign an actual probability to one image, we also need

$$Z = \sum_{x \in \{-1,+1\}^k} W(x).$$

This displayed sum has 2^k terms. Computing $W(x)$ for one image is easy, but adding it over all images is generally difficult for a large model.

Week 3 tutorial, pp. 5–6

Where the learning rule comes from

Bring back the parameter subscript. For one observed image,

$$\log p_\theta(x) = \log W_\theta(x) - \log Z(\theta).$$

Since $\partial w_\theta(x, h)/\partial J_{ij} = x_i h_j w_\theta(x, h)$, the two derivatives are

$$\frac{\partial \log W_\theta(x)}{\partial J_{ij}} = \frac{\sum_h x_i h_j w_\theta(x, h)}{W_\theta(x)} = x_i \mathbb{E}_\theta[H_j \mid x],$$

$$\frac{\partial \log Z(\theta)}{\partial J_{ij}} = \frac{\sum_{x', h} x'_i h_j w_\theta(x', h)}{Z(\theta)} = \mathbb{E}_\theta[X_i H_j].$$

The first average keeps the observed image fixed and uses $p_\theta(h \mid x)$. The second averages over both images and hidden vectors using $p_\theta(x, h)$.

This is the same idea as in the exponential-family example: differentiating a log of a total weight gives an average.

Hinton (2010), section 2

Learning from observed and generated images

Subtract the two derivatives. For one training image,

$$\frac{\partial \log p_\theta(x)}{\partial J_{ij}} = \underbrace{x_i \mathbb{E}_\theta[H_j \mid x]}_{\text{observed-image term}} - \underbrace{\mathbb{E}_\theta[X_i H_j]}_{\text{model term}}.$$

Sum this expression over the training images to obtain the gradient for the full dataset. Increase J_{ij} when the observed-image term is larger on average.

Observed-image term. Compute it exactly using the hidden probabilities:

$$\mathbb{E}_\theta[H_j \mid x] = (+1)P(H_j = +1 \mid x) + (-1)P(H_j = -1 \mid x) = 2P(H_j = +1 \mid x) - 1.$$

Model term. Approximate it by averaging $X_i H_j$ over pairs generated by the model.

Contrastive divergence starts at training images and runs only a few Gibbs steps. This shortcut gives a **biased** estimate: its average need not equal the true likelihood gradient.

Hinton (2010), sections 2–3

What the examples have in common

- **Undirected graphs:** the graph tells us which groups of variables may appear together in a factor.
- **Gaussians:** zero entries of the inverse covariance describe missing edges. We fit the allowed entries from the observations.
- **Exponential families:** log-potentials become parameters, and indicators select their values. The log of the total weight gives useful averages when differentiated.
- **Variable elimination:** compute small sums once and reuse their tables. The order affects the amount of work.
- **RBMs:** hidden variables can link pixels. Knowing one group makes the variables in the other group independent, even though the complete image distribution can be hard to normalize.

For a new model, first write its factors. Then ask which variables are known and which need to be summed out.

References and reading notes

Course materials

- Week 3 lecture: Markov Random Fields / Exact Inference. Graphs and factors: slides 6–8. Exponential families: 15–16. Gaussian models: 20–23. Elimination: 24–35.
- Week 3 tutorial notes. The five examples used here.
- CSC412-TUT03: the accompanying expanded tutorial notes.

RBM learning

- Geoffrey Hinton (2010). A Practical Guide to Training Restricted Boltzmann Machines, sections 2–3. Its binary encoding differs from the $\{-1, +1\}$ encoding used here.

When reading the official tutorial. On page 3, Z is generally not 1, and log-partition derivative identities still hold when parameters have extra freedom. On page 6, the sigmoid expression gives $P(H_j = +1 \mid x)$. Use the explicit formulas in these slides.