an interesting decomposition

Big problems:

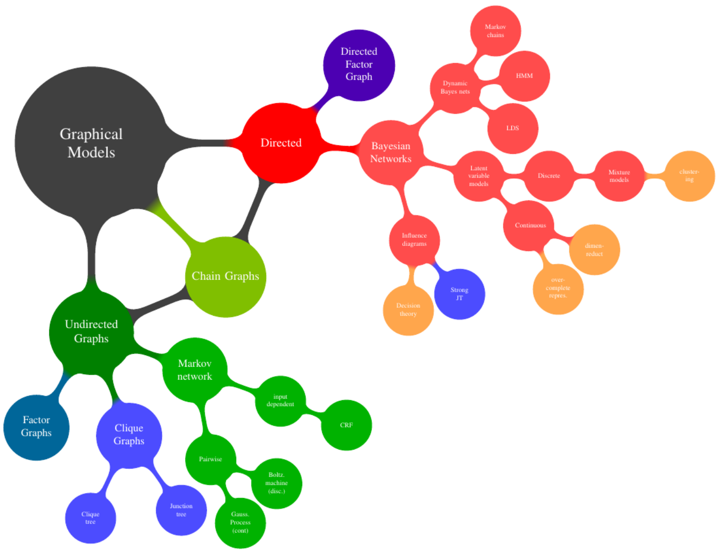
1. **the learning problem**: given a training set $\mathcal{D}$ of vectors $\mathbf{x}$, what is $P(\mathbf{x} \mid \mathcal{D})$? ie. capture a complex joint distribution.
2. **the inference problem**: given a training set $\mathcal{D}$ of vectors $\mathbf{x}$, and knowledge of the values taken by some subset "obs" of the elements in x that are *observed*, what is $P(x_i \mid \texttt{obs}, \mathcal{D})$? ie. find a posterior distribution.

answer: inference and learning are intractable in general, but perhaps could use prior knowledge of conditional independencies (eg. causation)? $\implies$ PGMs

---

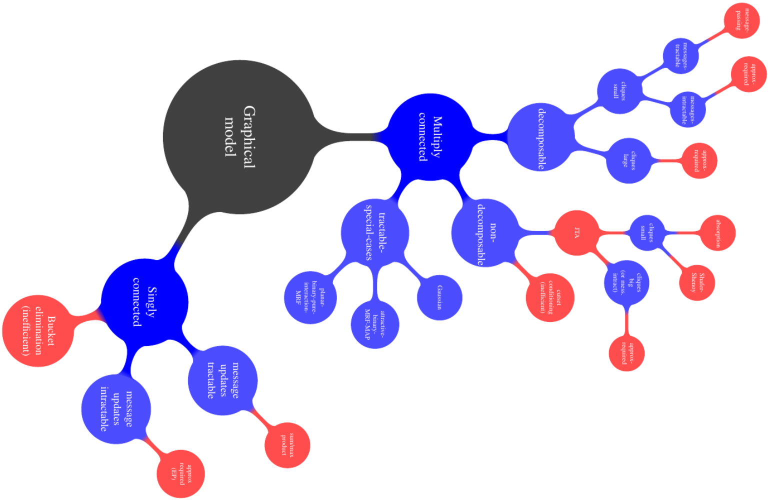
Today we will see that the LEARNING PROBLEM actually depends on a solution to the INFERENCE PROBLEM.

the PGM family



(from David Barber’s amazing, free, book: google “Brml”)
(`family-of-PGMs1.png` in github week9)

the PGM family, by inference method



(from David Barber's amazing, free, book: google "Brml")
(`family-of-PGMs2.png` in github week9)

PGMs

I can think of three possible tasks:

1. infer $p(x|\texttt{obs})$, for any query variable x in the light of any set of observed variables $\texttt{obs}$.
2. infer the *most likely joint state* $p(\mathbf{x}|\texttt{obs})$, for all nodes simultaneously.
3. improve the tables (or learn from scratch) using a data set.

For a discrete-valued PGM...

#1 is solved by the SUM-PRODUCT algorithm, a.k.a. “probability propagation”, “belief propagation”, the “forward-backward algorithm”, and “turbo decoding”. (COMP307).

#2 is solved by the MAX-SUM algorithm, a.k.a. “Viterbi algorithm”. We will mention in the context of HMMs (next week).

#3 is the learning problem. Let’s look at that...

a very general view of learning: max likelihood

- my notation: $\mathbf{x}$ for all variables $(x_0, x_1, ..)$, but $\mathbf{v}$ for those that are "visible" (observed), $\mathbf{h}$ for those remaining hidden. $\mathbf{x} = (\mathbf{v}, \mathbf{h})$.
- we have a **model**, that has **parameters** $\boldsymbol{\theta}$
- For given $\boldsymbol{\theta}$ the model specifies a joint $P(\mathbf{x} \mid \boldsymbol{\theta})$
- **data set** on the visibles: $\mathcal{D} = \{\mathbf{v}_n\}, n = 1 \dots N$

For any given data set, the model gives a *likelihood*, $P(\mathcal{D} \mid \theta)$, often abbreviated as $\mathcal{L}$. This is the chance that our model could make the dataset *if we were to sample repeatedly from the joint*. It's a function of θ so we can view it as a "surface" in θ-space.

The most intuitive approach to learning: find and use the parameter values that maximize the likelihood. ie. those θ for which our data appears most likely. This amounts to finding the highest point on a surface.

unpacking the likelihood

$$\mathcal{L}(\theta) = P(\mathcal{D} \mid \theta)$$
$$= \prod_{\mathbf{v} \in \mathcal{D}} P(\mathbf{v} \mid \theta) \qquad \text{(if data is i.i.d.)}$$

Nb. from now on we'll just drop the "$\mid \boldsymbol{\theta}$" from the r.h.s., as it occurs in everything.

$$\log \mathcal{L} = \sum_{\mathbf{v} \in \mathcal{D}} \log P(\mathbf{v})$$
$$\propto \underbrace{\frac{1}{N} \sum_{\mathbf{v} \in \mathcal{D}} \log P(\mathbf{v})}_{\text{av. log likelihood per pattern}}$$

log likelihood of a dataset of $\mathbf{v}$

Consider a belief net, and a parameter θ of (say) the j^{th} factor (which will be the factor $P(x_j|\text{par}_j)$ in the net. We're going to look at the gradient of $\log \mathcal{L}$ with respect to this parameter.

At the "best" θ, this gradient must be zero, so this is a way of identifying the highest point.

Here is a trick for figuring out this gradient:

1. $\dfrac{\partial}{\partial \theta} \log P = \dfrac{1}{P} \dfrac{\partial P}{\partial \theta}$ and thus (trivially)

2. $\dfrac{\partial P}{\partial \theta} = P \dfrac{\partial}{\partial \theta} \log P$

gradient of $\log \mathcal{L}$ for the whole data set

$$\log \mathcal{L} \quad \propto \quad \underbrace{\frac{1}{N} \sum_{\mathbf{v} \in \mathcal{D}} \log P(\mathbf{v})}_{\text{av. log likelihood per pattern}}$$

and so its gradient must be

$$\frac{\partial}{\partial \theta} \log \mathcal{L} \quad \propto \frac{1}{N} \sum_{\mathbf{v} \in \mathcal{D}} \underbrace{\frac{\partial}{\partial \theta} \log P(\mathbf{v})}_{\text{so what's this?}}$$

gradient of the log likelihood for a single pattern

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\log P(\mathbf{v}) &= \frac{1}{P(\mathbf{v})}\;\frac{\partial}{\partial\theta}P(\mathbf{v}) && \leftarrow \text{via trick 1}\\
&= \frac{1}{P(\mathbf{v})}\;\frac{\partial}{\partial\theta}\sum_{\mathbf{h}}P(\mathbf{v},\mathbf{h}) && \leftarrow \text{sum rule}\\
&= \sum_{\mathbf{h}}\frac{1}{P(\mathbf{v})}\;\frac{\partial}{\partial\theta}P(\mathbf{v},\mathbf{h}) && \leftarrow \text{reordering}\\
&= \sum_{\mathbf{h}}\frac{P(\mathbf{v},\mathbf{h})}{P(\mathbf{v})}\;\frac{\partial}{\partial\theta}\log P(\mathbf{v},\mathbf{h}) && \leftarrow \text{via trick 2}\\
&= \underbrace{\sum_{\mathbf{h}}P(\mathbf{h}\mid\mathbf{v})}_{\text{av. over posterior!}}\;\frac{\partial}{\partial\theta}\log P(\mathbf{x}) && \leftarrow \text{product rule}
\end{aligned}
$$

(back to) gradient of $\log \mathcal{L}$ for the whole data set

Now we can put the whole thing back together:

$$\frac{\partial}{\partial \theta} \log \mathcal{L} \;\; \propto \underbrace{\frac{1}{N} \sum_{\mathbf{v} \in \mathcal{D}}}_{\text{data}} \underbrace{\sum_{\mathbf{h}} P(\mathbf{h} \mid \mathbf{v})}_{\text{av. over posterior}} \;\; \frac{\partial}{\partial \theta} \log P(\mathbf{x})$$

Notice that now it's an average over the the gradient of the joint, $\frac{\partial}{\partial \theta} \log P(\mathbf{x})$, so that's a quantity of crucial importance!

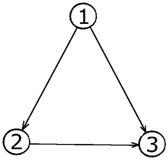
EM algorithm:

- E: infer the posterior (holding θ const.)
- M: take a step up the gradient (holding posterior const.)

eg: Mixtures of Gaussians, but EM applies to any directed PGM.

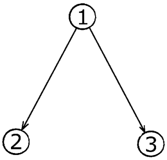
directed PGMs = belief nets = "causal" nets

fully connected



$$x_i \not\perp\!\!\!\perp x_j$$
$$x_i \not\perp\!\!\!\perp x_j \mid x_k$$

naive Bayes

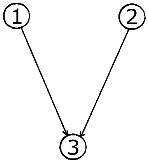


$$x_2 \not\perp\!\!\!\perp x_3$$
$$x_2 \perp\!\!\!\perp x_3 \mid x_1$$

a chain

$$x_1 \not\perp\!\!\!\perp x_3$$
$$x_1 \perp\!\!\!\perp x_3 \mid x_2$$

explaining away



$x_1 \perp\!\!\!\perp x_2,$ and yet
$$x_1 \not\perp\!\!\!\perp x_2 \mid x_3$$

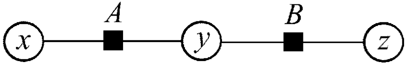
undirected PGMs (a.k.a. Markov random fields)

Graphical models describe joint probability distributions that *factor*. As we've seen, one way a distribution can factor is via application of the product rule to the joint as in, say, $p(x, y, z) = p(x)\ p(y|x)\ p(z|x, y)$, which corresponds to a directed graph called a Belief Net. However other factorisations exist. For example we could have

$$p(x, y, z) = \frac{1}{Z}\ \ \phi_A(x, y)\ \ \phi_B(y, z)$$

where Z is a normalisation factor. The ϕ are usually called "potentials".

Eg. if x, y, x are binary, ϕ_A and ϕ_B are 2x2 tables.



The potentials ϕ need only be positive.

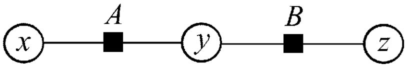
One way to ensure this positivity is to use exponentials of another function: $\phi_A = e^{E_A}$. That way the function E is free to roam over *any* values. Then we have

$$p(x, y, z) \;=\; \frac{1}{Z} e^{E_A(x,y)} e^{E_B(y,z)} \;=\; \frac{1}{Z} e^{E_A(x,y)+E_B(y,z)}.$$

Physicists note: the E are completely analogous to (negative) energies in a physical system with Boltzmann distribution p

---

Note that the potentials ϕ *don't* need to be normalised along either their rows or columns.



Are x and z conditionally independent given y?

$$p(x,z) = \sum_{Y} p(x,y,z) \qquad \propto \sum_{Y} \phi_A(x,y)\ \phi_B(y,z)$$

It seems clear that they won't de-couple if we don't know y. But:

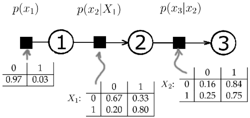
$$\begin{aligned} p(x,z|y) &= \frac{p(x,y,z)}{\sum_x \sum_Z p(x,y,z)} \qquad \propto \ \phi_A(x,y)\, \phi_B(y,z) \\ &= p(x|y)\ p(z|y) \end{aligned}$$

Once we know y, the distribution $p(x,z|y)$ factors.

In an undirected graph, a variable becomes conditionally independent of *all other* variables, given its neighbours.

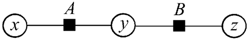
PGM summary

directed



- each factor is normalised
- product of all factors is automatically normalised
- can exhibit “explaining away”
- arrows are suggestive of a “causal” interpretation

undirected



- factors aren’t normalised
- product of all factors is not normalised
- no “causal” interpretation?
- seem to be a superset of directed models in fact...

Graphical models **simplify** the full joint by making **assumptions** about conditional independencies between variables.

a general view of learning generative models

Earlier we looked at EM-like learning in directed graphical models. The steps were:

1. we wrote down the log likelihood,
2. took the gradient
3. discovered this involved adding up little gradients using samples from the data and the posterior over “hidden” nodes.
4. noted that we can use Gibbs Sampling to generate those samples

Now we will extend this to any graphical model, directed or not.

a note on normalised *vs* unnormalised probabilities

Denote probabilities by P, or by $P^\star$ if they are not yet normalised. For example,

$$P(\mathbf{v},\mathbf{h}) = \frac{P^\star(\mathbf{v},\mathbf{h})}{Z} \qquad \text{with} \qquad Z = \sum_{\mathbf{v}} \sum_{\mathbf{h}} P^\star(\mathbf{v},\mathbf{h}) \tag{1}$$

- In *some* cases of interest it is easy to ensure P is normalized (e.g. directed PGMs / belief nets).
- But in many cases it's easy to specify a plausible $P^\star$ yet hard to find Z and know P exactly (e.g. undirected PGMs)

the catch for undirected models

The sum in Z is over all *configurations* (all possible vectors $\mathbf{x}$), so in general it's likely to be intractable.

log likelihood of a dataset of $\mathbf{v}$

$$\begin{aligned}
\log L &= \log P(\mathcal{D}) \\
&= \sum_{\mathbf{v}\in\mathcal{D}} \log P(\mathbf{v}) \\
&= \sum_{\mathbf{v}\in\mathcal{D}} \log \left(P^\star(\mathbf{v})/Z\right) && \leftarrow \text{in terms of } P^\star \\
&= \sum_{\mathbf{v}\in\mathcal{D}} \left(\log P^\star(\mathbf{v}) \quad - \quad \log Z\right) \\
&\propto \underbrace{\frac{1}{N}\sum_{\mathbf{v}\in\mathcal{D}} \log P^\star(\mathbf{v})}_{\text{av. log likelihood per pattern}} \qquad - \qquad \log Z
\end{aligned}$$

Recap: The trick for finding the gradient of this: notice that

1. $\frac{\partial}{\partial w}\log P = (\frac{\partial}{\partial w}P)/P$ and conversely,
2. $\frac{\partial}{\partial w}P = P\frac{\partial}{\partial w}\log P$.

Recap: gradient of the first term (average of $\log P^\star$)

$$\frac{\partial}{\partial w}\left[\frac{1}{N}\sum_{\mathbf{v}\in\mathcal{D}} \log P^\star(\mathbf{v})\right]$$

$$= \sum_{\mathbf{v}\in\mathcal{D}} \frac{1}{P^\star(\mathbf{v})} \frac{\partial}{\partial w} P^\star(\mathbf{v}) \qquad \leftarrow \text{via trick 1}$$

$$\vdots$$

$$= \underbrace{\frac{1}{N}\sum_{\mathbf{v}\in\mathcal{D}}}_{\text{over data}} \underbrace{\sum_{\mathbf{h}} P^\star(\mathbf{h} \mid \mathbf{v})}_{\text{av. over posterior}} \frac{\partial}{\partial w} \log P^\star(\mathbf{x}) \qquad \leftarrow \text{product rule}$$

gradient of the second term ($\log Z$)

The second term is all about the normalisation factor, Z.
(NB. gradient will be *automatically* be zero in any belief net!)

$$\begin{aligned}
\frac{\partial}{\partial w}\log Z &= \frac{1}{Z}\frac{\partial}{\partial w}\sum_{\mathbf{v}}\sum_{\mathbf{h}} P^{\star}(\mathbf{v},\mathbf{h}) && \leftarrow \text{trick 1} \\
&= \frac{1}{Z}\sum_{\mathbf{v}}\sum_{\mathbf{h}} \frac{\partial}{\partial w} P^{\star}(\mathbf{v},\mathbf{h}) \\
&= \frac{1}{Z}\sum_{\mathbf{v}}\sum_{\mathbf{h}} P^{\star}(\mathbf{v},\mathbf{h})\frac{\partial}{\partial w}\log P^{\star}(\mathbf{v},\mathbf{h}) && \leftarrow \text{trick 2} \\
&= \underbrace{\sum_{\mathbf{v}}\sum_{\mathbf{h}} P(\mathbf{v},\mathbf{h})}_{\text{average over joint!}}\frac{\partial}{\partial w}\log P^{\star}(\mathbf{v},\mathbf{h}) && \leftarrow \text{via eqtn 1}
\end{aligned}$$

gradient as a whole

Putting the two terms back together we get a total gradient of:

$$\frac{\partial}{\partial w} \log L \quad \propto$$

$$\underbrace{\frac{1}{N} \sum_{\mathbf{v} \in \mathcal{D}}}_{\text{data}} \underbrace{\sum_{\mathbf{h}} P(\mathbf{h} \mid \mathbf{v})}_{\text{av. over posterior}} \; \frac{\partial}{\partial w} \log P^{\star}(\mathbf{x}) \quad - \quad \underbrace{\sum_{\mathbf{v},\mathbf{h}} P(\mathbf{v}, \mathbf{h})}_{\text{av. over joint}} \; \frac{\partial}{\partial w} \log P^{\star}(\mathbf{x})$$

Both terms are some sort of average over $\frac{\partial}{\partial w} \log P^{\star}(\mathbf{x})$, so that's a quantity of crucial importance.

Another way to write the overall gradient:

$$\left\langle \frac{\partial}{\partial w} \log P^{\star}(\mathbf{x}) \right\rangle_{\mathbf{v} \in \mathcal{D},\ \mathbf{h} \sim P(\mathbf{h}|\mathbf{v})} \qquad - \qquad \left\langle \frac{\partial}{\partial w} \log P^{\star}(\mathbf{x}) \right\rangle_{\mathbf{x} \sim P(\mathbf{x})}$$

clamped / wake phase

↑↑↑ conditioned hypotheses

unclamped / sleep / free phase

↓↓↓ random fantasies