Learn with Hidden Variables

\[ P(a,h|x) = 1 \cdot P(a) \cdot P(h|x) \]
\[ P(d|x) = \sum_h P(d|a,h) \]

Maximum Likelihood (ML) estimate of \( \theta \):
\[ \hat{\theta} = \arg \max_{\theta} P(d|x) = \arg \max_{\theta} -\log P(d|x) \]

Claim: minimizing \(-\log P(d|x)\) with respect to \( \theta \) is equivalent to minimizing
\[ F(\theta, q) = -\log P(d|x) + \sum_h q(h) \log \frac{P(h|x)}{P(h|d,x)} \]

with respect to \( \theta \) and \( q(h) \), where \( q(h) \) is a probability distribution on the hidden variables \( h \), i.e., \( q(h) \geq 0 \) for all \( h \), and \( \sum_h q(h) = 1 \).

Proof:
\[ \sum_h q(h) \log q(h) \] is the Kullback-Leibler divergence,
\[ P(h|x) \] divergence.

It \( \geq 0 \), with \( = 0 \) only if \( q(h) = P(h|x) \).

So to minimize \( F(\theta, q) \), you can minimize w.r.t. \( q(h) \) to set \( q(h) = P(h|x) \), then you have to minimize \(-\log P(d|x)\) w.r.t. \( \theta \), which is the original problem.

\[ F(\theta, q) \] can be rewritten as
\[ F(\theta, q) = \sum_h q(h) \log q(h) - \sum_h q(h) \log P(d|a,h) \]

Try to minimize \( F(\theta, q) \) by coordinate descent:

(i) Fix \( \theta^{i+1} \), minimize \( F(\theta, q) \) w.r.t. \( q \) to get \( q^{i+1} \).

(ii) Fix \( q^{i+1} \), minimize \( F(\theta, q) \) w.r.t. \( \theta \) to get \( \theta^{i+1} \).

Repeat.

Each step is guaranteed to reduce \( F(\theta, q) \). So the algorithm will converge to a local or global minimum.

\( F(\theta, q) \) can have local minima, so no guarantee that the algorithm will reach a global minimum, i.e., EM may not converge to the ML estimate of \( \theta \).
(1) Minimise \( F \{ \hat{q}^t, \hat{\lambda} \} \) wrt. \( \lambda \).

Guideline: \( q^{t+1}(h) = P(\hat{h} | d, \hat{\lambda}^t) = \frac{P(h, d | \hat{\lambda}^t)}{\sum_h P(h, d | \hat{\lambda}^t)} \)

requires the ability to calculate \( \sum_h P(h, d | \hat{\lambda}^t) \)

may be difficult

(2) Minimise \( F \{ \hat{\lambda}, \hat{q}^t \} \) wrt. \( \lambda \).

Solve: \( \frac{\partial F}{\partial \lambda} = 0 \) \( \Rightarrow \sum_h \frac{\partial}{\partial \lambda} \hat{q}(h) \langle \hat{q}(h, d) - \log \hat{q}(h) \rangle \)

\( \sum_h \hat{q}(h) \hat{q}(h, d) = \sum_h \hat{q}(h, d) P(h, d | \hat{\lambda}^t) \)

\( \text{statistic of } \hat{\lambda} \text{ wrt. the model with parameter } \hat{q}(h) \)

\( \text{expected statistic of the hidden variables } \text{ wrt. } \hat{q}(h) \)

\( \text{Compare to ML formula for learning a model without hidden variables} \)

Note: Solving this equation is often not easy because it requires computing \( \sum \frac{\partial}{\partial \lambda} \hat{q}(h, d) P(h, d | \hat{\lambda}^t) \)

There are hidden variables so we cannot match the data statistics \( \hat{q}(h, d) \) to the model statistic \( \sum \hat{q}(h, d) P(h, d | \hat{\lambda}^t) \) because we do not know \( h \).

So instead we try to estimate a distribution \( \hat{q}(\lambda) \) over \( h \).

This is like a chicken and egg problem

\( \Rightarrow \) we estimate \( q(h) \) assuming we know \( \lambda \) \( \Rightarrow \frac{\partial F}{\partial \lambda} = 0 \)

\( \Rightarrow \) then we estimate \( \lambda \) assuming we know \( q(h) \) \( \Rightarrow \frac{\partial F}{\partial \lambda} = 0 \)

Extension to multiple data \( D = \{ d^m : m = 1 \to N \} \)

ML minimises \( \sum \log P(d^m | h) \)

\( F \{ \lambda, \{ q^m \} \} = -\sum \sum m \hat{q}^m(h_m) \log q^m(h_m) \)

\( \text{wrt. } \lambda \text{ and } \{ q^m \} \).

Minimise \( \text{wrt. } q^m \)

\( q^{t+1}(h_m) = P(h_m | d_m, \hat{\lambda}^t) \quad m = 1 \to N \)

Minimise \( \text{wrt. } \lambda \)

\( -\sum \sum \hat{q}(h_m) \hat{q}(h_m, d_m) \sum \hat{q}(h_m) \hat{q}(h_m, d_m) \quad m = 1 \to N \)