Bayes Network description of the learning problem

\[
P(S, h, x_4, y_4) = \prod_i P(y_i \mid x_i, h) \cdot P(x_i) \cdot P(h) \cdot P(y_4 \mid x_4, h) \cdot P(x_4)
\]

\[
= P(S\mid h) \cdot P(h) \cdot P(y_4 \mid x_4, h) \cdot P(x_4)
\]

\[
= P(h\mid S) \cdot P(S) \cdot P(y_4 \mid x_4, h) \cdot P(x_4)
\]
Making a Prediction: Bayesian Model Averaging

Goal: given $S$, $x_4$, predict $y_4$

\[
P(y_4|x_4, S) = \frac{P(S, x_4, y_4)}{P(S, x_4)} = \frac{\sum_h P(S, x_4, y_4, h)}{P(S)P(x_4)} = \frac{\sum_h P(h|S)P(S)P(x_4)P(y_4|x_4, h)}{P(S)P(x_4)} = \sum_h P(h|S)P(y_4|x_4, h)
\]
Maximum A Posteriori (MAP) Estimation

- Bayesian model averaging is usually infeasible to compute.

- Replace the Bayesian model average by the best single model $h^{MAP}$

$$
P(y = k | S, x) = \sum_{h \in H} P(y = k | h, x) P(h | S) \approx P(y = k | h^{MAP}, x)
$$

where

$$
h^{MAP} = \arg\max_h P(h | S) = \arg\max_h P(S | h) P(h)
$$
MAP = Penalized Maximum Likelihood

We can view $P(h)$ as a “complexity” penalty on the maximum likelihood hypothesis.

$$h^{MAP} = \arg \max_h P(S|h)P(h)$$
$$= \arg \max_h \log P(S|h) + \log P(h)$$
$$= \arg \max_h \ell(h) + \log P(h)$$
Where does $P(H)$ come from?

**Theory:** $P(H)$ should encode *all and only* our prior knowledge about $H$.

**Practice:**
- Complexity-based priors
  - penalize large neural network weights
  - penalize large SVM weights
  - penalize large decision trees
  - penalize long “description lengths”
- Knowledge-based priors
  - Bayes net structure prior
  - qualitative monotonicity priors
  - smoothness priors