Outline

• Controlling complexity in Bayesian neural networks
• Controlling complexity in infinite mixture models
• Discussion
  – Computational strengths and weaknesses
  – Cognitive relevance
How to choose control parameters?

- Bayesian Occam’s razor
Demo

- Smaller weights (higher $\alpha$) yield simpler models
  - neural_net.m
  - architecture:
    - 2 inputs
    - 1 output
    - 100 hidden units
Two approaches to choosing control parameters

- Evidence maximization (traditional Bayesian Occam’s razor).
- Automatic relevance determination (ARD).
How to choose control parameter?

- Bayesian Occam’s razor

\[ H = \alpha \]
Evidence maximization

\[
p(y|X, \alpha) = \int p(y|X, \theta)p(\theta|\alpha)\,d\theta
\]

\(\theta\): Weight space

Image removed due to copyright considerations.
Bayesian Occam’s Razor

\[ P(D|\mathcal{H}_i) = \int P(D|w, \mathcal{H}_i) P(w|\mathcal{H}_i) \, dw \quad \mathcal{H} = \alpha \]

\[ \begin{align*}
D & \quad P(D|w, \mathcal{H}_i) \\
\mathcal{H}_i & \quad P(w|\mathcal{H}_i) \\
\mathcal{H}_i & \quad P(D|w, \mathcal{H}_i) P(w|\mathcal{H}_i)
\end{align*} \]

Images removed due to copyright considerations.

\[ P(D|\mathcal{H}_i) \sim \text{peak height} \times \text{width} \]
Bayesian Occam’s Razor

\[ P(D | \mathcal{H}_i) = \int P(D|\mathbf{w}, \mathcal{H}_i) P(\mathbf{w}|\mathcal{H}_i) \, d\mathbf{w} \]

\begin{align*}
D & \quad P(D|\mathbf{w}, \mathcal{H}_i) & \quad P(\mathbf{w}|\mathcal{H}_i) & \quad P(D|\mathbf{w}, \mathcal{H}_i)P(\mathbf{w}|\mathcal{H}_i) \\
\end{align*}

Images removed due to copyright considerations.

\[ P(D | \mathcal{H}_i) \simeq \text{peak height} \times \text{width} \]

\[ P(D | \mathcal{H}_i) \simeq P(D | \mathbf{w}_{\text{MP}}, \mathcal{H}_i) \times P(\mathbf{w}_{\text{MP}} | \mathcal{H}_i) \sigma_w |_D \]
Bayesian Occam’s Razor

\[ P(D | \mathcal{H}_i) = \int P(D | w, \mathcal{H}_i) P(w | \mathcal{H}_i) \, dw \]

\begin{align*}
D & \quad P(D|w, \mathcal{H}_i) \quad P(w | \mathcal{H}_i) \quad P(D|w, \mathcal{H}_i)P(w | \mathcal{H}_i)
\end{align*}

Images removed due to copyright considerations.

\[ P(D | \mathcal{H}_i) \simeq \underbrace{P(D | w_{\text{MP}}, \mathcal{H}_i)}_{\text{Evidence}} \times \underbrace{P(w_{\text{MP}} | \mathcal{H}_i) \sigma_w}_{\text{Best fit likelihood}} \]

Evidence \simeq \text{Best fit likelihood} \times \text{Occam factor}
Bayesian Occam’s Razor

\[ P(D | \mathcal{H}_i) = \int P(D | w, \mathcal{H}_i) P(w | \mathcal{H}_i) \, dw \]

\[ D \quad P(D | w, \mathcal{H}_i) \quad P(w | \mathcal{H}_i) \quad P(D | w, \mathcal{H}_i) P(w | \mathcal{H}_i) \]

Images removed due to copyright considerations.

\[ P(D | \mathcal{H}_i) \simeq P(D | w_{MP}, H_i) \times P(w_{MP} | \mathcal{H}_i) \det^{-\frac{1}{2}}(A/2\pi) \]

Evidence \simeq \text{Best fit likelihood} \times \text{Occam factor}

\[ A = -\nabla \nabla \log P(w | D, \mathcal{H}_i) \]
Multiple levels of inference

Different architectures:
# number of hidden layers, kinds of hidden units, etc.

Image removed due to copyright considerations.
Automatic Relevance Determination

• Relation to Kruschke’s “Backprop with attentional weights on inputs”.

• Could specify different classes of features, and learn which class is most relevant for a given classification.
  – Shape and material properties in word learning
  – Internal anatomy versus surface markings in biological classification.

• Applied to weights from hidden units to output units, can effectively infer “size” of bottleneck hidden layer.

• Can apply the same idea to other probabilistic models, e.g., sparseness priors in generative models.
Comparison with cross-validation

• Advantages:
  – Clear theoretical justification.
  – Uses all of the data.
  – Works with many control parameters.
  – Optimize over control parameters in parallel to (or instead of) optimizing over model parameters.
  – Works well in practice (Neal’s ARD triumph)

• Disadvantages
  – Not as intuitive
Comparison with SVMs

• A deep similarity
  – Classification using a model with as many free parameters as possible.
  – Control complexity via sparseness

• Some differences
  – SVM (max margin hyperplane) uses data vectors sparsely, while ARD uses features sparsely.
  – SVM is rotationally invariant; ARD is not.
  – ARD solution may be more interpretable.
  – ARD idea more extendable.
Comparison with SVMs

• What makes a good model?
  – SVM (PAC learning approach): high probability of good generalization
  – Bayesian Occam’s razor: most likely to be the model that generated the data.

• In a non-parametric setting, generalization guarantees seem desirable.
  – PAC-Bayesian theorems (MacAllester, 1998 ff)
PAC-Bayes error bounds for stochastic model selection (McAllester 1998):

- Given model class $T$, classify by choosing consistent hypotheses in $T$ in proportion to their probability.

- For any model class $T$ and any $d > 0$, with probability $1 - d$ over the choice of an I.I.D. sample of $m$ labeled instances $Y_{obs}$, the expected error rate of classifying based on is bounded by:

$$
\ln \frac{1}{p(Y_{obs} \mid T)} + \frac{1}{\delta} + 2 \ln m + 1
$$

Label evidence:

$$
\frac{\ln \frac{1}{p(Y_{obs} \mid T)} + \frac{1}{\delta} + 2 \ln m + 1}{m}
$$

The better the model class fits the observed labels, the tighter the bound on generalization.
Comparison with SVMs

• What makes a good model?
  – SVM (PAC learning approach): high probability of good generalization
  – Bayesian Occam’s razor: most likely to be the model that generated the data.

• In a non-parametric setting, generalization guarantees seem desirable.
  – PAC-Bayesian theorems (MacAllester, 1998 ff)
  – PAC-Bayes-MDL (Langford and Blum)
Outline

• Controlling complexity in Bayesian neural networks
• Controlling complexity in infinite mixture models
• Discussion
  – Computational strengths and weaknesses
  – Cognitive relevance
Outline

• Controlling complexity in Bayesian neural networks
• Controlling complexity in infinite mixture models
• Discussion
  – Computational strengths and weaknesses
  – Cognitive relevance
Advantages of the infinite mixture relative to finite model w/ Bayesian Occam’s razor

• Allows number of classes to grow as indicated by the data.
• Doesn’t require that we commit to a fixed -- or even finite -- number of classes.
• Computationally much simpler than applying Bayesian Occam’s razor to finite mixture models of varying sizes, or thorough cross-validation procedures. *Experience this yourself*….
• Use of MCMC avoids problem of local minima in EM approach to learning finite mixture models.
• BUT: Do we lose the “objective” nature of our complexity control?
Unsupervised learning of topic hierarchies
(Blei, Griffiths, Jordan & Tenenbaum, NIPS 2003)

Image removed due to copyright considerations. Please see:
A generative model for hierarchies

Nested Chinese Restaurant Process:

Image removed due to copyright considerations.
J. ACM abstracts

Image removed due to copyright considerations.