Introduction to
Infinite Models

Radford M. Neal
University of Toronto

NIPS 2003
Why use an infinite model?

- Simple models don't correspond to reality.
- We avoid the problem of deciding on the size of a finite model.
- Infinite models are sometimes simpler than finite models.
But won't an infinite model overfit?

- Not for a Bayesian model with a well-designed prior distribution.
- Not for some other methods as well (e.g., SVMs).
- How can we tell if our prior is well-designed? Sample from it, and look at the results.
How can we implement an infinite model?

- Analytically reduce it to something finite (e.g., Gaussian process models).
- Approximate it by a large finite model.
- Find a trick that lets us represent only as much complexity as is needed for a given data set.
- Usually, we use MCMC.
Examples of infinite models

- Neural networks with infinite numbers of hidden units
  → Gaussian processed, if priors have finite variance.
  → Non-gaussian stable processes for infinite variance priors.

- Mixture models with infinite number of components — "Dirichlet process mixture models"

- Infinite HMMs.

- Infinite factor analysis models? Others...?
The Multilayer Perceptron Network

The network computes its outputs, $y_1, \ldots, y_K$, from its inputs, $x_1, \ldots, x_I$, using a layer of $N$ hidden units:

$$y_k(x) = b_k + \sum_{j=1}^{N} v_{jk} h_j(x)$$

$$h_j(x) = \tanh(a_j + \sum_i u_{ij} x_i)$$

These networks can approximate any function arbitrarily well, with enough hidden units, and appropriate weights, $w = \{a_j, u_{ij}, b_j, v_{jk}\}$.

Variations: More layers of hidden units
Direct input-to-output connections
Other activation functions
Gaussian Network Priors

Recall the equations defining the function computed by a network with \( N \) hidden units:

\[
y_k(x) = b_k + \sum_{j=1}^{N} v_{jk} h_j(x)
\]

\[
h_j(x) = \tanh(a_j + \sum_i u_{ij} x_i)
\]

We can give independent Gaussian priors to all the network parameters:

\[
a_k \sim \text{Gaussian}(0, \sigma_a^2)
\]

\[
u_{ij} \sim \text{Gaussian}(0, \sigma_u^2)
\]

\[
b_k \sim \text{Gaussian}(0, \sigma_b^2)
\]

\[
v_{jk} \sim \text{Gaussian}(0, \sigma_v^2)
\]

This produces a prior distribution for the outputs of the network at any selection of input points — i.e. for \( y(x_1), y(x_2), \ldots \).
Limit for a Gaussian Network Prior

Consider the prior over functions as the number of hidden units, $N$, goes to infinity.

Look at one component, $k$, of the function evaluated at a single point, $x_1$:

$$y_k(x_1) = b_k + \sum_{j=1}^{N} v_{jk} h_j(x_1)$$

The first term above is Gaussian.

The second will become Gaussian as $N \to \infty$, as long as each term has finite variance. Since $h_j(x_1)$ is bounded, this must be the case.

Hence $y_k(x_1)$ becomes Gaussian for large $N$. Its distribution reaches a limit if we make $\sigma_v$ scale as $N^{-1/2}$.

Similarly, the joint distribution of the function at any number of input points converges to a multivariate Gaussian — i.e. we have a Gaussian process prior over functions.
Some Properties of the Gaussian Network Prior

- The hidden-to-output weights go to zero as the number of hidden units goes to infinity.

- With a smooth hidden unit activation function, the functions are smooth, and
  \[
  \text{Corr}[y(x_1), y(x_2)] \approx 1 - |x_1 - x_2|^2
  \]

- If the hidden units use a step function, the functions are locally Brownian, and
  \[
  \text{Corr}[y(x_1), y(x_2)] \approx 1 - |x_1 - x_2|
  \]

- With more complex priors, one can get fractional Brownian functions, with
  \[
  \text{Corr}[y(x_1), y(x_2)] \approx 1 - |x_1 - x_2|^p
  \]
  with \(1 < p < 2\).

- The functions computed by different output units are independent.
Functions from the Gaussian Prior

These networks had 10000 hidden units
How Interesting is the Gaussian Prior?

The Gaussian prior stays reasonable as $N$ goes to infinity, but the result is a bit disappointing:

- No "hidden features" exist, since the effects of any single hidden unit are infinitesimal.

- Even with more hidden layers, we still get a Gaussian process prior (perhaps with a different covariance function).

- We can probably do inference for Gaussian processes more easily without using neural networks — though perhaps neural networks produce interesting covariance functions.
Priors Based on Other Stable Distributions

If $X_1, \ldots, X_N$ are i.i.d. from a stable distribution of index $\alpha \in (0, 2]$, then

$$(X_1 + \cdots + X_N) N^{-1/\alpha}$$

has the same stable distribution. The same is true as $N \to \infty$ if the $X_i$ are in the "domain of attraction" of the stable distribution.

If we give the hidden-to-output weights a prior in the domain of attraction of the stable distribution of index $\alpha$, and scale the width of the prior as $N^{-1/\alpha}$, we should get a well-defined $N \to \infty$ limit.

For example: Use a Cauchy prior for the $v_{jk}$ with width parameter scaling as $N^{-1}$. 
Functions from the Cauchy Prior

These networks had 10000 hidden units
Properties of Non-Gaussian Stable Priors

- The hidden-to-output weights do not go to zero, but asymptotically come from a Poisson process. This is an alternative way of defining the prior.

- The functions computed by different output units can be made dependent without being correlated, by making the weights from the same hidden unit be dependent.

- Networks with more than one hidden layer can produce interesting new effects.

- Analysis seems more difficult than for Gaussian priors.
The Dirichlet Process

A DP is a distribution over discrete distributions.

DP parameters: Base distribution, Go
               "concentration" parameter, a

Picking a distribution from a DP can be visualized as picking an indefinite number of points from such a distribution.

- Pick the first point from Go
- Pick the n-th point from Go (independently) with probability \( \frac{\alpha}{\alpha+n-1} \). Otherwise, let the n-th point be the same as the i-th point, picking i numbers from \( 1, 2, \ldots, n-1 \).

Since \( \sum_{n=1}^{\infty} \frac{\alpha}{\alpha+n-1} \to \infty \), an infinite number of distinct values will be chosen from Go, but only a few will have high weight.
Illustration of a DP

Let $G \sim DP(G_0, \alpha)$, with $G_0 = N(0, 1)$.

![Graph showing a DP distribution with a stick plot]
Dirichlet Process Mixtures

We may not want a discrete distribution. We can make a distribution from a DP continuous by convolving it with a Gaussian.

More generally: Use the DP to define a mixing distribution:

\[ G \sim \text{DP}(\theta_0, \alpha) \]
\[ \Theta_i | G \sim G, \]
\[ X_i | \Theta_i \sim f_{\Theta_i} \]

Example: \( \Theta = (\mu, \sigma^2) \)
\[ f_{\Theta} = N(\mu, \sigma^2) \]

\[ \sigma^2 \]

\[ \mu \]

\[ \alpha \]

\[ \theta_0 \]
Another view of DP mixtures

Consider a mixture model with $K$ components:

$\Pi_i \sim \text{Dirichlet} \left( \frac{\alpha}{K}, \ldots, \frac{\alpha}{K} \right), \ i = 1 \ldots K$

$\Theta_i \sim \mathcal{G}_0, \ \ i = 1 \ldots K$

$Z_j | \Pi \sim \Pi, \ \ j = 1 \ldots n$

$X_j | Z_j, \Theta \sim f_{\Theta, Z_j}, \ \ j = 1 \ldots n$

As $K \to \infty$, this model converges to the corresponding DP mixture model.

The $\frac{\alpha}{K}$ scaling ensures that as $K \to \infty$, a few of the components still have large values for $\Pi_i$. 

Data From a Dirichlet Process Mixture

Data sets of increasing size from a Gaussian model with $\alpha = 1$, $\sigma_x = 0.14$, and $\sigma_\mu = 0.99$. 

$n = 10$

$n = 40$

$n = 200$

$n = 1000$
Predictions from models of increasing complexity

1. Gaussians fit to each component
2. Three-component mixture, predictive distribution
3. Dirichlet process mixture, predictive distribution
4. Dirichlet diffusion tree, predictive distribution
Distributions from the posterior of the three-component mixture
Distributions from the posterior of the Dirichlet process mixture
Distributions from the posterior of the Dirichlet diffusion tree