Analysis Techniques
Inference – part 2

Harrison B. Prosper

NEPPSR 2007

Outline

- Introduction
- Kernel Density Estimation
- Function Approximation

Introduction – 1

- Typical tasks in an analysis
  - Generating simulated data
  - Constructing likelihood function
  - Constructing functions
  - Constructing prior density
  - Computing posterior density
  - Optimization

Introduction – 2

Inferences can done, at all stages, using Bayes' theorem

posterior density | likelihood | prior density

$$p(\theta,\lambda \mid x) = \frac{p(x \mid \theta,\lambda)\,\pi(\theta,\lambda)}{\int p(x \mid \theta,\lambda)\,\pi(\theta,\lambda)\,d\theta\,d\lambda}$$

marginalization

$$p(\theta \mid x) = \int_{\lambda} p(\theta,\lambda \mid x)\,d\lambda$$

Example: Top Mass – Run I

Data partitioned into **M** bins and modeled by a sum of **N** sources of strength **p**. The numbers **A** are the source distributions for the **m**th model.

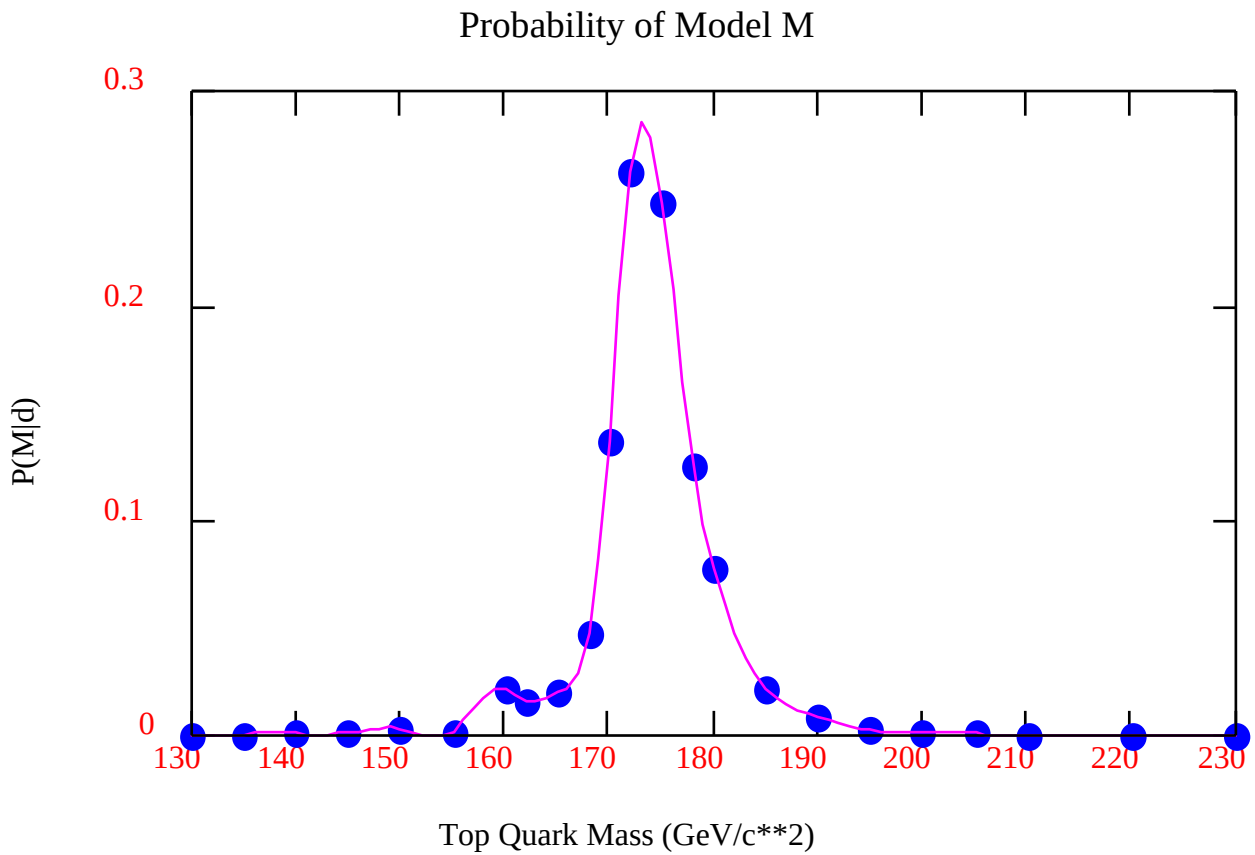
model $$d_i = \sum_{j=1}^{N} p_j a_{ji}$$

likelihood $$P(D \mid a, p, m) = \prod_{i=1}^{M} \exp(-d_i) d^{D_i} / D_i !$$

prior $$\pi(a, p, m) = \pi(p) \prod_{j=1}^{N} \exp(-a_{ji}) a_{ji}^{A_{ji}} / A_{ji} !$$

posterior $$P(m \mid D) = \int \cdots \int P(a, p, m \mid D)\, da\, dp$$

Example: Top Mass – Run I



mtop = 173.5 ± 4.5 GeV

s = 33 ± 8

b = 50.8 ± 8.3

To Bin Or Not To Bin – 1

- Binned – Pros
  - Likelihoods can be modeled accurately
  - No fitting is required
  - Bins with low counts can be handled precisely
  - Statistical uncertainties easily handled

- Binned – Cons
  - Information loss can be severe
  - Suffers from the curse of dimensionality

To Bin Or Not To Bin – 2

- Un-Binned – Pros
  - No loss of information (in principle)
- Un-Binned – Cons
  - Fitting required
  - Can be difficult to model data accurately
  - If done badly, can suffer from the curse of dimensionality

Eg: Cross Section Measurement – 1

The standard cross section probability model is

model $$d_i = a_i\sigma + b_i$$

likelihood $$P(D\,|\,\sigma, a, b) = \prod_{i=1}^{M} \exp(-d_i) d^{D_i} \Big/ D_i!$$

prior $$\pi(\sigma, a, b) = \pi(a, b\,|\,\sigma)\pi(\sigma)$$

$$= \prod_{i=1}^{M} \exp(-a_i/\alpha)(a_i/\alpha)^{A_i} \Big/ A_i!$$

$$\times \prod_{i=1}^{M} \exp(-b_i/\beta)(b_i/\beta)^{B_i} \Big/ B_i!$$

$$\times\ \pi(\sigma)$$

Exercise 8: Derive the posterior density p(σ|D), assuming a π(σ) = 1

Eg: Cross Section Measurement – 2

Consider making the bins smaller and smaller

$$d_i = \int_i d(x)dx \approx [a(x_i)\sigma + b(x_i)]\Delta x_i$$

the likelihood becomes

$$P(D \mid \sigma, a, b) = \exp(-\sum_i [a(x_i)\sigma + b(x_i)]\Delta x_i)$$

where K is the number of events and a(x) and b(x) are the effective luminosity and background densities

$$\times \prod_{i=1}^{K} [a(x_i)\sigma + b(x_i)]\Delta x_i$$

$$\propto \exp(-a\sigma - b) \prod_{i=1}^{K} [a(x_i)\sigma + b(x_i)]$$

Eg: Cross Section Measurement – 3

The un-binned likelihood function

$$p(D\,|\,\sigma,a,b) = \exp(-a\sigma - b)\prod_{i=1}^{K}[a(x_i)\sigma + b(x_i)]$$

is an example of a marked Poisson likelihood. Each event is "marked" by the signal/background discriminating variable x_i, which can be multi-dimensional.

In principle, this is a much more efficient way to measure a cross section. The downside is the need to model the densities a(x) and b(x).

Kernel Density Estimation

Kernel Density Estimation – 1

The idea is to approximate a density by a sum over kernels, one placed at *each* of the N points x_i of the training sample.

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} k\left(\left\| \frac{x - z_i}{h} \right\| \right)$$

h is a smoothing parameter, called the bandwidth, that is adjusted to provide the best approximation to the unknown density p(x).

If *h* is too small, the model will be very spiky; if *h* is too large, important features of the true density p(x) may be lost.

Kernel Density Estimation – 3

Why does this work? Consider the limit as N → ∞ of

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} k\left(\left\|\frac{x - z_i}{h}\right\|\right)$$

$$\hat{p}(x) \to \int k\left(\left\|\frac{(x - z)}{h}\right\|\right) p(z)dz$$

In the limit N → ∞, the true density p(x) will be recovered provided that h → 0 in such a way that

$$k\left(\left\|\frac{x - z_i}{h}\right\|\right) \to \delta(x - z)$$

Kernel Density Estimation – 4

As long as the kernel behaves sensibly in the N → ∞ limit *any* kernel will do. In practice, the most commonly used kernel is the Gaussian, one for each dimension:

$$k\left(\|x-z\|/h\right)=\exp\left[-\sum_{j=1}^{d}\left(\frac{x-z_{ji}}{h_j}\right)^2/2\right]/\prod_{j=1}^{d}h_j\sqrt{2\pi}$$

One advantage of the KDE approximation is that it contains very few adjustable parameters, namely, the bandwidths h_i, a rough estimate of which is

$$h_j=\sigma_j\left\{\frac{4}{(d+2)N}\right\}^{1/(d+4)}$$

σ_i is standard deviation of the data in the i^{th} dimension

Kernel Density Estimation – 5

How is the value of the smoothing parameter to be chosen?

One way, is to minimize the Kullback-Leibler divergence:

$$d(p, \hat{p}) = \int p(x) \ln \left(\frac{p(x)}{\hat{p}(x)} \right) dx$$

$$= \int p(x) \ln p(x) dx - \int p(x) \ln \hat{p}(x) dx$$

$$\approx \text{constant} - \frac{1}{N} \sum_{i=1}^{N} \ln \hat{p}(x_i)$$

Or, equivalently, minimize $\boxed{-\frac{1}{N} \sum_{i=1}^{N} \ln \hat{p}(x_i)}$ with respect to the bandwidth

Kernel Density Estimation

Practical Issues

- One difficulty with smoothing globally is that in regions where the density of points is relatively low, the kernels will tend to be too far apart.
- A sharp boundary is difficult to model unless a way is found, in effect, to continue the data across the boundary.
- Every evaluation of p(x) requires the evaluation of N (d-dimensional) kernels. If N is large this can be computationally burdensome.

Function Approximation

Function Approximation – 1

We are interested in the relationship between "inputs", or "features"

$$x = (x_1, x_2, \ldots, x_n)$$

and some "output", or "response", y, where

$$y = f(x)$$

But usually neither f(x) nor the form of the probability model Pr(x,y) that generated the data is known

Function Approximation – 2

Given N examples $(x,y)_1$, $(x,y)_2$,…$(x,y)_N$ we wish to construct an approximation to y = f(x).

There are two general approaches to the problem:

Machine Learning

Teach a "machine" to learn f(x) by feeding it examples, that is, training data T.

Bayesian Inference

Infer f(x) given the likelihood of the training data T and a prior on the space of functions f(x).

Machine Learning – 1

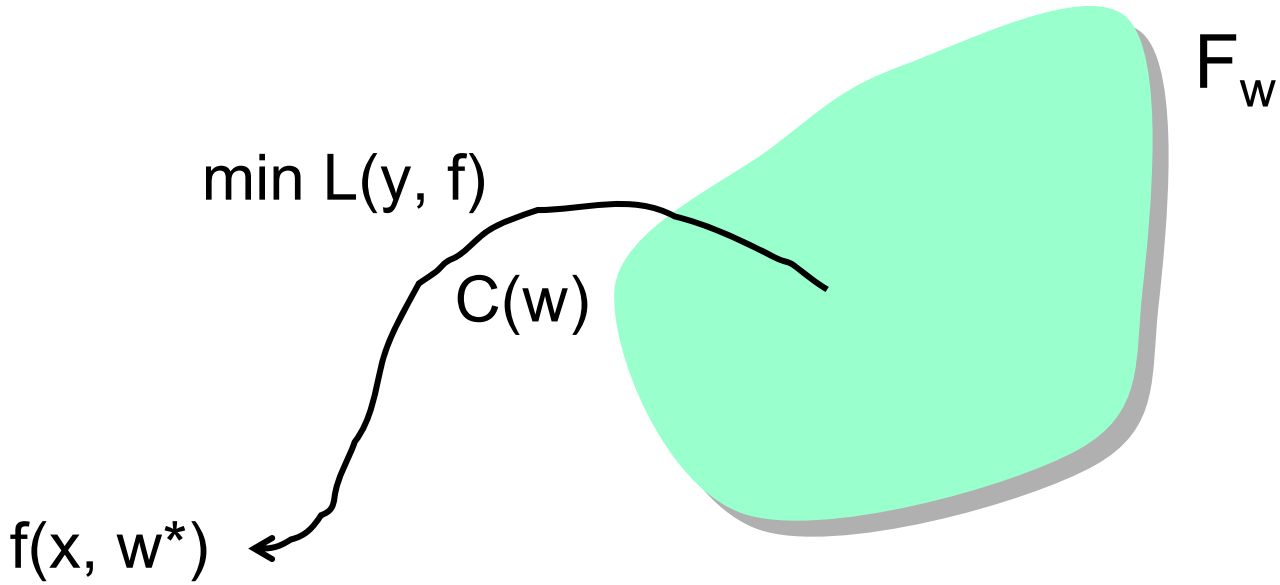
Given N examples $(x,y)_1$, $(x,y)_2$,…$(x,y)_N$ we proceed as follows

We specify:

- A function class $F_w = \{ f(x, w) \}$
- A loss function $L(y, f)$
- A constraint $C(w)$ on the parameters w

L(y, f) measures how much we lose if we make a poor choice from the function class.

Machine Learning – 2



We choose a function f by minimizing the loss L(y, f), subject to the constraint C(w). But, unfortunately, the choice can be quite unstable as we move from one example (x,y) to another.

Machine Learning – 3

To get a more stable choice we minimize not the loss, but rather its ensemble average, called the risk

$$R(f) = \int_{x,y} L(y, f(x,w)) \Pr(x,y)$$

where Pr(x,y) = p(x,y) dxdy

But, again, we do not know R(*f*) so, in practice, we minimize the *empirical risk*:

$$\hat{R}(f) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i, w))$$

Bayesian Approach – 1

Ingredients:

Pr(T|f) the likelihood (of examples)

Pr(f) the prior (over functions)

Then compute:

Pr(f|T) the posterior

using Bayes' theorem:

$$Pr(f|T) = Pr(T|f)\ Pr(f) / Pr(T)$$

Bayesian Approach – 2

In practice, we choose some function class

$F_w = \{ f(x, w) \}$

of parameterized functions f(x, w) and make inferences on the parameters:

$Pr(w|T) = Pr(T|w) Pr(w) / Pr(T)$

Pr(w|T) assigns a probability to each element of the parameter space and hence to each function, f, in F_w.

Bayesian Approach – 3

Given Pr(w|T), how do we pick a function from F_w?

One way is to pick f(x, w*) such that w* maximizes the posterior probability Pr(w|T).

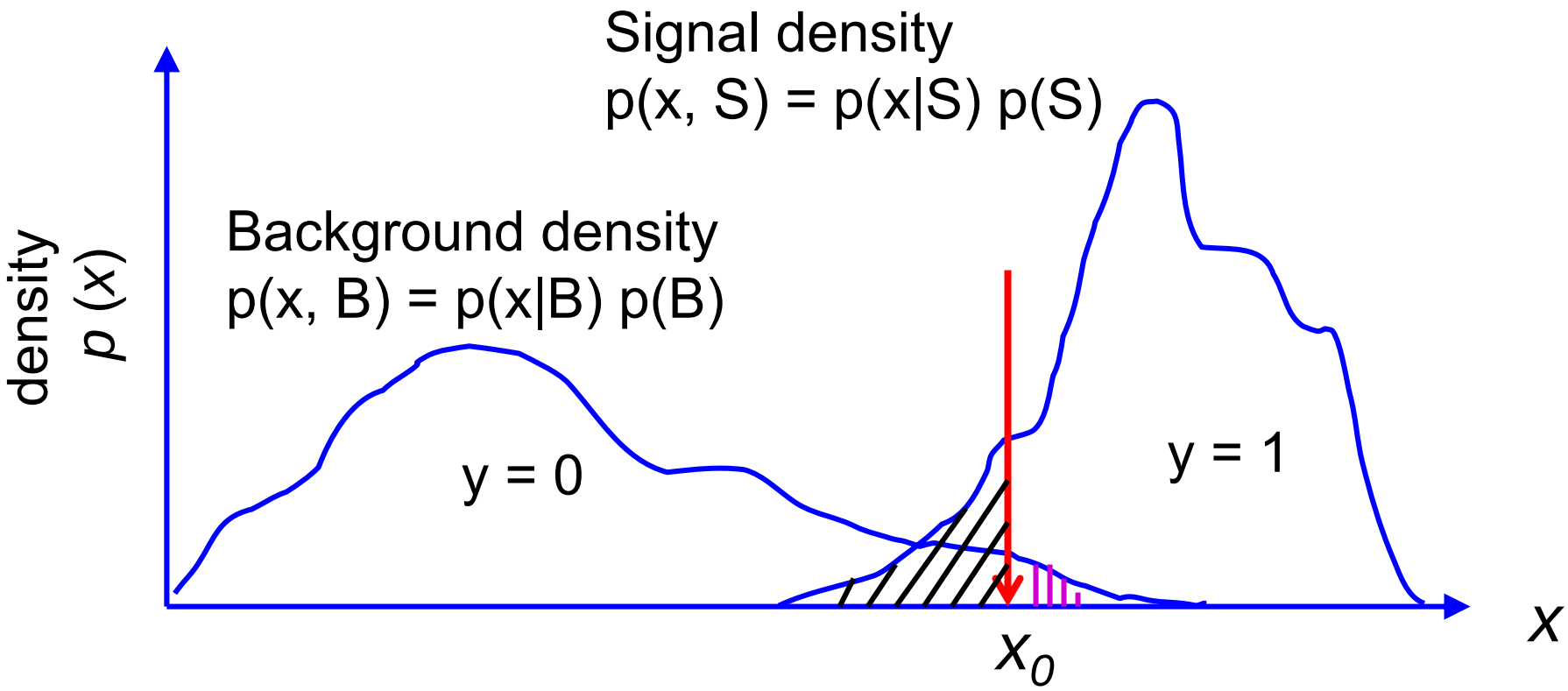
Another is to compute the average

$$f(x|T) = \int f(x, w) \Pr(w|T)$$

Signal/Background Discrimination

Signal/Background Discrimination – 1

Consider the problem in 1-dimension



We wish to minimize the misclassification rate

Signal/Background Discrimination – 2

The cost C of a misclassification is given by

$$C = C_S \int H(x_0 - x)\, p(x, S)\, dx \quad \text{Signal loss}$$

$$+ C_B \int H(x - x_0)\, p(x, B)\, dx \quad \text{Background contamination}$$

where *H(x)* is the Heaviside step function

H(x) = 1 if x > 0, 0 otherwise

and C_S and C_B are costs of misclassifying a signal event and background event, respectively

Signal/Background Discrimination – 3

Minimizing

$$C = C_S \int H(x_0 - x)\, p(x, S)dx + C_B \int H(x - x_0)\, p(x, B)dx$$

with respect to the decision boundary x_0

$$0 = C_S \int \delta(x_0 - x)\, p(x, S)dx - C_B \int \delta(x - x_0)\, p(x, B)dx$$
$$= C_S\, p(x_0, S) - C_B\, p(x_0, B)$$

gives the Bayes discriminant $$r \equiv \frac{C_B}{C_S} = \frac{p(x_0 \mid S)\, p(S)}{p(x_0 \mid B)\, p(B)}$$

Signal/Background Disrimination – 4

The Bayes discriminant, which holds also in the multivariate case, is optimal in that it minimizes the error rate.

$$r = \frac{p(x \mid S)\, p(S)}{p(x \mid B)\, p(B)}$$

It is called the Bayes discriminant because it is just Bayes' theorem in disguise:

$$p(S \mid x) = \frac{r}{1 + r}$$

Any classifier that achieves the minimum error rate is said to have reached the Bayes limit.

Tutorial

- Learn Python
- For a given top quark mass hypothesis, construct likelihood function $p(x|m_t)$ using KDE method with N=5000 points
- Compute log-likelihood for K = 2000 events

$$l(m_t) = -\ln \prod_{i=1}^{K} p(x_i \mid m_t) = -\sum_{i=1}^{K} \ln p(x_i \mid m_t)$$

- Plot $l(m_t)$ vs m_t