Data Mining

Tamás Budavári - budavari@jhu.edu
Class 3

- Sampling from distributions
- Density estimation

Samples, PDFs in 1- and 2-D

Descriptive Statistics

- Characterization of location, dispersion, etc.

	Sample Estimates (notations)	Probabilisty Density Functions
Average	$\bar{x} = \dfrac{1}{N}\sum_{i=1}^{N} x_i = \left\langle x_i \right\rangle_{i=1}^{N}$	$\mu = \mathbb{E}[X] = \int x\, p(x)\, dx$
Variance	$s^2 = \dfrac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2$	$\mathbb{Var}[X] = \int (x-\mu)^2 p(x)\, dx$

- Useful connection to sampling

Sampling from distributions

- Uniform between a and b: scale and shift

> $U_{ab} = a + (b-a)\,U_{01}$

- Inverse transform sampling in $\mathbb{R}$

> $X = \mathrm{CDF}^{-1}(U_{01})$
>
> Unhomework: prove it!

- Rejection sampling - also works in $\mathbb{R}^N$

Numerical Methods

If the $\{x_i\}$ set is sampled from the probability density function $p(\cdot)$, the following will be true:

- Average

$$\mathbb{E}[X] = \int x\, p(x)\, dx \;\approx\; \frac{1}{N} \sum_i x_i$$

- Variance

$$\mathbb{E}[(X-\mu)^2] = \int (x-\mu)^2\, p(x)\, dx \approx \frac{1}{N} \sum_i (x_i-\mu)^2$$

compare to

$$s^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(x_i - \bar{x}\right)^2$$

Bessel correction: $N-1$ independent $(x_i - \bar{x})$ differences

$$\sum_{i=1}^{N} (x_i - \bar{x}) = \;??? \text{ .... 0 ...}$$

```
In [1]: %pylab inline
        from scipy.stats import norm as gaussian
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:

```
# generate sample with size N
mu, sigma, N = 0, 1, 10
x = gaussian.rvs(mu, sigma, N)

avg = np.mean(x)
# variance estimates
s2   = np.sum( (x-avg)**2 ) /(N-1)  # correct
s2n  = np.sum( (x-avg)**2 ) / N     # biased
s2k  = np.sum( (x- mu)**2 ) / N     # known mean
# standard deviation estimates
sqrt(s2), sqrt(s2n), sqrt(s2k)
```

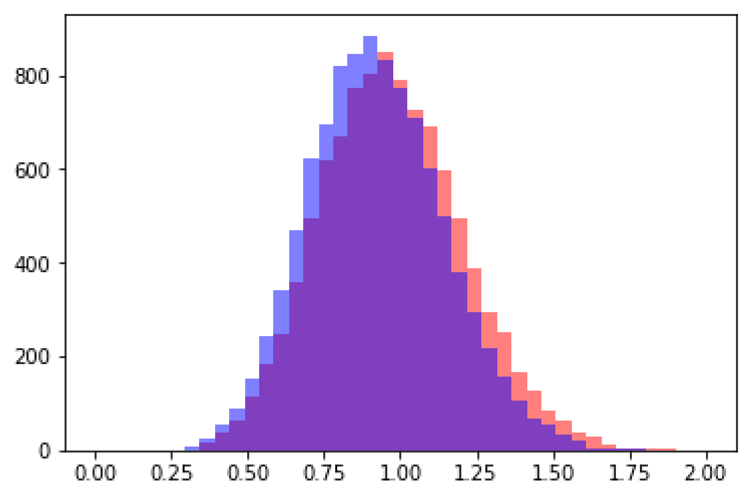
Out[2]:

```
(1.0277817857798131, 0.97503941420983975, 0.99793347576774416)
```

In [3]:

```
# generate M runs with N samples each
mu, sigma, N, M = 0, 1, 10, 10000
X = gaussian.rvs(loc=mu, scale=sigma, size=(N,M))
avg = np.mean(X, axis=0)
print (X.shape, avg.shape)

# variance estimates - check out broadcasting in X-avg
s2   = np.sum( (X-avg)**2, axis=0) /(N-1) # correct
s2n  = np.sum( (X-avg)**2, axis=0) / N    # biased
s2k  = np.sum( (X- mu)**2, axis=0) / N    # known mean

print (s2.shape)

# standard deviation estimates
s, sn, sk = np.sqrt(s2), np.sqrt(s2n), np.sqrt(s2k)
print (mean(s), mean(sn), mean(sk))

hist(s , 41, range=[0,2], color='r', alpha=0.5);
hist(sn, 41, range=[0,2], color='b', alpha=0.5);
```

```
(10, 10000) (10000,)
(10000,)
0.972529285104 0.922622289643 0.975008828713
```



Density Estimation

- Histograms
  - Width of bins, h
  - Start of bin boundary, x_0

$$\text{Hist}(x) = \frac{1}{N}\sum_i \mathbf{1}_{\text{bin}(x_i;x_0,h)}(x)$$

- Kernel Density Estimation (KDE)
  - Bandwidth h

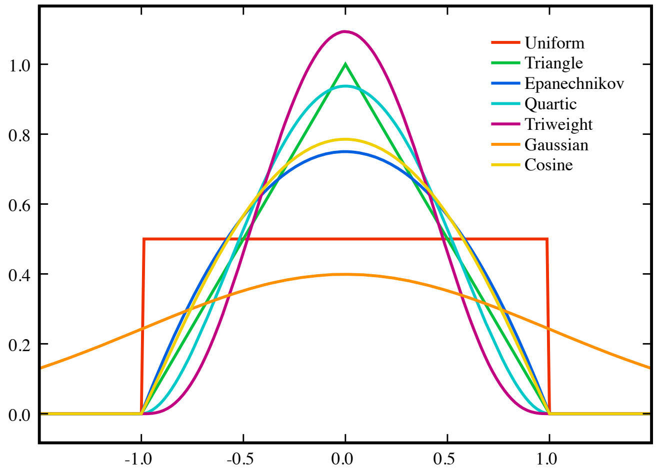
$$\text{KDE}(x) = \frac{1}{N}\sum_i K_h(x-x_i) = \frac{1}{Nh}\sum_i K\left(\frac{x-x_i}{h}\right)$$

  - Can use different $K(\cdot)$ kernel functions
    - E.g., Uniform, Triangular, Gauss, Epanechnikov

See animations at http://www.mglerner.com/blog/?p=28 (http://www.mglerner.com/blog/?p=28)

Kernel Function

- Finite vs Infinite support
- Numerical evaluations
- Frequently used kernels



Learn more about KDE here (https://jakevdp.github.io/blog/2013/12/01/kernel-density-estimation/) and also check out Bayesian Blocks here (https://jakevdp.github.io/blog/2012/09/12/dynamic-programming-in-python/)
— tutorials by Jake Vanderplas

Detour: Dirac delta

- In the limit of $h \to 0$, the kernel will become strange:

> **Dirac's** δ "function" is 0 everywhere except at 0 such that
>
> $$\int \delta(x)\, dx = 1$$

- Interesting properties, e.g.,

> $$\int f(x)\, \delta(x-a)\, dx = f(a)$$

- See **distribution theory** and **functionals** for more background

An interesting result

- Bad density estimation but if...

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i)$$

- The expectation value

$$\mathbb{E}[X] = \int x \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i)\, dx$$

$$\mathbb{E}[X] = \frac{1}{N} \sum_{i=1}^{N} \int x\, \delta(x - x_i)\, dx$$

$$\mathbb{E}[X] = \frac{1}{N} \sum_{i=1}^{N} x_i$$

Unhomework

1. Sample from a mixture of two Gaussians using uniform random numbers in the [0,1) interval. Try different (μ_1, σ_1) and (μ_2, σ_2) values!
2. Build different density estimators and compare to the original PDF. Try histogramming and KDE with different parameters.