Note

• Project topic/papers due in two weeks (10/24).
  – Your literature survey topic and papers must be approved by me. (Submit it to iLearn forum thread).
  – Late policy will be applied.

• Homework #3
  – On Lecture 6-7
  – Due in one week
  – Submit your answers on 10/17 Tuesday 4pm in class

Note

• Complete the Exercise #1
  – HW Assignment: Complete FP#1 on PCA and make your final submission of your code and results (screen shots/short doc report) via iLearn by tomorrow midnight.
  – Submit the code by midnight tonight for extra credit

• Fast Prototyping Exercise #2 on Mean Shift Segmentation starts next week
  – HW Assignment: Read carefully and thoroughly the reference paper:
    https://bidal.sfsu.edu/~kazokada/csc872/PD2.pdf
**Non-Parametric Statistical Modeling**

CSC 872
Pattern Analysis and Machine Intelligence

References
Andrew Moore’s great slides at http://www.cs.cmu.edu/~awm/tutorials

---

**Statistical Modeling: Review**

- **Problem**
  - Estimating probability distribution from data (Bayesian framework: needs likelihood & prior)

- **Assumption**
  - Data = Samples drawn from an unknown underlying distribution

- **Question**
  - What is the underlying distribution given these data?

\[
P(X) \xrightarrow{\text{data}} \{x_1, \ldots, x_N\}
\]
KR: Probability Distribution

- **Probability Mass Function**
  - For a discrete random variable $X$
  - $P(X): \sum x_i P(X = x_i) = 1$
  - $\forall x P(X = x) \geq 0$

- **Probability Density Function**
  - For a continuous random variable $X = x$
  - $p(x): \int_{-\infty}^{\infty} p(x)dx = 1$
  - $\forall x p(x) \geq 0$
  - $P(a < X < b) = \int_{a}^{b} p(x)dx$

What for?

- Bayesian $X$ (= **inference**, classification etc)

- Computing Expectation
  - $E[X] = \sum x_i P(x_i) = \mu$ (population mean)
  - $E[f(X)] = \sum x_i f(x_i) P(x_i)$
  - $E[aX + Y + b] = aE[X] + E[Y] + b$ (linear operator)

- Solving Maximum Likelihood Estimation Numerically

- Anomaly Detection
  - Sort events by associated probability (anomaly=small prob)

- Clustering/Segmentation/VectorQuantization!!!
Non-Parametric Modeling

- **Most simple way** for representing distribution
- Density Estimation
- Basically Two Steps
  1) Domain quantization (Binning)
  2) Measure Frequency / Density (Counting)

- Useful **when no known prior information about functional property** of the distribution

PF: Basic Density Estimation

- Given a sample $x$, a density estimator $M$ can tell you how likely each data item is:
  $$\hat{p}(x|M) \sim p(x)$$

- Given a sample set $\{x_1, \ldots, x_N\}$, find a density estimator $M$ that most accurately estimate the likelihood function

- You can model this with constructing a **normalized histogram** from $\{x_1, \ldots, x_N\}$
PF: Histogram as a Density Estimator

- Given a dataset \( \{x_1, \ldots, x_N\} \)
- Quantization of data space
- Frequency counts

- **Normalization** = divide by total counts
- Modeling Probability Mass Function \( P(x) \)
- Modeling Joint Distribution
  - \( P(x, y, z, \ldots) \)
  - Frequency counting in \( N \)-D feature space

Issues: Bin Size

- Estimation results are very sensitive to quantization!

- For continuous RV, quantization may be difficult
- You cannot cover entire domain accurately
Issues: Differentiability

• For some quantization, you may have some bins having zero values… So what?

• You cannot compute dataset density

\[ \hat{p}(\text{dataset}|M) = \hat{p}(x_1 \land x_2 \land \cdots \land x_N|M) = \prod_{n=1}^{N} \hat{p}(x_n|M) \]

• You cannot differentiate \( P(x|M) \) at bins with zero
  – You cannot compute gradient
  – Later you will see how you don’t like this..

Modeling Probability Density Function

• Probability that \( x \), drawn from unknown density function \( p(x) \), fall inside some region \( R \) (\( V \) is volume of \( R \))

\[ P = \int_{R} p(x') \, dx' \approx p(x) \, V \]

• Given \( N \) points, probability that \( K \) of them fall into inside region \( R \) follows the binomial law. Thus

\[ E[K/N] = P \]

• Since the variance vanishes as \( N \to \infty \), we approximate

\[ P \approx K/N \]

• With above results (two approximations) we have

\[ p(x) \approx \frac{K}{NV} \]

Biship: p50-55
PF: Kernel Density Estimation

- Set region $R$ as a hypercube of edge-length $h$

- **Kernel function** $H(x, h)$ is used to count data points lying inside the hypercube (also known as Parzen window)

$$H(x - x_n, h) = \begin{cases} 
1 & \frac{|x_j - x_{nj}|}{h} < 0.5 \quad j = 1, \ldots, d \\
0 & \text{otherwise}
\end{cases}$$

- So a kernel is a windowing function!!!

PF: Kernel Density Estimation cond

- Counting with the kernel

$$K = \sum_{n=1}^{N} H\left(\frac{|x - x_n|}{h}\right)$$

- **Kernel Density Estimate (KDE)**

$$\hat{p}(x) \approx \frac{K}{NV} = \frac{1}{Nh^d} \sum_{n=1}^{N} H\left(\frac{|x - x_n|}{h}\right)$$
Is KDE different from Histogram?

- Both do simple counting but.....
  - histogram defines the bins in advance
  - YES, KDE removes this quantization step all once!
  - Bin size = Hypercube size = $h$
  - Ok. This seems more flexible but what about the differentiability?
  - **NO!** still discontinuous. $H()$ may not be able to be differentiated at everywhere.

A Good News

- You can generalize the original KDE to a smooth function simply by choosing a right kernel function !!!

- What should we use then?

- **Gaussian**
  - among others ...
KR: Gaussian (Normal) PDF

- Univariate Gaussian PDF
  
  \[ p(x; \mu, h) = N(x; \mu, h) = \frac{1}{\sqrt{2\pi h^2}} \exp\left(-\frac{(x-\mu)^2}{2h^2}\right) \]

- Multivariate Gaussian PDF
  
  \[ p(x; \mu, \Sigma) = \frac{1}{|2\pi\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \]

1) Define \( \delta = x - \mu \)
2) Count number of contours crossed by \( \delta \)
3) \( D = \text{Constant} = \sqrt{(\delta^T\Sigma^{-1}\delta)} = \text{Mahalanobis distance} \)
4) \( \exp(-D^2/2) \)
5) \( x \) close to \( \mu \) in the squared Mahalanobis distance gets higher weight!

Why Gaussian?

- We like Gaussian because
  
  - It is very useful to understand basic PAMI concepts
  - Smooth function: derivative of \( \exp \) is \( \exp \)
  - Fourier transform of Gaussian is Gaussian
  - Gaussian marginal, conditional are Gaussian
  - Linear transform of Gaussian is still Gaussian

- OK. there are much more to this
  
KDE with Gaussian

• Now we have smooth density estimator without the quantization hassle!!!

\[ \hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{|2\pi\Sigma|^{1/2}} \exp(-\frac{(x-x_n)^T\Sigma^{-1}(x-x_n)}) \]

It’s a weighted sum!

Any Catch?

• No quantization hassle but
• You still need to tune the **bandwidth**!
• Which is another research topic…

• **And its complexity is high**
• You need to keep all data points
• For every location, you also need to evaluate every data points
• So it is a lot of computation with lots of data…
PF: Finding Peaks of PDF

- We want to find a peak of PDF because
  - Statistical Modes
  - (Locally) Maximum Likelihood Estimate

This can mean correct location estimate of your video tracker
This could mean there are quite a few pixels with pink color
This could mean that we have aging society (more older people)

PEAKS ARE IMPORTANT

Color=pink  X=“at target”  Age=80

PS: Hill-Climbing

- Find a peak iteratively from an initial point \( x^{init} \)
- Local peak defined as \( \nabla p(x) = 0 \) (horizontal tangent!)
  \[ x^* = \arg \max_x p(x | x_1, \ldots, x_N) \]
- Hill-Climbing by Gradient Ascent
  - Iterative update by \( x_{n+1} = x_n + \eta \nabla p(x_n) \) with small positive \( \eta \)
  - It could oscillate & diverge, depending on \( \eta \)
  - Small \( \eta \) sure-convergence; large \( \eta \) faster convergence
PS: Mean Shift

- Adaptive step-size gradient ascent for PDFs constructed by KDE
  - no $\eta$!
- Provably convergent
  - no oscillation and no divergence!
- Implicit construction of KDE
  - It is local hill-climbing so there is no need to construct KDE for entire domain at once
  - Local density is computed only when needed

Basic Algorithm

Goal: Find the most dense point
Data Sample (Darts/Stars/BodySize)
Goal: Find the most dense point
Data Sample (Darts/Stars/BodySize)

Slides from “Mean Shift Theory and Applications” by Ukrainitz and Sarel
Basic Algorithm

Goal: Find the most dense point
Data Sample (Darts/Stars/BodySize)

Mean Shift Vector
Region of Interest
Mean

Goal: Find the most dense point
Data Sample (Darts/Stars/BodySize)
Basic Algorithm

Goal: Find the most dense point
Data Sample (Darts/Stars/BodySize)

Mean Shift Vector

Region of Interest

Mean
PS: Mean Shift Formulae

- Mean Shift Vector given samples \(x_1, \ldots, x_n, \ldots, x_N\)
  \[
  m(x, h) = \frac{\sum_{n=1}^{N} x_{n} g\left(\left\|\frac{x-x_{n}}{h}\right\|^{2}\right)}{\sum_{n=1}^{N} g\left(\left\|\frac{x-x_{n}}{h}\right\|^{2}\right)} - x
  \]

- Epanechnikov Kernel
  \[g\left(\left\|\frac{x-x_{n}}{h}\right\|^{2}\right) = \begin{cases} C & \left\|x-x_{n}\right\| \leq h \\ 0 & \text{otherwise} \end{cases}\]

- Gaussian Kernel
  \[g\left(\left\|\frac{x-x_{n}}{h}\right\|^{2}\right) = \exp\left(-\frac{\left\|x-x_{n}\right\|^{2}}{h^{2}}\right)\]

- Mean Shift Procedure (iterate this from \(x_i\) to \(x_i^{\text{mean}}\))
  \[x_{k+1} = m(x_k, h) + x_k\]

- Stopping Criteria
  \[
  \frac{\|m(x_k, h)\|^2}{h^2} < TH^2
  \]

PF: Relation to Clustering

- MS is a solution for adaptive clustering!
- No need to know how many clusters are there in data \textit{a priori}

1) For all data points, perform the MS procedure, recording their convergence to nearby modes
2) Label each mode by grouping
3) Re-assign each data points with mode labels
Mean Shift Clustering in 2D

• Advantages
  – No need to know the number of clusters a priori
  – The shape of cluster does not need to be regular

PF: Is it learning?

• Just summarizing data?
  – Descriptive Statistics
  – Inferential Statistics

• What designer must choose?
  – Quantization / Bin Size / Kernel Bandwidth
  – Types of estimation

• What must be derived from data?
  – Data Count / Frequency / Density
Summary

• Non-Parametric Statistical Modeling
  – Work with arbitrary distributions
  – Histogram
  – Kernel Density Estimation
  – Mean Shift Mode Seeking
  – Clustering
  – Memory and Time Complexity…

• Next
  – How to make use of prior knowledge of distribution?
  – Parametric Modeling!
  – FP#2 on Mean Shift Segmentation: READ THE PAPER!!!