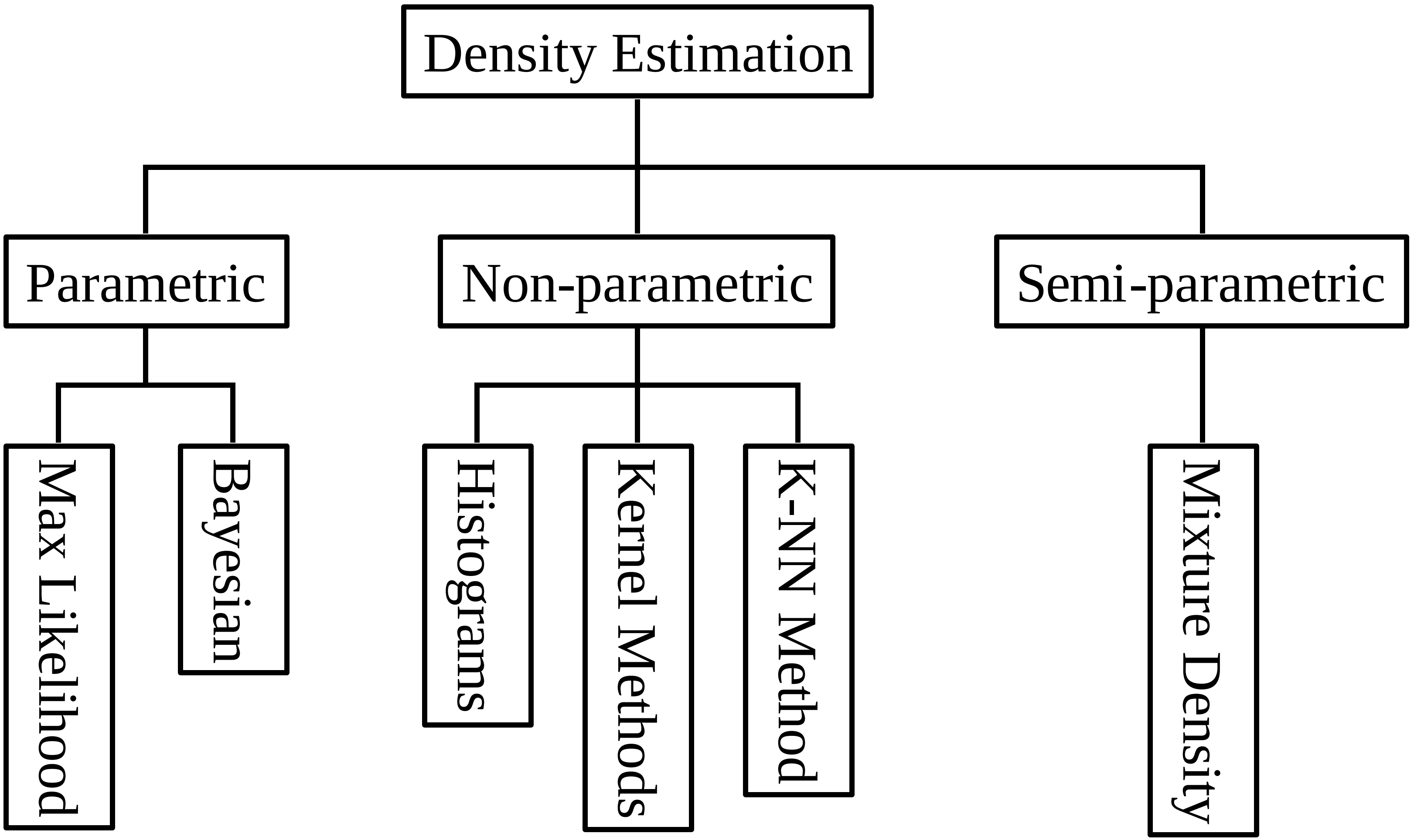
# 9.913 Pattern Recognition for Vision

## Class VI – Density Estimation

Yuri Ivanov

# Road Map


Density Estimation
Parametric
Non-parametric
Semi-parametric
Max Likelihood
Bayesian
Histograms
Kernel Methods
K-NN Method
Mixture Density

# Generative vs. Discriminative

There are two schools of thought in Machine Learning:

1. Generative:
   - *Estimate* *class* *models* from data
   - *Compute* the *discriminant* function
   - Plug in your data – get the answer

2. Discriminative:
   - *Estimate* the *discriminant* from data
   - Plug in your data – get the answer

*Last class*

Density Estimation is at the core of *generative* Pattern Recognition

$$P(a < x < b) = \int_a^b p(x)dx$$

$$\text{mean}: \; E[x] = \int xp(x)dx$$

$$\text{covariance}: E[(x - E[x])(x - E[x])^T]$$

$$= \int \left[ (x - E[x])(x - E[x])^T \right] p(x)dx$$

$$\text{function mean}: \; E[f(x)] = \int f(x)p(x)dx$$

$$\text{conditional mean}: \; E[y \mid x] = \int yp(y \mid x)dx$$

# Refresher

Minimum expected risk:

$$R^* = \int \min_{w} \left[ R(a \mid x) \right] p(x) dx$$

… is based on conditional risk:

$$w_i = \arg\min_{w} R(a \mid x)$$

… which is computed from the posterior:

$$R(a \mid x) = L(a \mid w) P(w \mid x)$$

… which depends on the likelihood:

$$P(w \mid x) = \frac{p(x \mid w) P(w)}{p(x)}$$

# Setting

Data:

$$D = \{D_i\}_{i=1}^{C}$$

$$Assume\ that\ D_j\ contains\ no information\ about\ w_i,\ \forall i \neq j$$

***NOTATIONALLY*** - we abandon the class label:

$$p(x \mid \cancel{w_i}) \Rightarrow p(x)$$

Keep in mind: $p(x \mid w_i) \neq p(x)$

Goal:
model the probability density function $p(x)$, given a finite number of data points, $x_1$, $x_2$, …, $x_N$, drawn from it.

# Three Methods

1. Parametric
   - Good: small number of parameters
   - Bad: choice of the parametric form
2. Non-parametric
   - Good: data “dictates” the approximator
   - Bad: large number of parameters
3. Semi-parametric
   - Good: combine the best of both worlds
   - Bad: harder to design
   - Good again: design can be subject to optimization

# Parametric Density Estimation

Estimate the density from a given functional family

Given: $$p(x \,|\, q) = f(x, q)$$

Find: $$q$$

Two methods of parameter estimation:

1. *Maximum Likelihood* method
   - Parameters are viewed as unknown but fixed values

2. *Bayesian* method
   - Parameters are random variables that have their distributions

# Normal (Gaussian) Density Function

A common assumption - *Gaussian*

$$q = (m, \Sigma)$$

$$p(x \mid q) = \frac{1}{(2p)^{d/2} \mid \Sigma \mid^{1/2}} \exp\left( -\frac{1}{2} \underbrace{\left( (x-m)^T \Sigma^{-1} (x-m) \right)}_{\text{Squared Mahalanobis distance}} \right)$$

Number of dimensions

"Volume" of the covariance

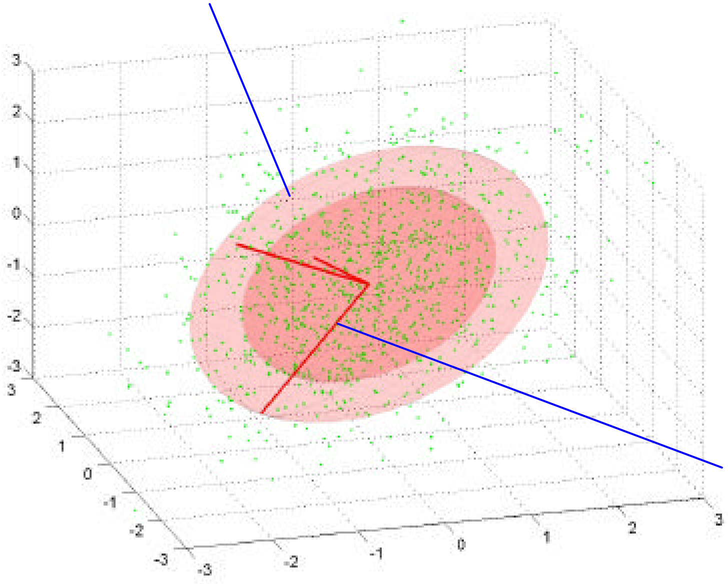
$$m = E[x]$$ – $d$ parameters

$$\Sigma = E\left[ (x-m)^T (x-m) \right]$$ – $d(d+1)/2$ parameters

# Normal Density

$$m = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0 & .5 \\ 0 & 1 & .3 \\ .5 & .3 & 1 \end{bmatrix}$$

Constant density, $(x-m)^T \Sigma^{-1} (x-m) = C$ - quadratic surface



$\Sigma$ - Positive semidefinite

$\Downarrow$

ellipsoid

Principal axes: eigenvectors of $\Sigma$

Length: $\sqrt{l_i}$, $l$ - eigenvalues of $\Sigma$

# Whitening Transform

Define:

$$\Lambda = diag\left(eigval\left(\Sigma\right)\right)$$ - Scaling matrix

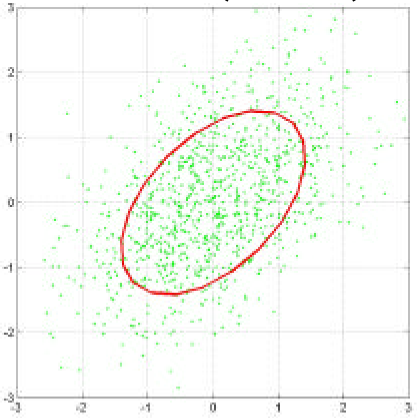
$$\Phi = eigvec\left(\Sigma\right)$$ - Rotation matrix

Then:

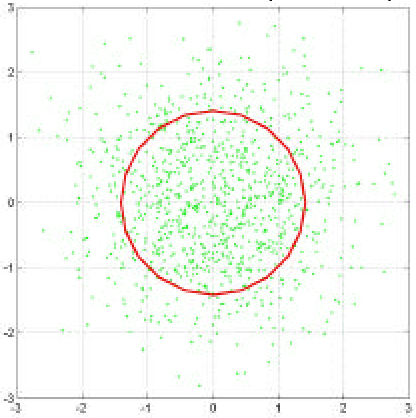
$$W = \Lambda^{-1/2}\Phi^T$$ -“Unscales” and “unrotates” the data

For all:

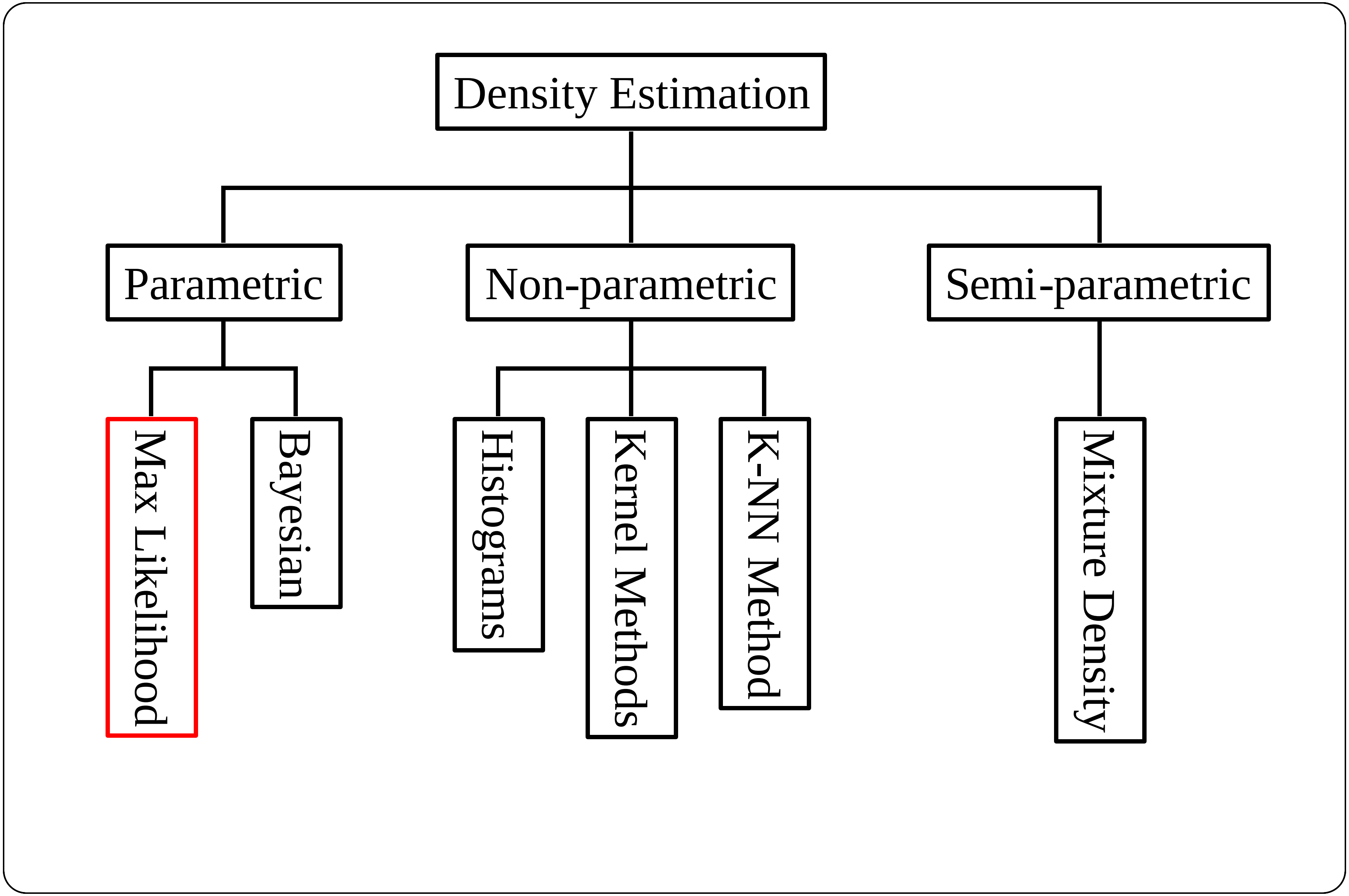
$$x \sim N\left(\mathbf{0}, \Sigma\right)$$



$$Wx \sim N\left(\mathbf{0}, I\right)$$



# You Are Here


Density Estimation
Parametric
Non-parametric
Semi-parametric
Max Likelihood
Bayesian
Histograms
Kernel Methods
K-NN Method
Mixture Density

# Maximum Likelihood

Parameters are fixed but unknown.

$$D \equiv \left\{ x^1, x^2, \ldots, x^N \right\}$$ - a data set, drawn from $p(x)$

Notationally, we make density explicitly dependent on parameters:

$$p(x) \Longrightarrow p(x \mid q)$$

Assuming that the data is drawn independently (i.i.d.):

$$L(q) \equiv p(D \mid q) = \prod_{n=1}^{N} p(x^n \mid q)$$ - a *likelihood function*

To find θ Maximize $L(\theta)$ w.r.t. parameters.

# Maximum Likelihood

Maximizing $L(\theta)$ is equivalent to maximizing *log-likelihood function*:

$$l(q) \equiv \log L(q) = \log \prod_{n=1}^{N} p(x^n \mid q) = \sum_{n=1}^{N} \log p(x^n \mid q)$$

To find θ set the derivative to 0:

$$\nabla_q l(q) = \sum_{n=1}^{N} \nabla_q \log p(x^n \mid q) = 0$$

And solve for θ

# Quick Summary – ML Parameter Estimation

$$P(w_i \mid x) = P(w_i \mid x, q_i) = \frac{\boxed{p(x \mid w_i, q_i)}\, P(w_i \mid q_i)}{p(x \mid q_i)}$$

easy

$$\boxed{p(x \mid \boxed{\hat{q}})}$$

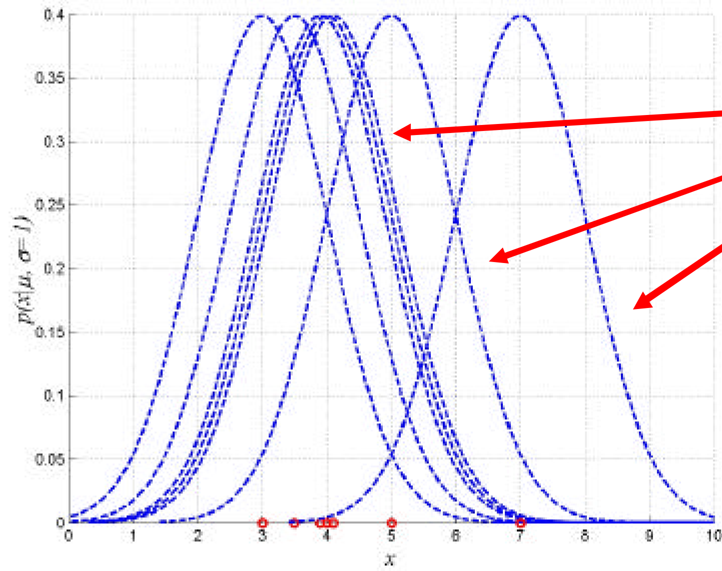
we pick that

$$\boxed{\operatorname*{argmax}_{q} \boxed{p(D \mid q)}}$$

$$\boxed{\prod_{n=1}^{N} p(x^n \mid q)}$$

# Solving a Maximum Likelihood Problem

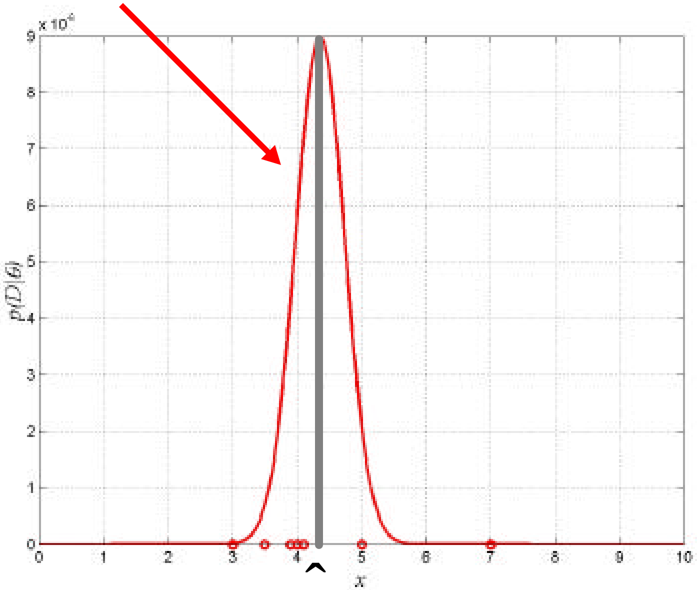
Fixed covariance:



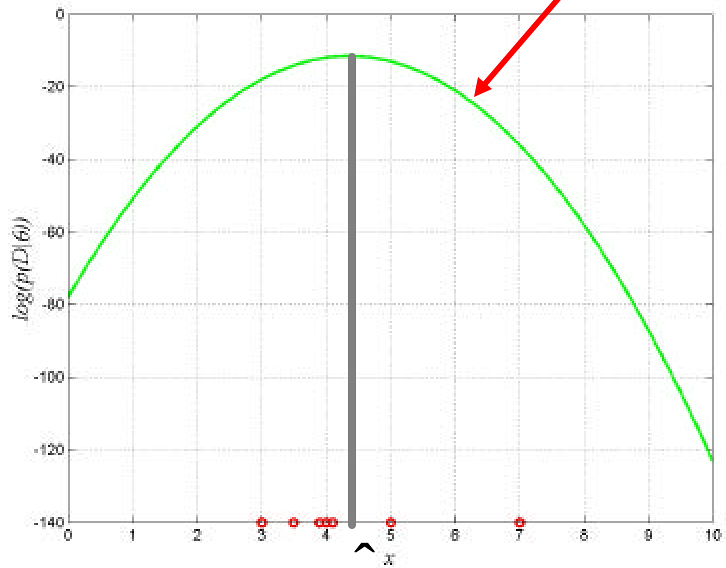

some candidates

$p(D\,|\,q)$

log $p(D\,|\,q)$




$\hat{q}$




$\hat{q}$

# Maximum Likelihood Example

In d-dimensions:

$$\nabla_q l(q) = \sum_n \nabla_q \left\{ -\frac{d}{2}\log\left[2p\right] - \frac{1}{2}\log\left[\left|\Sigma\right|\right] - \frac{1}{2}(x^n - m)^T \Sigma^{-1}(x^n - m) \right\}$$

Solving for the mean:

$$\nabla_m l(q) = -\frac{1}{2}\sum_n \Sigma^{-1}(x^n - \hat{m}) = 0 \;\Rightarrow$$

$$\boxed{\hat{m} = \frac{1}{N}\sum_{n=1}^{N} x^n}$$ - arithmetic average of samples

# Maximum Likelihood Example (cont.)

$$\nabla_q l(q) = \sum_n \nabla_q \left\{ -\frac{d}{2}\log\left[2p\right] - \frac{1}{2}\log\left[\left|\Sigma\right|\right] - \frac{1}{2}(x^n - m)^T \Sigma^{-1}(x^n - m) \right\}$$

Solving for the covariance:

For symmetric $M$: $\quad \dfrac{d\left|M\right|}{dM} = \left|M\right| M^{-1}$ and $\dfrac{d\left(a^T M^{-1} b\right)}{dM} = M^{-1} a b^T M^{-1} \quad \Rightarrow$

$$\nabla_\Sigma l(q) = -\frac{1}{2}\sum_n \left\{ \hat{\Sigma}^{-1} - \hat{\Sigma}^{-1}(x^n - \hat{m})(x^n - \hat{m})^T \hat{\Sigma}^{-1} \right\} = 0 \;\Rightarrow$$

biased

$$\boxed{\hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}\left(x^n - \hat{m}\right)\left(x^n - \hat{m}\right)^T}$$

-arithmetic average of indiv. covariances

# Recursive ML

What if data comes one sample at a time?

$$\boxed{\hat{m}_N} = \frac{1}{N}\sum_{n=1}^{N} x^n = \frac{1}{N}\left[x^N + \sum_{n=1}^{N-1} x^n\right]$$

$$= \frac{1}{N}\left[x^N + (N-1)\hat{m}_{N-1}\right] = \boxed{\hat{m}_{N-1} + \frac{1}{N}\left[x^N - \hat{m}_{N-1}\right]}$$

This estimate "stiffens" with more data (as it should).

One idea – fix the fraction. Then the estimate can track a non-stationary process

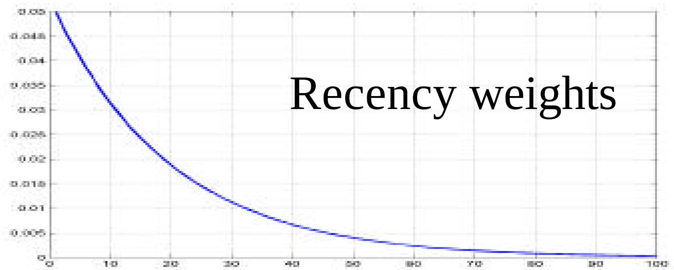
# Recursive ML

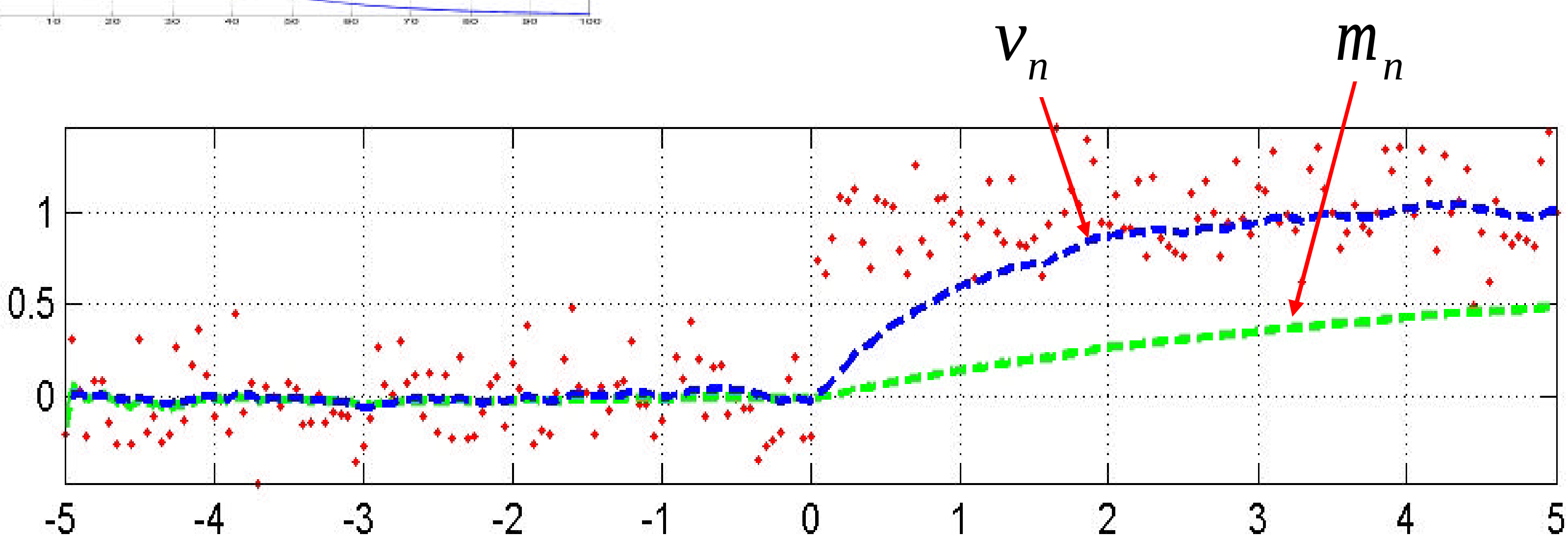
Fix the update rate and retrace the steps:

$$v_N = v_{N-1} + g\left[x^N - v_{N-1}\right] = (1-g)v_{N-1} + g\,x^N$$

$$= (1-g)^2 v_{N-2} + (1-g)g\,x^{N-1} + g\,x^N$$

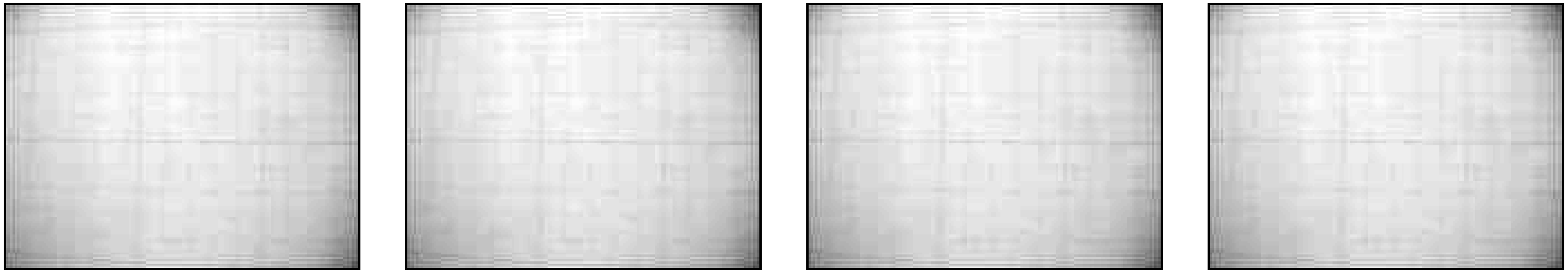
$$= (1-g)^M v_{N-M} + \sum_{k=1}^{M} (1-g)^{M-k} g\,x^k$$



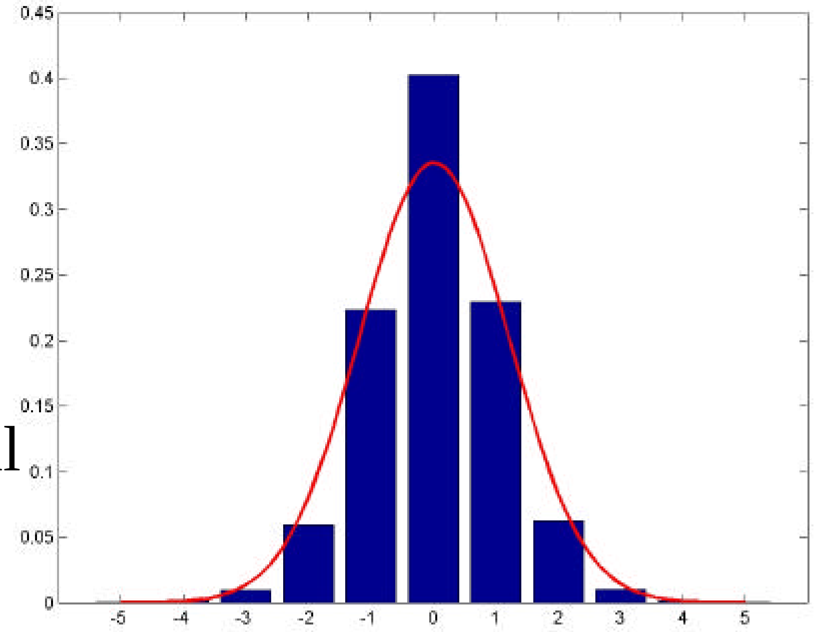
# Simple Example

Several images from a static camera:



How much noise is in it?

$$x = vec\left(I_t - I_{t-1}\right)$$
$$m = 0$$
$$s = 1.2$$

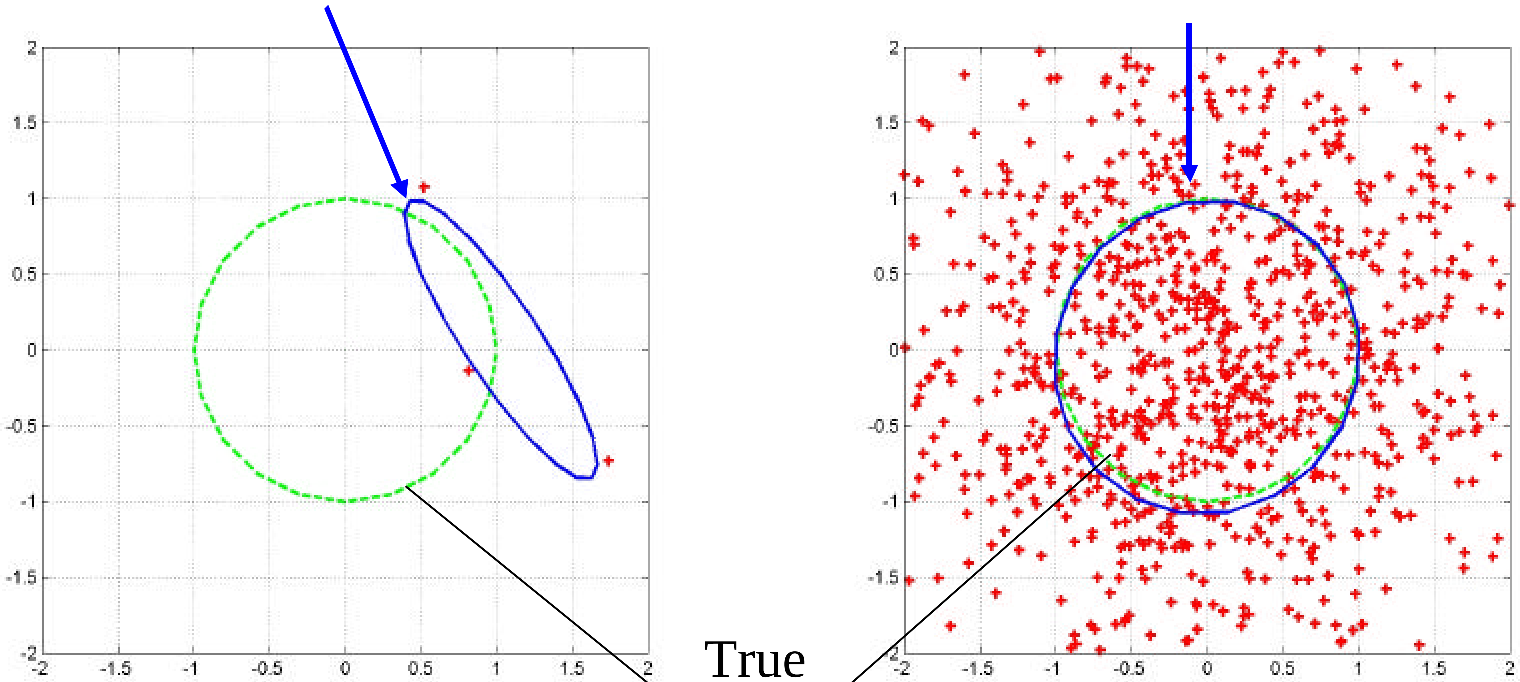



Now we can set a threshold that will *statistically* distinguish pixel noise from an object

# Problems with ML

We are given two estimates:

$\mu_1, \Sigma_1$ $\qquad\qquad$ $\mu_2, \Sigma_2$

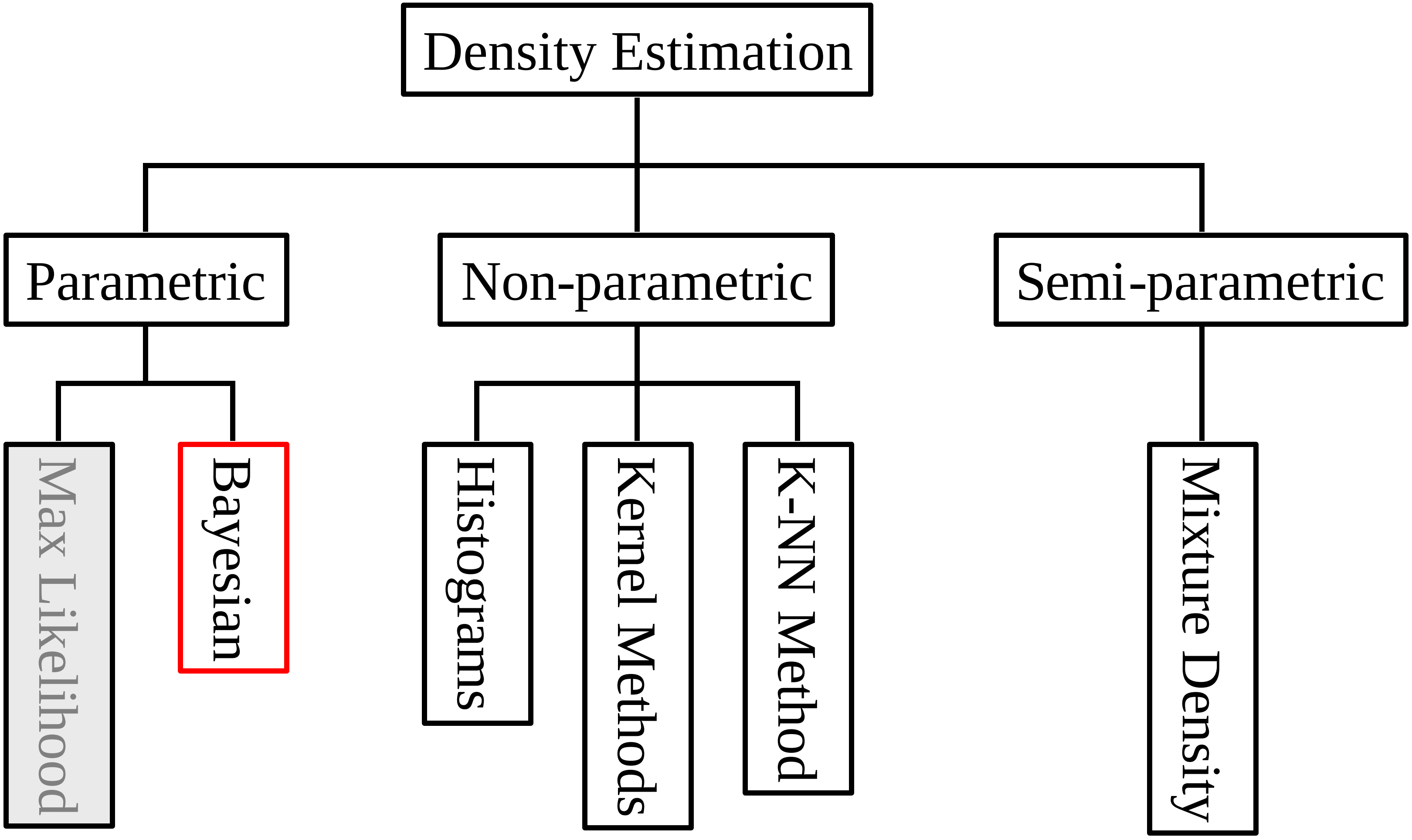


True distribution

Which one do we believe?

ML gives a single solution, regardless of uncertainty.

# You Are Here


Density Estimation
Parametric
Non-parametric
Semi-parametric
Max Likelihood
Bayesian
Histograms
Kernel Methods
K-NN Method
Mixture Density

# Bayesian Parameter Estimation

In classification our goal so far has been to estimate $P(w \mid x)$

Let's make the dependency on the *data* explicit:

$$P(w_i \mid x, D) = \frac{p(x \mid w_i, D) P(w_i \mid D)}{p(x \mid D)}$$

$P(w_i \mid D)$ - this is easy to compute

$P(x \mid D)$ - this is easy to compute by marginalization

What about $p(x \mid w_i, D)$ ?

# Bayesian Parameter Estimation

$$P(w_i \mid x, D) = \frac{p(x \mid w_i, D)P(w_i \mid D)}{p(x \mid D)}$$

This is a supervised problem so far:

$$D = \{D_1, D_2, \ldots, D_N\}$$

$$p(x \mid w_i, D) = p\left(x \mid w_i, \{D_j\}_{j=1\ldots N}\right)$$

$$= p\left(x \mid w_i, D_i, \{\cancel{D_j}\}_{j \neq i}\right) = p\left(x \mid w_i, D_i\right)$$

$$\Downarrow$$

$$P(w_i \mid x, D) = \frac{\boxed{p(x \mid w_i, D_i)}P(w_i \mid D)}{p(x \mid D)}$$

# Bayesian Parameter Estimation

We will assume that we can obtain "labeled" data, so again:

Notationally:

$$p(x\,|\,\cancel{w_i}, D_i) \Longrightarrow p(x\,|\,D)$$

Now our problem is to compute density for *x* given the data *D*.

We assume the form of $p(x)$ – the source density for *D*:

$$p(x) \Longrightarrow p(x\,|\,q)$$

… and treat θ as a *random variable*

# Bayesian Parameter Estimation

Instead of choosing a value for a parameter, we use them all:

$$p(x \mid D) = \int p(x, q \mid D)\, dq = \int p(x \mid q, \cancel{D})\, p(q \mid D)\, dq$$

*Data predicts the new sample*

*x is independent of D given q*

$$= \int \underbrace{p(x \mid q)}_{\text{We chose the form of this}} \underbrace{p(q \mid D)}_{\text{What is this?}} dq$$

Average densities $p(x|q)$ for ALL possible values of $q$ weighted by its posterior probability

$$\int p(x \mid q) p(q \mid D) dq$$

Computing the posterior probability for $q$ :

Using Bayes rule:

*What is this?*

*Prior belief about the parameters (<u>denisty</u>)*

$$p(q \mid D) = \frac{p(D \mid q)\, p(q)}{\int p(D \mid q)\, p(q) dq}$$

Using independence:

$$p(D \mid q) = \prod_{n=1}^{N} p(x^n \mid q)$$

Bayesian method does not commit to a particular value of $\theta$, but uses the <u>*entire distribution*</u>.

# Quick Summary – Bayesian Parameter Estimation

$$P(w_i \mid x) = P(w_i \mid x, D) = \frac{p(x \mid w_i, D_i)\, P(w_i \mid D)}{p(x \mid D)}$$

easy

hard

$$\int p(x \mid q)\, p(q \mid D)\, dq$$

we pick that

$$\frac{p(D \mid q)\, p(q)}{p(D)}$$

we “know” this*, **

hard

$$\prod_{n=1}^{N} p(x^n \mid q)$$

* *Non-informative* prior – doesn’t introduce bias

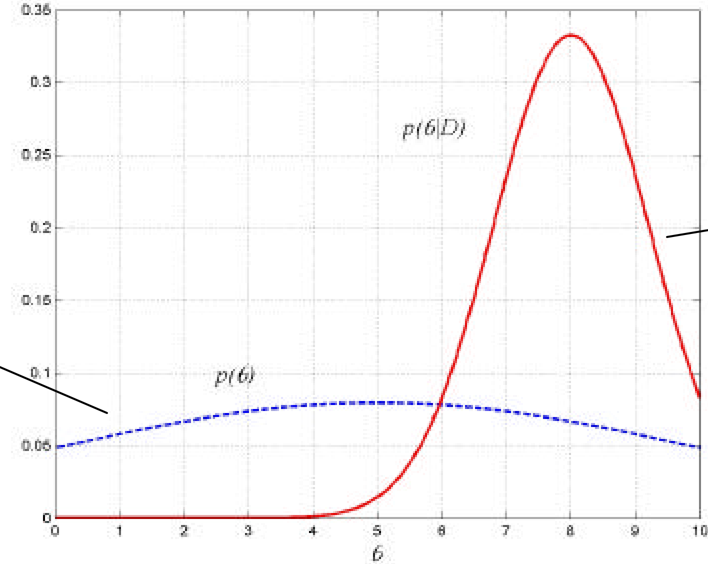
** *Conjugate prior* – causes $p(q \mid D)$ have the same functional form as $p(D \mid q)$

# Bayesian Parameter Estimation

$$\int p(x \mid m)\, p(m \mid D)\, dm$$

For $q = m$ :

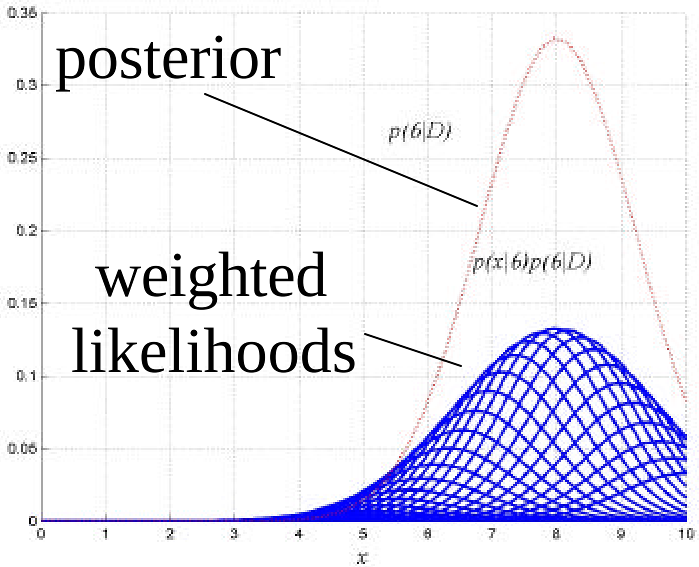
Parameter prior




Parameter posterior

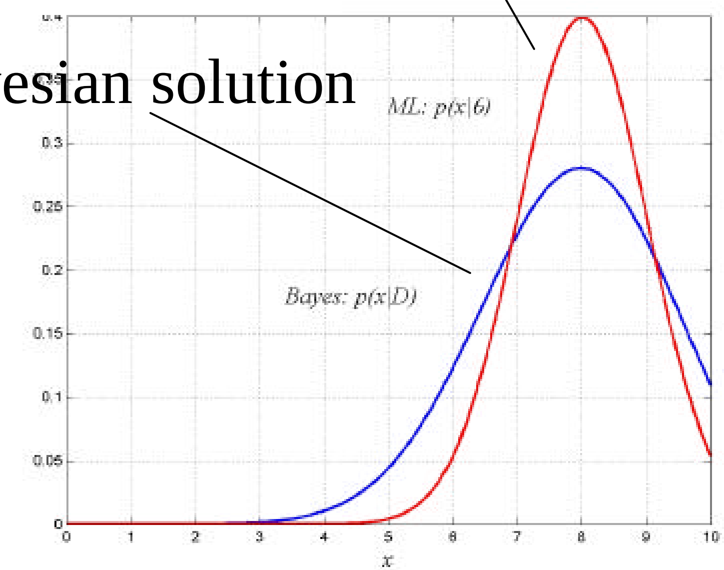
posterior

weighted likelihoods




ML solution

Bayesian solution



# Bayesian Parameter Estimation - Example

$\int p(x \mid m)\, p(m \mid D)\, dm$

First let's deal with the parameter:

Likelihood: $p(x \mid m) = \mathbf{N}(m, s^2)$

*fixed*

Parameter prior: $p(m) = \mathbf{N}(m_0, s_0^2)$

Need to find: $p(m \mid D)$

Bayes rule again:

$$p_N(m \mid D) = \frac{p(D \mid m)\, p(m)}{p(D)} = \underbrace{a \left[ \prod_{n=1}^{N} p(x^n \mid m) \right] \mathbf{N}(m_0, s_0^2)}_{\textit{This is a Gaussian}} = \mathbf{N}(m_N, s_N)$$

*N-sample parameter posterior*

*Need these*

# Bayesian Parameter Estimation - Example

So, the posterior is a Gaussian

$$p_N(m \mid D) = \mathbf{N}(m_N, s_N)$$

After some algebra and identifying the terms:

$$\frac{1}{s_N^2} = \frac{1}{s^2} N + \frac{1}{s_0^2} \qquad \text{- } \textit{when Gaussians multiply – precisions add}$$

… and

$$m_N = \frac{N s_0^2}{N s_0^2 + s^2} \bar{x} + \frac{s^2}{N s_0^2 + s^2} m_0$$

With increasing *N* covariance of the posterior decreases and the prior becomes unimportant.

# Bayesian Parameter Estimation - Example

Now the integral:

$$p(x \mid D) = \int p(x \mid q)\, p(q \mid D)\, dq$$

$$= \int \mathbf{N}(m, s^{2})\, \mathbf{N}(m_N, s_N^{2})\, dm = \mathbf{N}(m_N, s^{2} + s_N^{2})$$

*You can show that it is also a Gaussian*

Any guesses about why Gaussian is such a common assumption?

# Recursive Bayes

$$\int p(x\,|\,q)\,p(q\,|\,D)\,dq$$

For $N$-point likelihood:

$$p(D^N\,|\,q) = \prod_{n=1}^{N} p(x^n\,|\,q)$$

$$= p(x^N\,|\,q)\prod_{n=1}^{N-1} p(x^n\,|\,q) = \boxed{p(x^N\,|\,q)\,p(D^{N-1}\,|\,q)}$$

From this the recursive relation for the posterior:

$$p(q\,|\,D^N) = \frac{\boxed{p(x^N\,|\,q)\,p(D^{N-1}\,|\,q)}\,p(q)}{p(D^N)}$$

$$= \frac{p(x^N\,|\,q)\,p(q\,|\,D^{N-1})}{\int p(x^N\,|\,q)\,p(q\,|\,D^{N-1})\,dq}$$

# Recursive Bayes (cont.)

Again:

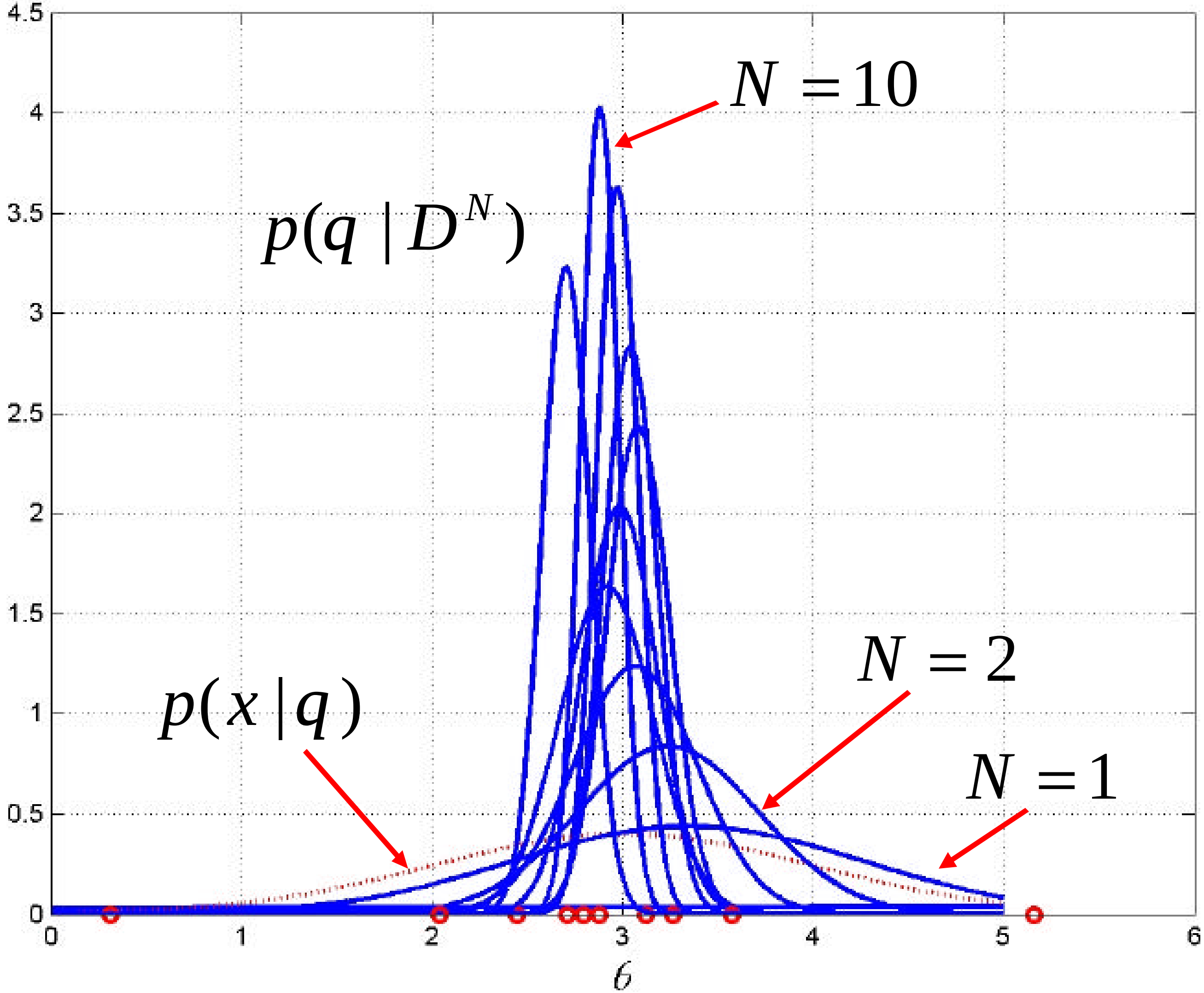
$$p(q \mid D^N) = \frac{p(x^N \mid q)\, p(q \mid D^{N-1})}{\int p(x^N \mid q)\, p(q \mid D^{N-1})\, dq}$$ - 1-point update.

Setting *N=1*:

$$\frac{1}{s_n^2} = \frac{1}{s^2} + \frac{1}{s_{n-1}^2}$$

$$m_n = \frac{s_{n-1}^2}{s_{n-1}^2 + s^2} x + \frac{s^2}{s_{n-1}^2 + s^2} m_{n-1}$$

# Recursive Bayes (cont.)



# Problems with Bayesian Method

1. Integration is difficult
2. Analytic solutions are only available for restricted class of densities
3. Technicality: If the true $p(x|q)$ is NOT what we assume it is, the prior probability of any parameter setting is *0*!
4. Integration is difficult
5. Did I mention that the integration is hard?

# Relation between Bayesian and ML Inference

$$p(q \mid D) \propto p(D \mid q)\, p(q)$$

peaks at $\hat{q}_{ML}$

$$= \left[ \prod_n p(x^n \mid q) \right] p(q) = L(q)\, p(q)$$

If the peak is sharp and $p(\theta)$ is flat, then:

$$p(x \mid D) = \int p(x \mid q)\, p(q \mid D)\, dq$$

$$\simeq \int p(x \mid \hat{q})\, p(q \mid D)\, dq = p(x \mid \hat{q}) \int p(q \mid D)\, dq = p(x \mid \hat{q})$$

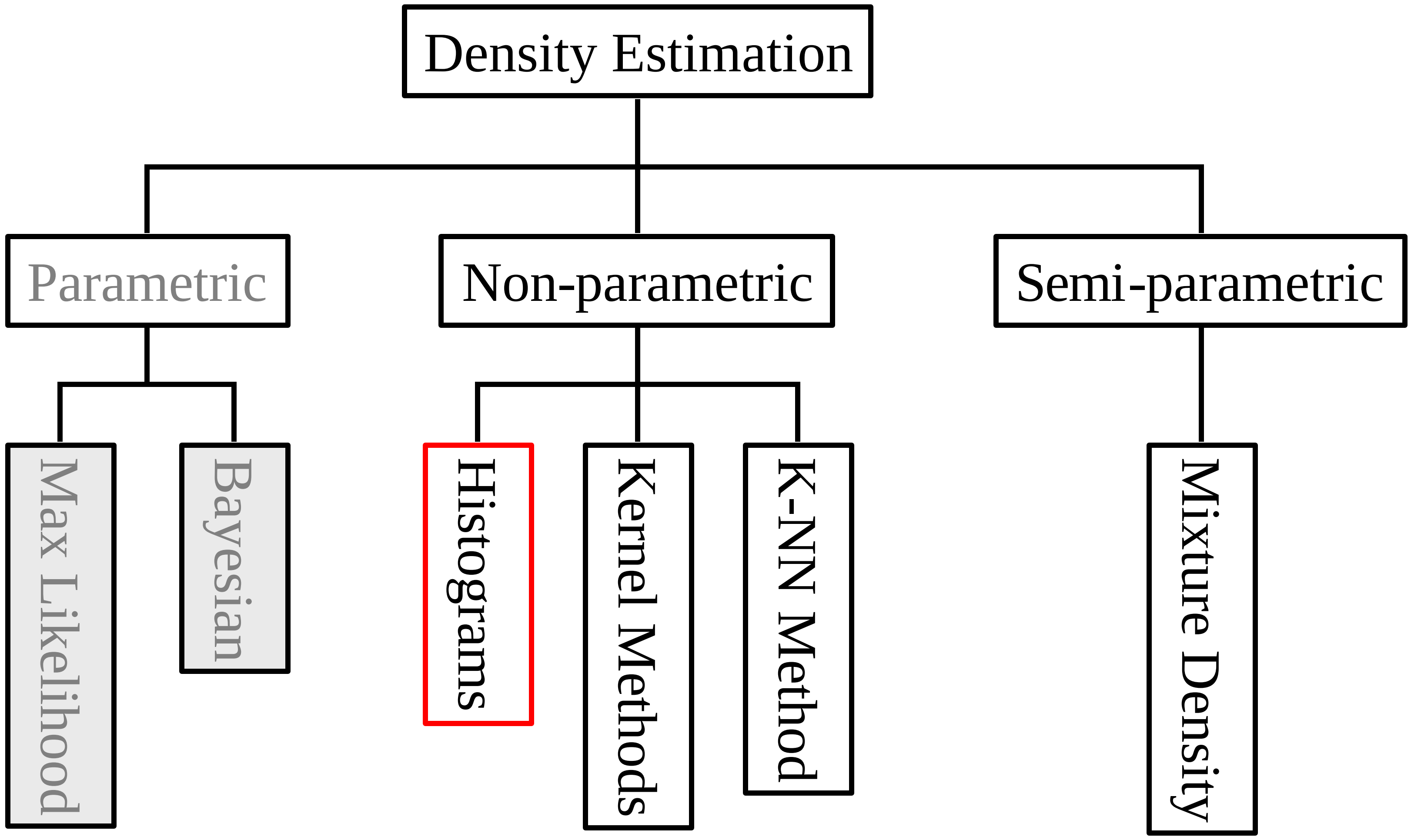
$$\text{As } N \to \infty,\ p(x \mid D) \leftrightarrow p(x \mid \hat{q})$$

# Non-Parametric Methods for Density Estimation

Non-parametric methods do not assume any particular form for $p(x)$

1. Histograms
2. Kernel Methods
3. K-NN method

# You Are Here


Density Estimation
Parametric
Non-parametric
Semi-parametric
Max Likelihood
Bayesian
Histograms
Kernel Methods
K-NN Method
Mixture Density

$\hat{P}(x)$ is a discrete approximation of $p(x)$

- Count a number of times that $x$ lands in the $i$-th bin

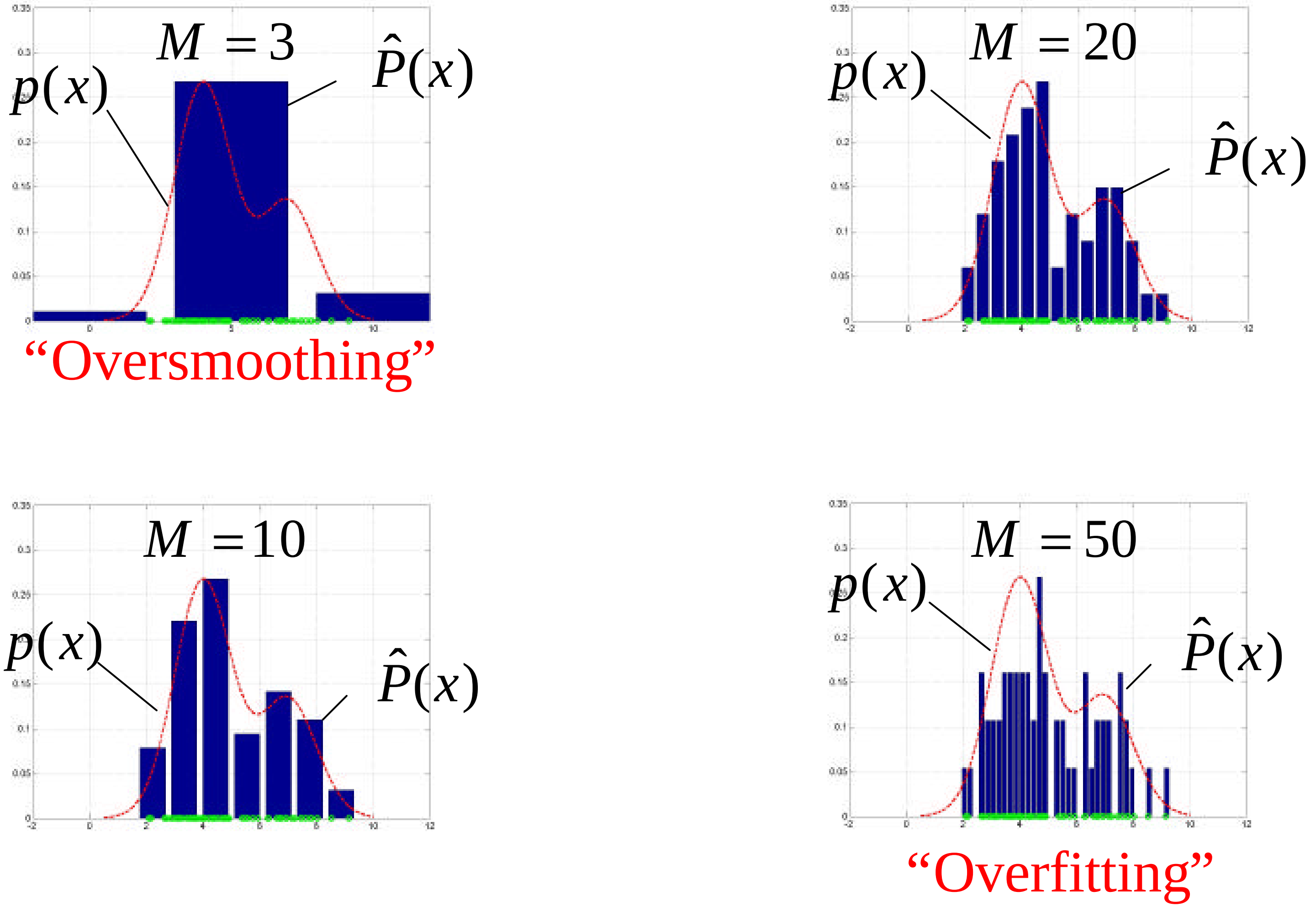
$$H(i) = \sum_{j=1}^{N} I(x \in R_i), \quad \forall i = 1, 2, \ldots, M$$

- Normalize

$$\hat{P}(i) = \frac{H(i)}{\sum_{j=1}^{M} H(j)}$$

# Histograms

How many bins?



# Histograms

Good:

- Once it is constructed, the data can be discarded
- Quick and intuitive

Bad:

- Very sensitive to number of bins, $M$
- Estimated density is not smooth
- Poor generalization in higher dimensions

# Aside: Curse of dimensionality (Bellman, '61):

- Imagine we build a histogram of a 1-d feature (say, *Hue*)
  - 10 bins
  - 1 bin = 10% of the input space
  - need at least 10 points to populate every bin
- We add another feature (say, *Saturation*)
  - 10 bins again
  - 1 bin = 1% of the input space
  - we need at least 100 points to populate every bin
- We add another feature (say, *Value*)
  - 10 bins again
  - 1 bin = 0.1% of the input space
  - we need at least 1000 points to populate every bin

$\boxed{N = b^d}$ - number of points grows exponentially

# Aside: Curse Continues

Volume of a cube in $R^d$ with side *l*:

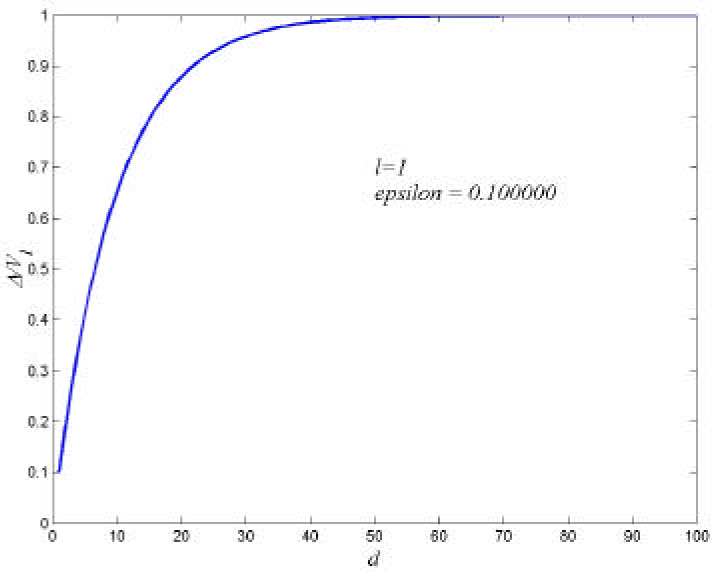
$$V_l = l^d$$

Volume of a cube with side *l*-ε:

$$V_e = (l - e)^d$$

Volume of the ε-shell:

$$\Delta = V_l - V_e = l^d - (l - e)^d$$




Ratio of the volume of the ε-shell to the volume of the cube:

$$\frac{\Delta}{V_l} = \frac{l^d - (l-e)^d}{l^d} = 1 - \left(1 - \frac{e}{l}\right)^d \rightarrow 1 \; as \; d \rightarrow \infty \;\; !!!!!$$

# Aside: Lessons of the curse

In generative models:

- Use as much data as you can get your hands on
- Reduce dimensionality as much as you can get away with

<End of Digression>

# General Reasoning

By definintion:

$$P(x \in R) = P = \int_R p(x')dx'$$

If we have $N$ i.i.d. points drawn form $p(x)$:

$$P(|x \in R| = k) = \underbrace{\frac{N!}{k!(N-k)!}}_{\text{Num. of unique splits K vs. (N-K)}} \underbrace{P^k}_{\text{Prob that } k \text{ of particular } x\text{-es are in R}} \underbrace{(1-P)^{N-k}}_{\text{Prob that the rest are not}} = B(N, P)$$

$B(N, P)$ is a *binomial* distribution of $k$

# General Reasoning (cont.)

Mean and variance of $B(N, P)$:

Mean: $$m = E[k] = NP \quad \Rightarrow P = E[k / N]$$

Variance: $$s^2 = E[(k-m)^2] = NP(1-P)$$

$$\Rightarrow E\left[\left(k / N - P\right)^2\right] = \left(\frac{s}{N}\right)^2 = P(1-P) / N$$

That is:

- $E[k/N]$ is a good estimate of $P$
- $P$ is distributed around this estimate with vanishing variance

So:

$$P \simeq k / N$$

# General Reasoning (cont.)

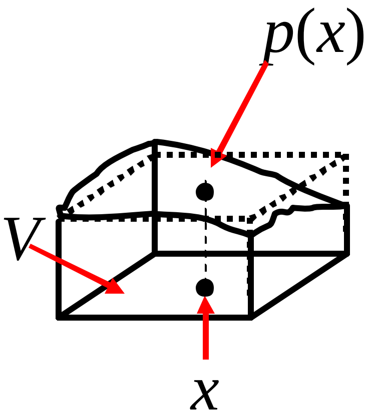
So:

$$P \simeq k / N$$

On the other hand, under mild assumptions:

$$P = \int_R p(x')dx' \simeq p(x)V$$

Volume of R (not $p(x)$)




… which leads to:

$$\boxed{p(x) \simeq \frac{k}{NV}}$$

# General Reasoning (cont.)

Now, given *N* data points – how do we really estimate $p(x)$?

$$\boxed{p(x) \simeq \frac{k}{NV}}$$

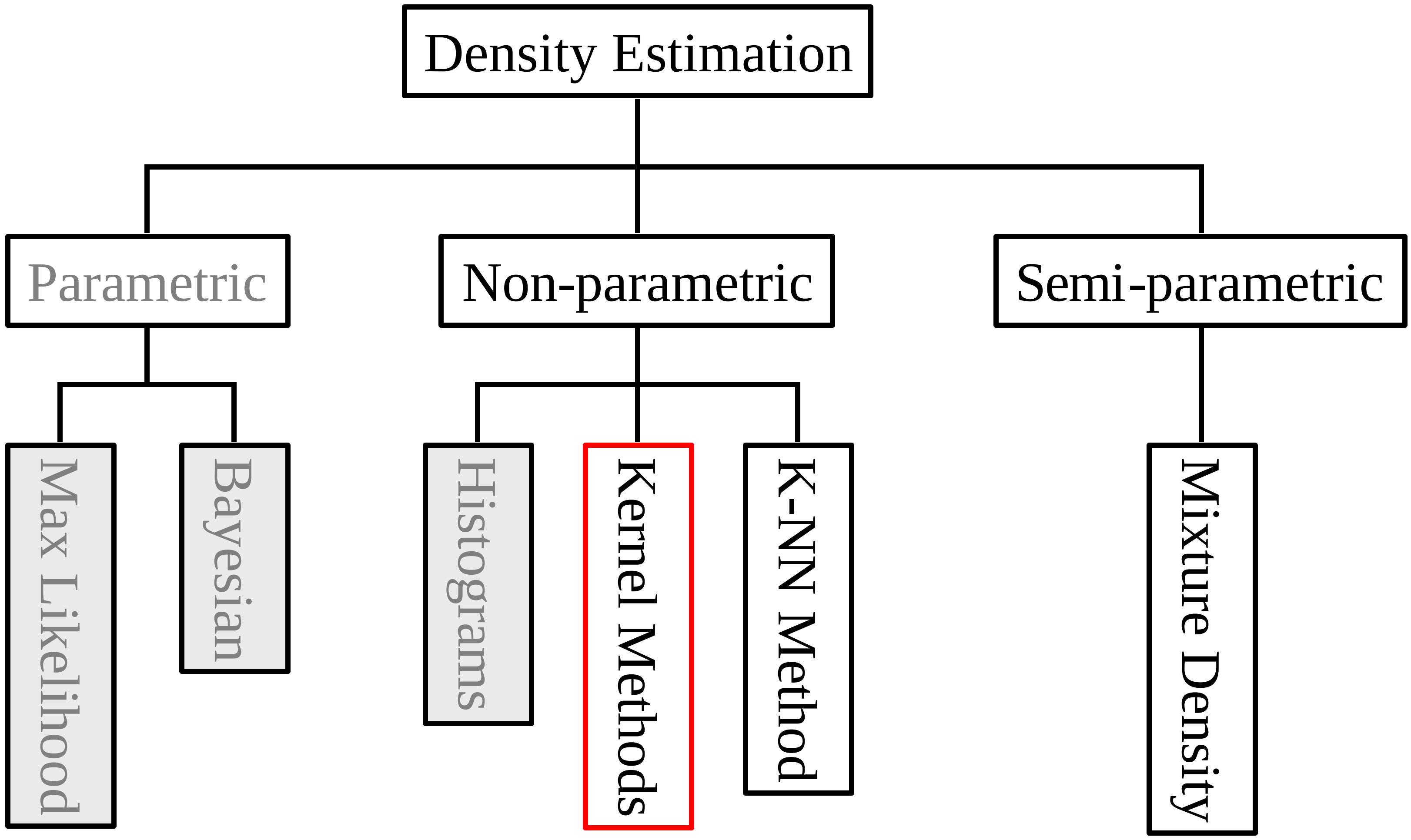
Fix *k* and vary *V* until it encloses *k* points

Fix *V* and count how many points (k) it encloses

*K-Nearest Neighbors* (*KNN*)

*Kernel* methods

# You Are Here



Density Estimation
Parametric
Non-parametric
Semi-parametric
Max Likelihood
Bayesian
Histograms
Kernel Methods
K-NN Method
Mixture Density

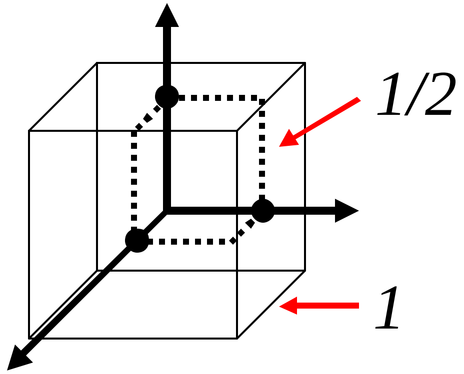
# Kernel Methods of Density Estimation

We choose $V$ by specifying a hypercube with a side $h$:

$$V = h^d$$

Mathematically:

$$H(\mathbf{y}) = \begin{cases} 1 & |y_j| < 1/2 \quad j = 1,\ldots,d \\ 0 & \text{otherwise} \end{cases}$$




*kernel function*:

$$H(\mathbf{y}) \geq 0, \quad \forall \mathbf{y} \quad \text{and} \quad \int H(\mathbf{y}) d\mathbf{y} = 1$$

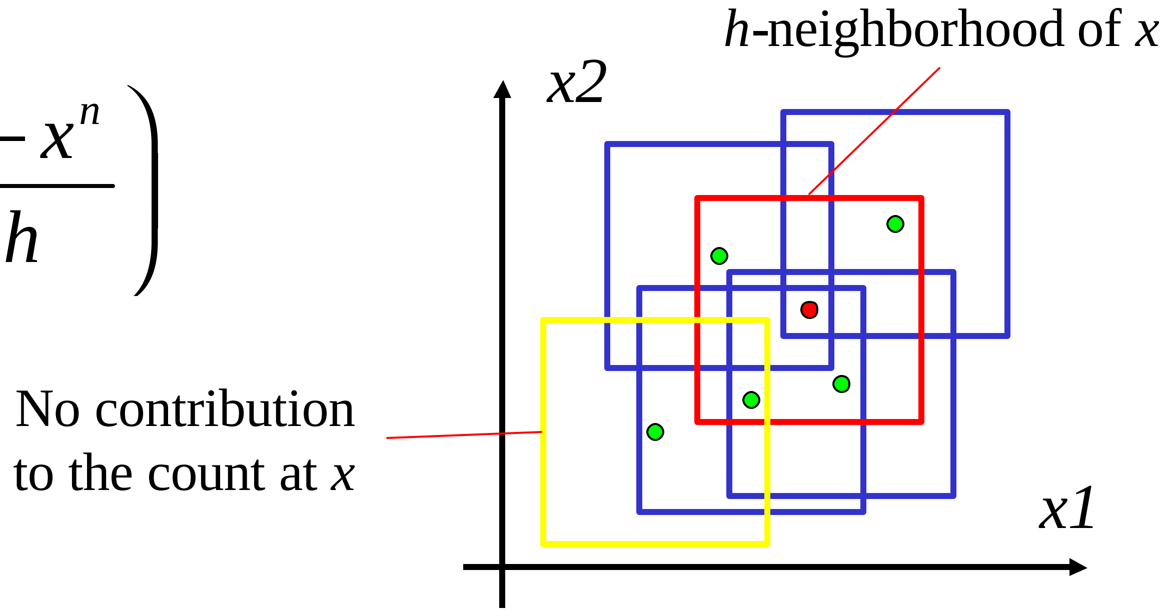
# Parzen Windows

Then

$H\left(\left(\mathbf{x}-\mathbf{x}^{n}\right)/h\right)$ - a hypercube with side $h$ centered at $\boldsymbol{x}^n$

$H$ can help count the points in a volume $V$ around any $x$:

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



# Rectangular Kernel

So the number of points in h-neighborhood of $x$:

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

… is easily converted to the density estimate:

$$\tilde{p}(x) = \frac{k(x)}{NV} = \frac{1}{N}\sum_{n=1}^{N} \boxed{\frac{1}{h^d} H\left(\frac{x - x^n}{h}\right)}$$

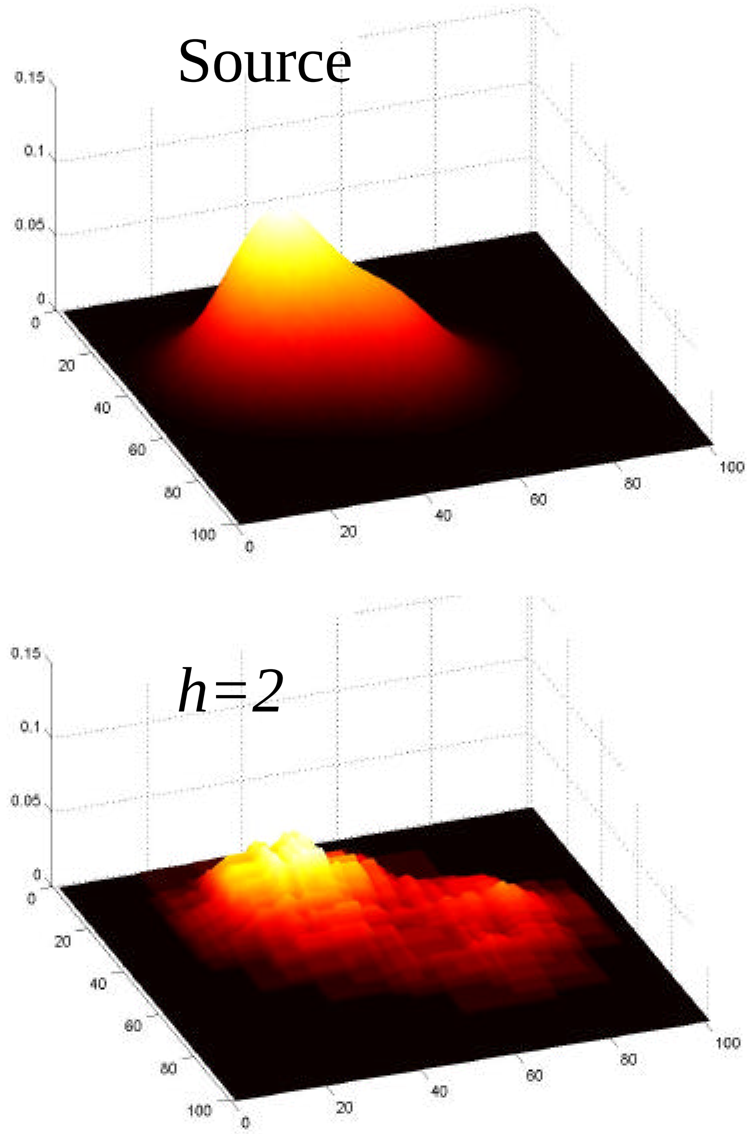
$K\left(x, x^n\right)$
Integrates to 1

Subtle point:

$$\int \left[\frac{1}{N}\sum_{n=1}^{N} K\left(x, x^n\right)\right] dx = \frac{1}{N}\sum_{n=1}^{N}\left[\int K\left(x, x^n\right) dx\right] = 1$$

$$\Rightarrow \int \tilde{p}(x)\, dx = 1$$

# Example



# Smoothed Window Functions

The problem is as in histograms – it is discontinuous

We can choose a smoother function, s.t.:

$$\tilde{p}(x) \geq 0, \quad \forall x \quad \text{and} \quad \int \tilde{p}(x)\,dx = 1$$

*Ensured by kernel conditions*

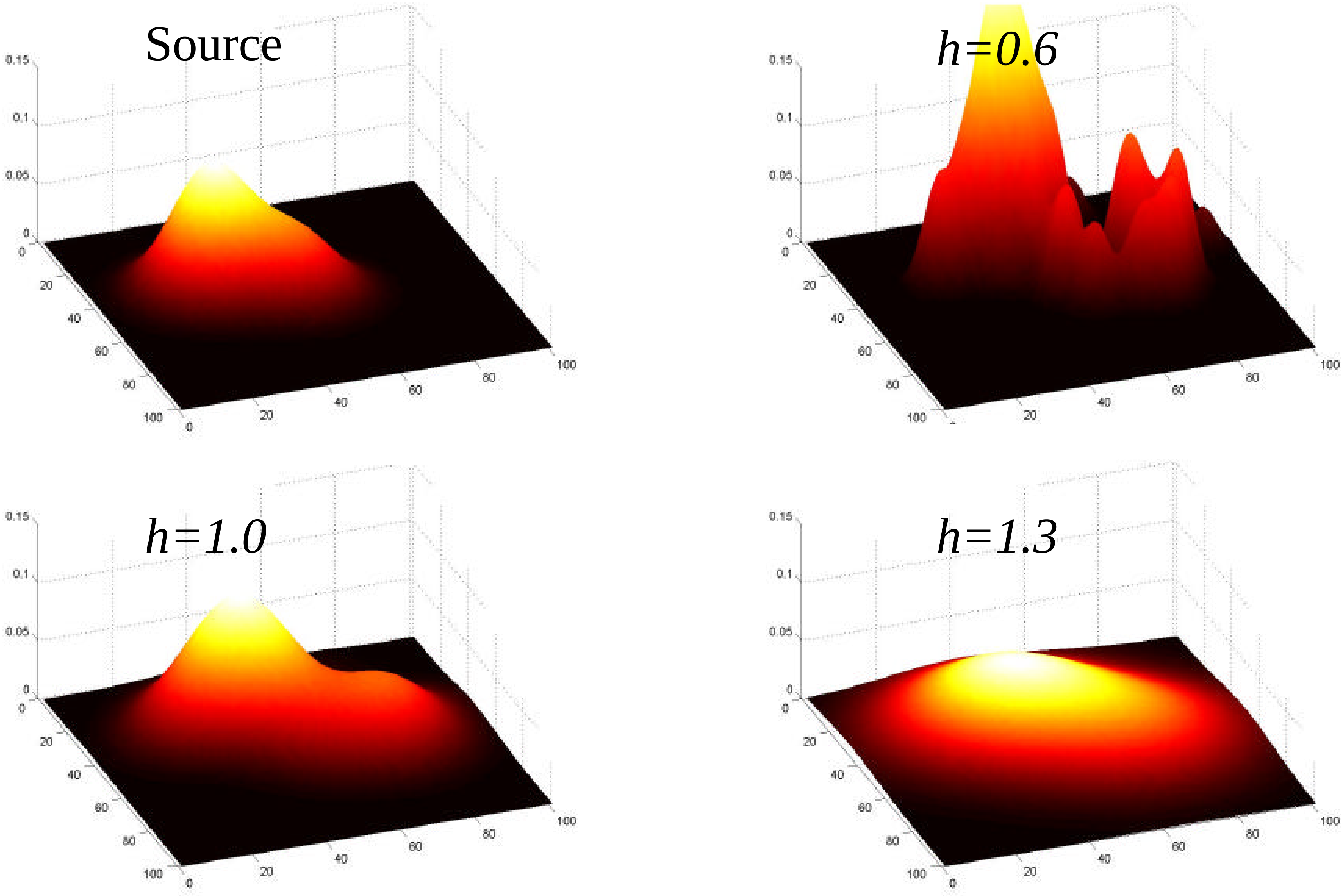
Eg: <loud cheer> a (spherical) Gaussian:

$$K(x, x^n) = \frac{1}{(\sqrt{2\pi}\,h)^d} \exp\left( -\frac{\left\| x - x^n \right\|}{2h^2} \right)$$

… so:

$$\tilde{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(\sqrt{2\pi}\,h)^d} \exp\left( -\frac{\left\| x - x^n \right\|}{2h^2} \right)$$

# Example



# Some Insight

Interesting to look at expectation of the estimate with respect to all possible datasets:

$$E\left[\tilde{p}(x)\right] = E\left[\frac{1}{N}\sum_{n=1}^{N} K(x, x^n)\right] = E\left[K(x, x')\right]$$

$$= \int K(x - x')p(x')dx'$$ - *convolution with true density*

That is:

$$\tilde{p}(x) = p(x) \ \text{ if } \ K(x, x') = d(x, x')$$

But not for the finite data set!

# Conditions for Convergence

How small can we make $h$ for a given $N$ ?

$$\lim_{N\to\infty} h_N^d = 0$$

- It should go to *0*

$$\lim_{N\to\infty} Nh_N^d = \infty$$

- But slower than *1/N*

Based on the similar analysis of variance of estimates

Eg:

$$h_N^d = h_1^d / \sqrt{N}$$

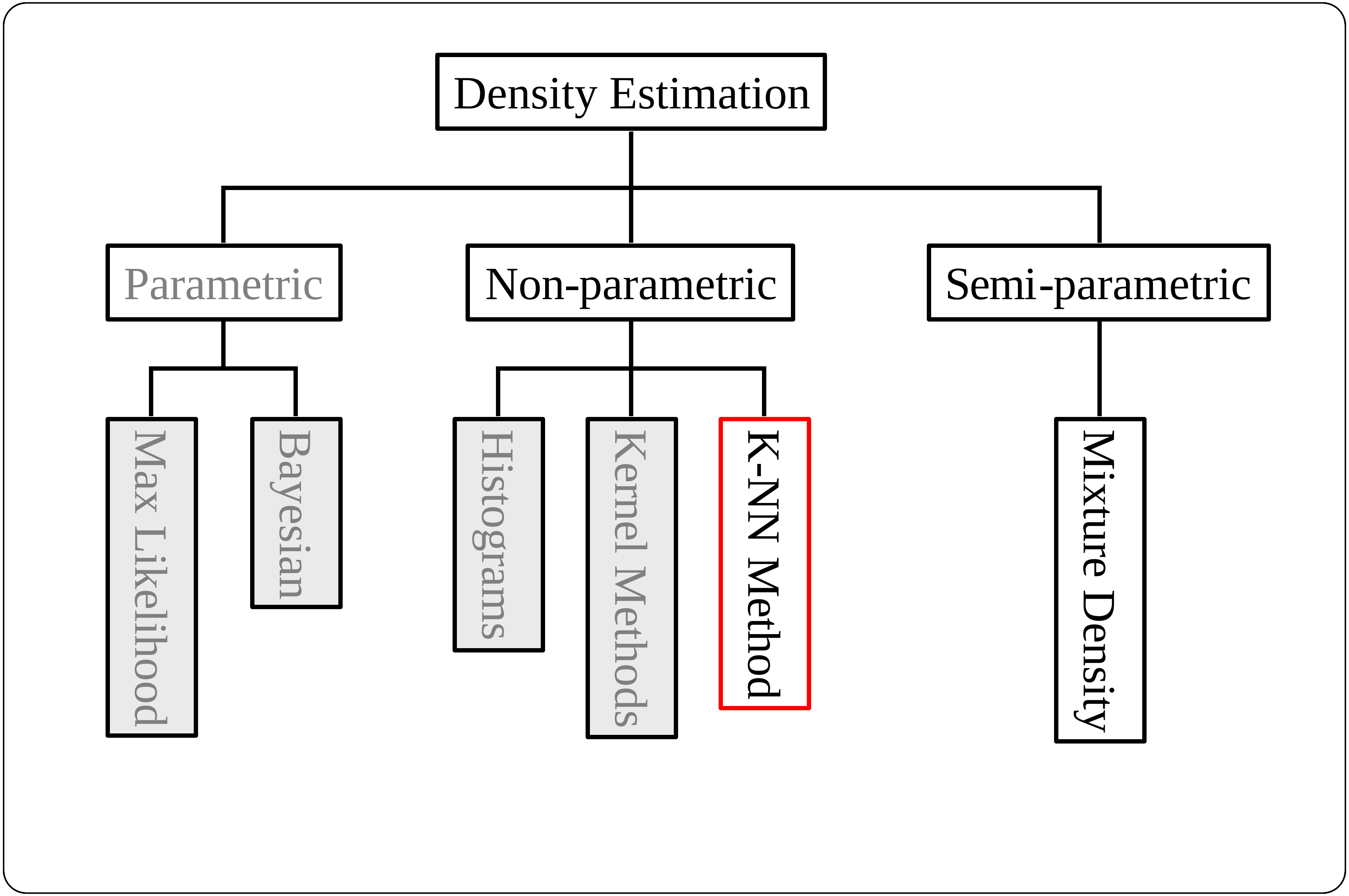
$$h_N^d = h_1^d / \log(N)$$

Note that the choice of $h_1^d$ is still up to us.

# Problems With Kernel Estimation

- Need to choose the width parameter, $h$
  - Can be chosen empirically
  - Can be adaptive, eg. $h_j = hd_{jk}$ – where $d_{jk}$ the distance from $x_j$ to $k$-th nearest neighbor
- Need to store all data to represent the density
  - Leads to Mixture Density Estimation

# You Are Here



Density Estimation
Parametric
Non-parametric
Semi-parametric
Max Likelihood
Bayesian
Histograms
Kernel Methods
K-NN Method
Mixture Density

# K-Nearest Neighbors

Recall that:

$$\tilde{p}(x) = \frac{k}{NV}$$

Now we fix $k$ (typically $k = \sqrt{N}$ ) and expand $V$ to contain $k$ points

This is not a true density!

Eg.: choose $N=1$, $k=1$. Then:

$$\tilde{p}(x) = \frac{1}{1 \cdot \left\| x - x_1 \right\|}$$

Oops!

BUT it is useful for a number of theoretical and practical reasons.

# K-NN Classification Rule

Let's try classification with K-NN density estimate

Data: $N$ - total points

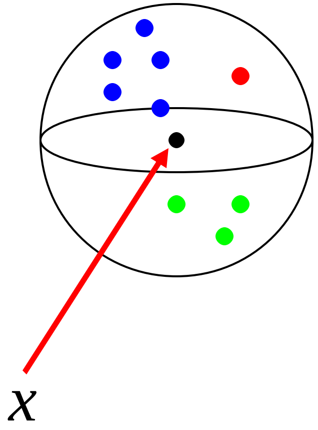
$N_j$ - points in class $w_j$

Need to find the class label for a query, $x$

Expand a sphere from $x$ to include $K$ points

$K$ - number of neighbors of $x$

$K_j$ - points of class $w_j$ among $K$



# KNN Classification

Then class priors are given by: $p(w_j) = \dfrac{N_j}{N}$

We can estimate conditional and marginal densities around any *x*:

$$p(x \mid w_j) = \frac{K_j}{N_j V} \qquad\qquad p(x) = \frac{K}{NV}$$

By Bayes rule: $$p(w_j \mid x) = \frac{K_j}{N_j V}\frac{N_j}{N}\frac{NV}{K} = \frac{K_j}{K}$$

Then for *minimum error rate* classification:

$$C = \arg\max_j K_j$$

# KNN Classification

Important theoretical result:

In the extreme case, *K=1*, it can be shown that:

*N-sample error rate*

for $$P = \lim_{N \to \infty} P_N(error)$$

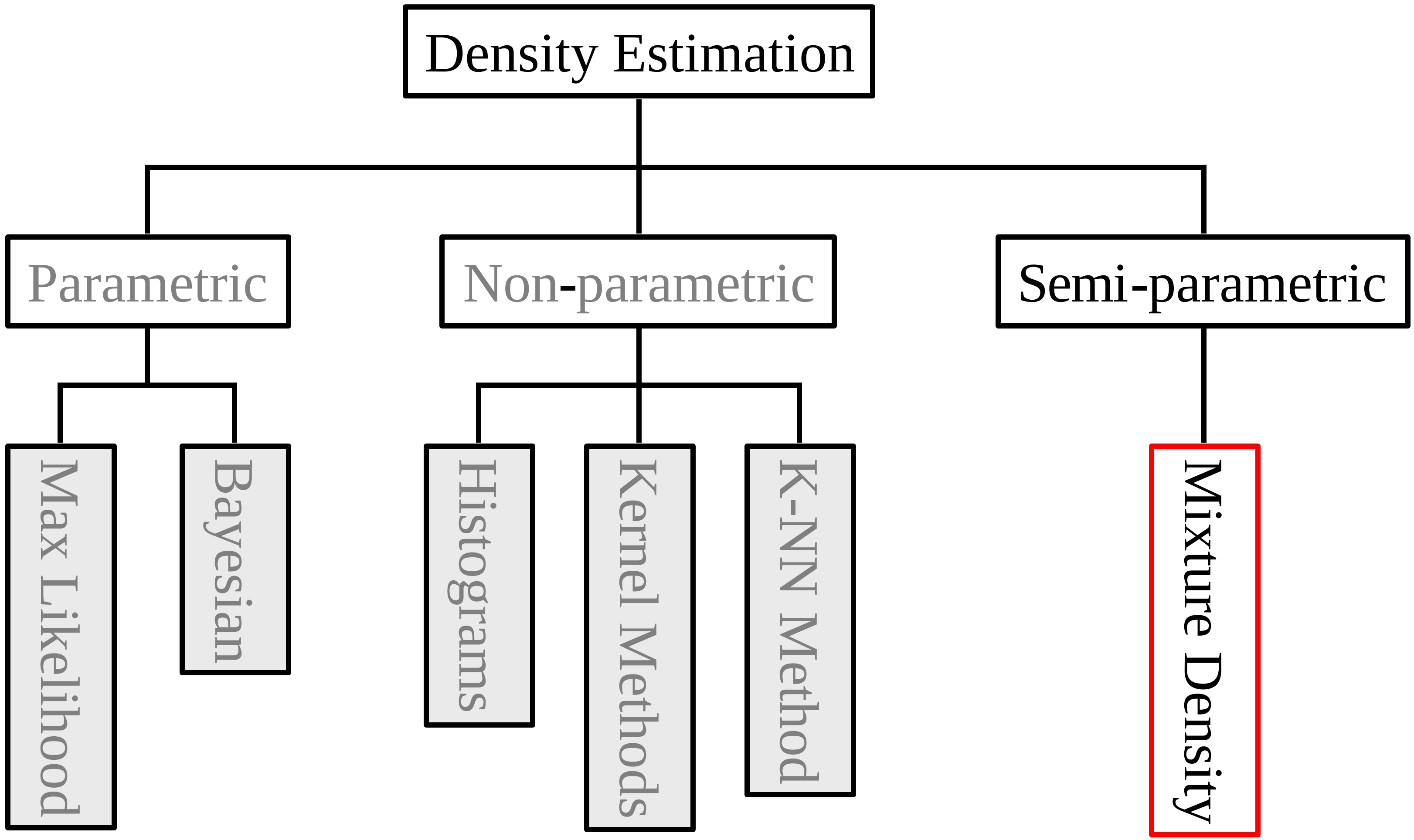
$$P^* \leq P \leq P^* \left( 2 - \frac{c}{c-1} P^* \right)$$

That is, using just a single neighbor rule, the error rate is at most twice the Bayes error!!!

# Problems with Non-parametric Methods

- Memory: need to store all data points
- Computation: need to compute distances to all data points every time
- Parameter choice: need to choose the smoothing parameter

# You Are Here



Density Estimation
Parametric
Non-parametric
Semi-parametric
Max Likelihood
Bayesian
Histograms
Kernel Methods
K-NN Method
Mixture Density

# Mixture Density Model

*Mixture model* – a linear combination of parametric densities

*Number of components*

$$p(x) = \sum_{j=1}^{M} p(x \mid j) P(j)$$

*Component density*

*Component "prior"*

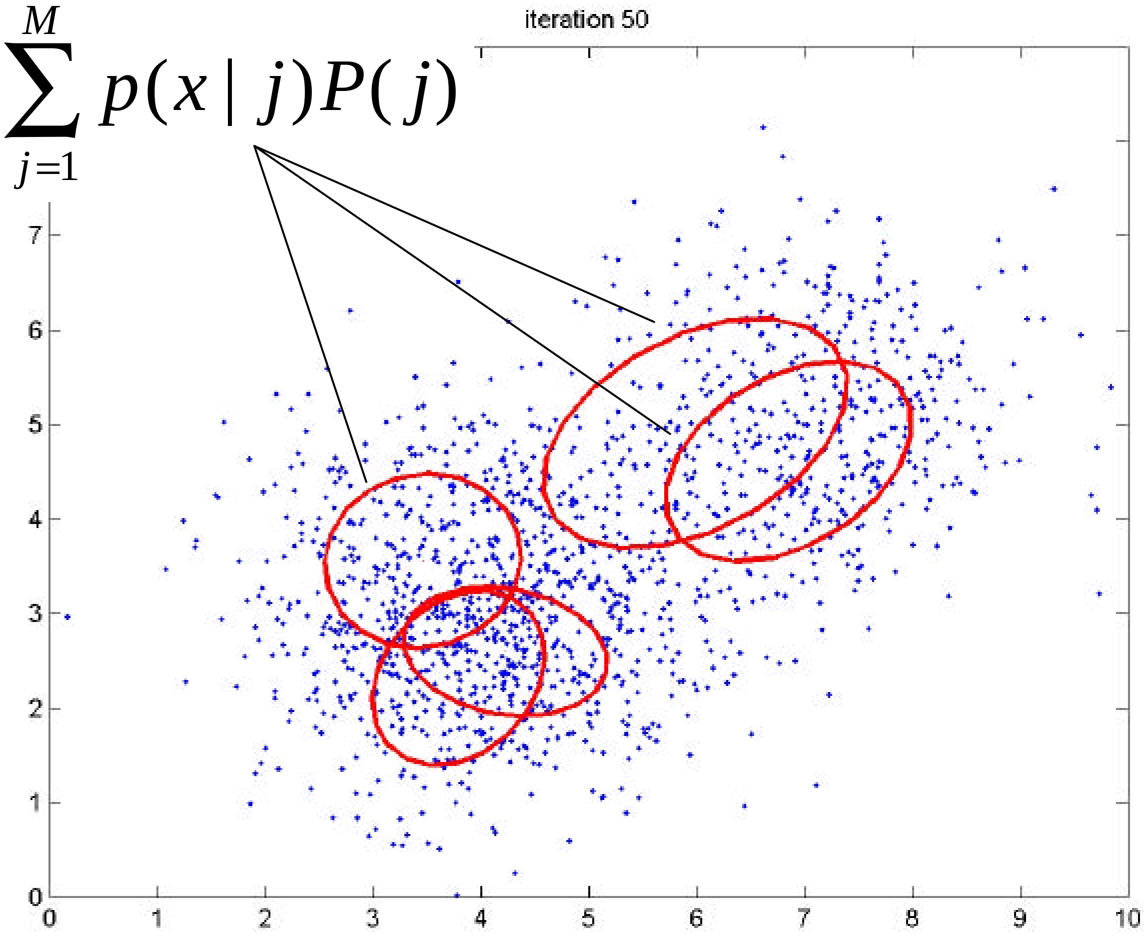
$$P(j) \geq 0, \quad \forall j \qquad \text{and} \qquad \sum_{j=1}^{M} P(j) = 1$$

Uses MUCH less "kernels" than kernel methods

Kernels are parametric densities, subject to estimation

# Example

$$p(x) = \sum_{j=1}^{M} p(x \mid j) P(j)$$



# Mixture Density

Using ML principle, the objective function is the *log-likelihood*:

$$l(q) \equiv \log \prod_{n=1}^{N} p(x^n) = \sum_{n=1}^{N} \log \left\{ \sum_{j=1}^{M} p(x^n \mid j) P(j) \right\}$$

Differentiate w.r.t. parameters:

$$\nabla_{q_j} l(q) = \sum_{n=1}^{N} \frac{\partial}{\partial q_j} \log \left\{ \sum_{k=1}^{M} p(x^n \mid k) P(k) \right\}$$

$$= \sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{M} p(x^n \mid k) P(k)} \frac{\partial}{\partial q_j} p(x^n \mid j) P(j)$$

Again let's assume that $p(x|\omega)$ is a Gaussian

We need to estimate *M* priors, and *M* sets of means and covariances

$$\frac{\partial l(q)}{\partial m_j} = \sum_{n=1}^{N} P(j\,|\,x^n)\left[\Sigma_j^{-1}(x^n - \hat{m}_j)\right]$$

Setting it to *0* and solving for $\mu_j$:

$$\hat{m}_j = \frac{\sum_{n=1}^{N} P(j\,|\,x^n)x^n}{\sum_{n=1}^{N} P(j\,|\,x^n)}$$

- convex sum of all data

# Mixture Density

Similarly for the covariances:

$$\frac{\partial l(q)}{\partial s_j^2} = \sum_{n=1}^{N} P(j \mid x^n)\left[\hat{\mathbf{S}}_j^{-1} - \hat{\mathbf{S}}_j^{-1}(x^n - \hat{m}_j)(x^n - \hat{m}_j)^T \hat{\mathbf{S}}_j^{-1}\right]$$

Setting it to *0* and solving for $\boldsymbol{\Sigma}_i$:

$$\hat{\mathbf{S}}_j = \frac{\sum_{n=1}^{N} P(j \mid x^n)\left(x^n - \hat{m}_j\right)\left(x^n - \hat{m}_j\right)^T}{\sum_{n=1}^{N} P(j \mid x^n)}$$

# Mixture Density

A little harder for $P(j)$ – optimization is subject to constraints:

$$\sum_{j=1}^{M} P(j) = 1 \quad and \quad P\left(j\right) \geq 0, \forall j$$

Here is a trick to enforce the constraints:

$$P(j) = \frac{\exp(g_j)}{\sum_{k=1}^{M} \exp(g_k)}$$

$$\frac{\partial P(i)}{\partial g_j} = d(i - j)P(j) - P(i)P(j)$$

# Mixture Density

Using the chain rule:

$$\nabla_{g_j} l(q) = \sum_{k=1}^{M} \frac{\partial l(q)}{\partial P(k)} \frac{\partial P(k)}{\partial g_j} = \sum_{k=1}^{M} \sum_{n=1}^{N} \frac{p(x^n \mid k)}{P(x)} \left( d_{jk} P(j) - P(j) P(k) \right)$$

$$= \sum_{n=1}^{N} \left\{ \frac{p(x^n \mid j)}{P(x)} P(j) - \sum_{k=1}^{M} \frac{p(x^n \mid k)}{P(x)} P(j) P(k) \right\}$$

$$= \sum_{n=1}^{N} \left\{ P(j \mid x^n) - P(j) \sum_{k=1}^{M} p(k \mid x^n) \right\} = \sum_{n=1}^{N} \left\{ P(j \mid x^n) - P(j) \right\} = 0$$

The last expression gives the value at the extremum:

$$P(j) = \frac{1}{N} \sum_{n=1}^{N} P(j \mid x^n)$$

# Mixture Density

What's the problem?

$$P(j) = \frac{1}{N}\sum_{n=1}^{N} P(j \mid x^n)$$

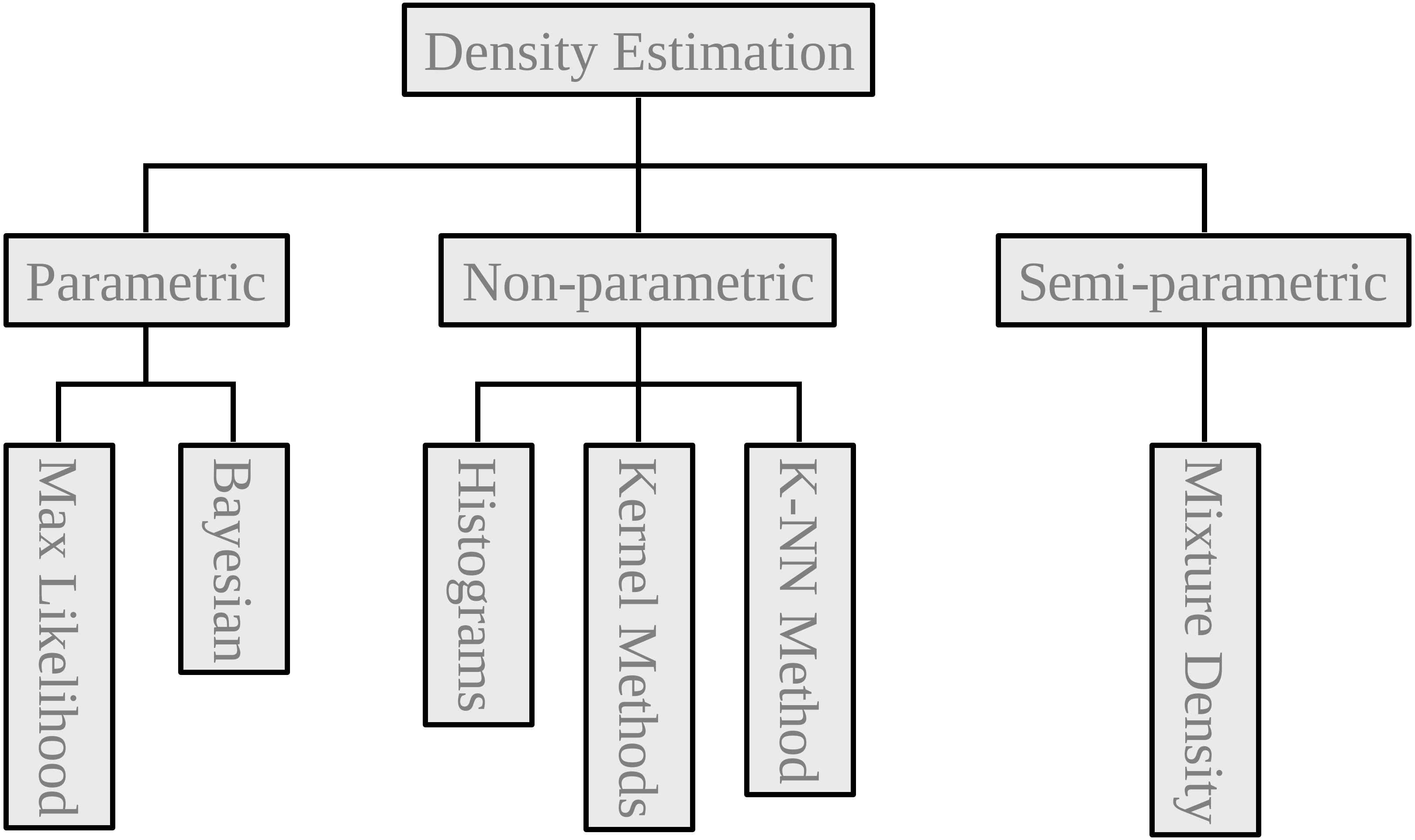
$$\hat{m}_j = \frac{\sum_{n=1}^{N} P(j \mid x^n) x^n}{\sum_{n=1}^{N} P(j \mid x^n)}$$

$$\hat{\mathbf{S}}_j = \frac{\sum_{n=1}^{N} P(j \mid x^n)\left(x^n - \hat{m}_j\right)\left(x^n - \hat{m}_j\right)^T}{\sum_{n=1}^{N} P(j \mid x^n)}$$

We can't compute these directly!

Solution – EM algorithm. We will study it in Clustering.

# You Are Here


Density Estimation
Parametric
Non-parametric
Semi-parametric
Max Likelihood
Bayesian
Histograms
Kernel Methods
K-NN Method
Mixture Density