Non-parametric Methods

Greg Mori - CMPT 419/726

Bishop PRML Ch. 2.5

Outline

Kernel Density Estimation

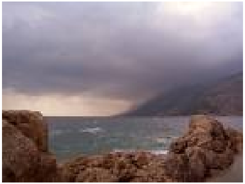
Nearest-neighbour

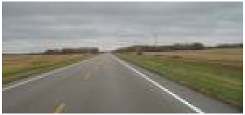
- These are non-parametric methods
  - Rather than having a fixed set of parameters (e.g. weight vector for regression, μ, Σ for Gaussian) we have a possibly infinite set of parameters based on each data point

Outline

Histograms

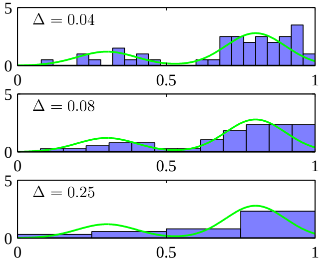




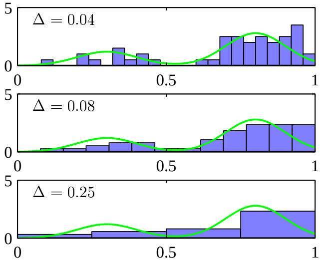
- Consider the problem of modelling the distribution of brightness values in pictures taken on sunny days versus cloudy days
- We could build histograms of pixel values for each class

Histograms



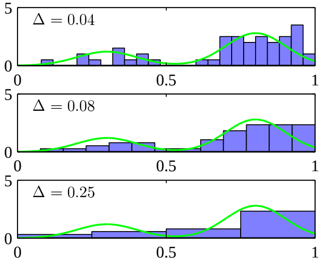
- E.g. for sunny days
- Count n_i number of datapoints (pixels) with brightness value falling into each bin: $p_i = \frac{n_i}{N\Delta_i}$
- Sensitive to bin width Δ_i
- Discontinuous due to bin edges
- In D-dim space with M bins per dimension, M^D bins

Histograms



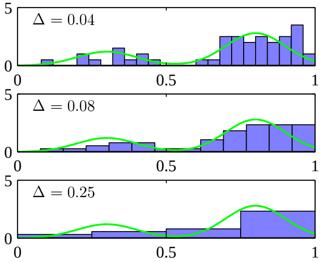
- E.g. for sunny days
- Count n_i number of datapoints (pixels) with brightness value falling into each bin: $p_i = \frac{n_i}{N\Delta_i}$
- Sensitive to bin width Δ_i
- Discontinuous due to bin edges
- In D-dim space with M bins per dimension, M^D bins

Histograms



- E.g. for sunny days
- Count n_i number of datapoints (pixels) with brightness value falling into each bin: $p_i = \frac{n_i}{N\Delta_i}$
- Sensitive to bin width Δ_i
- Discontinuous due to bin edges
- In D-dim space with M bins per dimension, M^D bins

Histograms



- E.g. for sunny days
- Count n_i number of datapoints (pixels) with brightness value falling into each bin: $p_i = \frac{n_i}{N\Delta_i}$
- Sensitive to bin width Δ_i
- Discontinuous due to bin edges
- In D-dim space with M bins per dimension, M^D bins

Local Density Estimation

- In a histogram we use *nearby* points to estimate density
- For a small region around x, estimate density as:

$$p(x) = \frac{K}{NV}$$

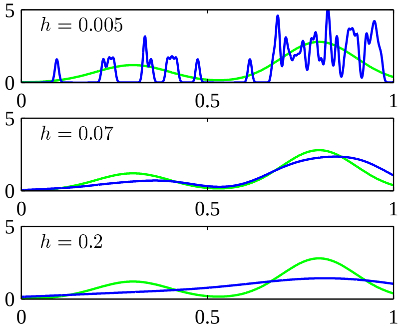
- K is number of points in region, V is volume of region, N is total number of datapoints

Kernel Density Estimation

- Try to keep idea of using *nearby* points to estimate density, but obtain smoother estimate
- Estimate density by placing a small bump at each datapoint
  - Kernel function $k(\cdot)$ determines shape of these bumps
- Density estimate is

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

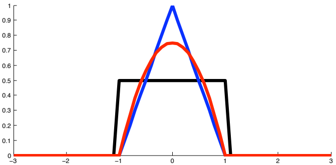
Kernel Density Estimation



- Example using Gaussian kernel:

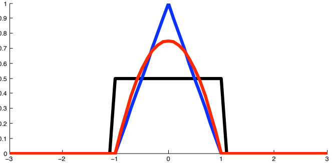
$$p(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{1/2}} \exp\left\{ -\frac{||\boldsymbol{x} - \boldsymbol{x}_n||^2}{2h^2} \right\}$$

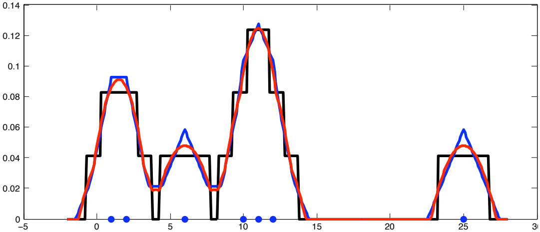
Kernel Density Estimation



- Other kernels: **Rectangle**, Triangle, Epanechnikov

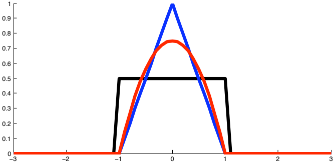
Kernel Density Estimation

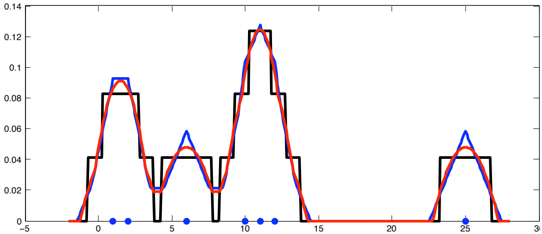




- Other kernels: **Rectangle**, Triangle, Epanechnikov

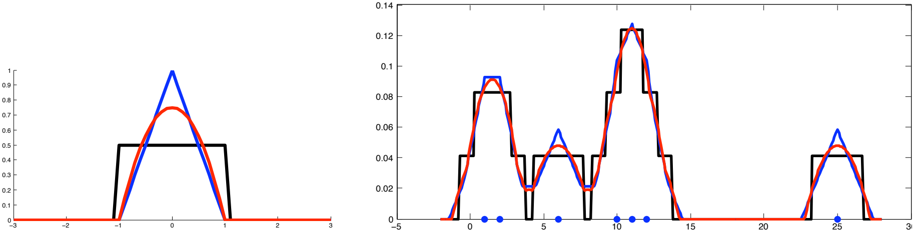
Kernel Density Estimation





- Other kernels: **Rectangle**, Triangle, Epanechnikov
- Fast at training time, slow at test time – keep all datapoints

Kernel Density Estimation



- Other kernels: **Rectangle**, Triangle, Epanechnikov
- Fast at training time, slow at test time – keep all datapoints
- Sensitive to kernel bandwidth h

Outline

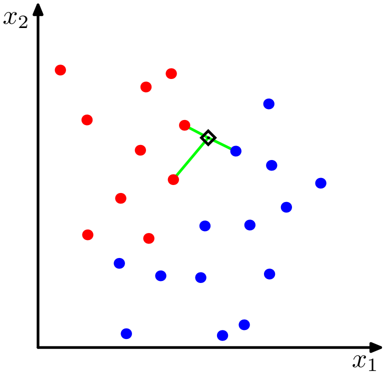
Kernel Density Estimation

Nearest-neighbour

Nearest-neighbour for Classification

- K Nearest neighbour is often used for classification
  - Classification: predict labels t_i from x_i

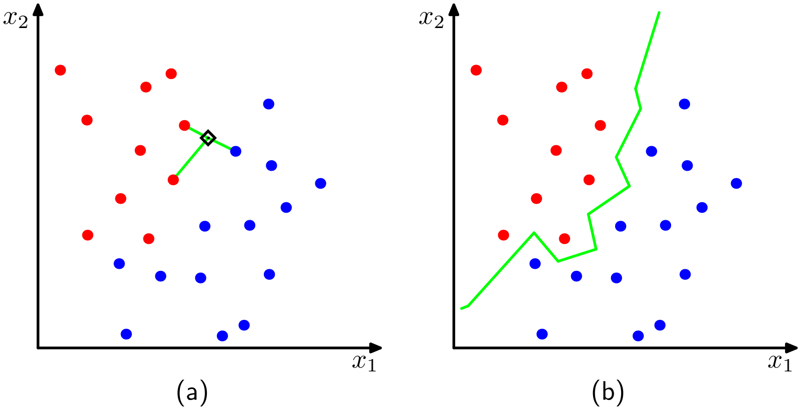
Nearest-neighbour for Classification



(a)

- K Nearest neighbour is often used for classification
  - Classification: predict labels t_i from x_i
  - e.g. $\boldsymbol{x}_i \in \mathbb{R}^2$ and $t_i \in \{0, 1\}$, 3-nearest neighbour

Nearest-neighbour for Classification

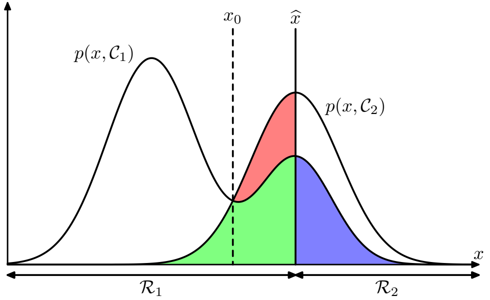


- K Nearest neighbour is often used for classification
  - Classification: predict labels t_i from x_i
  - e.g. $\boldsymbol{x}_i \in \mathbb{R}^2$ and $t_i \in \{0, 1\}$, 3-nearest neighbour
- $K = 1$ referred to as nearest-neighbour

Nearest-neighbour for Classification

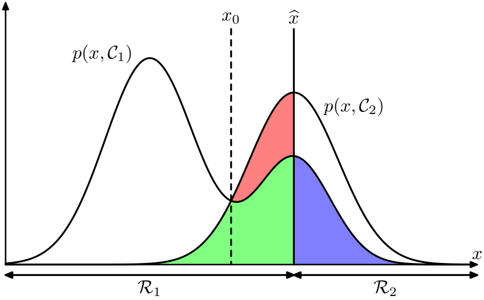
- Good baseline method
  - Slow, but can use fancy data structures for efficiency (KD-trees, Locality Sensitive Hashing)
- Nice theoretical properties
  - As we obtain more training data points, space becomes more filled with labelled data
  - As $N \to \infty$ error no more than twice Bayes error

Bayes Error



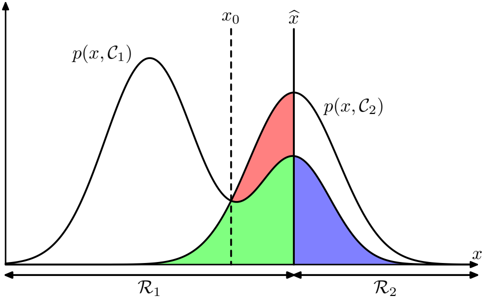
- Best classification possible given features
- Two classes, PDFs shown
- Decision rule: C_1 if $x \leq \hat{x}$; makes errors on red, green, and blue regions
- Optimal decision rule: C_1 if $x \leq x_0$, Bayes error is area of green and blue regions

Bayes Error



- Best classification possible given features
- Two classes, PDFs shown
- Decision rule: C_1 if $x \leq \hat{x}$; makes errors on red, green, and blue regions
- Optimal decision rule: C_1 if $x \leq x_0$, Bayes error is area of green and blue regions

Bayes Error



- Best classification possible given features
- Two classes, PDFs shown
- Decision rule: C_1 if $x \leq \hat{x}$; makes errors on red, green, and blue regions
- Optimal decision rule: C_1 if $x \leq x_0$, Bayes error is area of green and blue regions

Conclusion

- Readings: Ch. 2.5
- Kernel density estimation
  - Model density $p(x)$ using kernels around training datapoints
- Nearest neighbour
  - Model density or perform classification using nearest training datapoints
- Multivariate Gaussian
  - Needed for next lectures, if you need a refresher read pp. 78-81