Non parametric classification

Course of Machine Learning
Master Degree in Computer Science

University of Rome “Tor Vergata”

a.a. 2020-2021

Giorgio Gambosi

Probabilistic classification methods recap

The application of probabilistic classifier requires that the (at least approximate) knowledge of a suitable distribution is derived from the training set

- the class conditional distribution $p(C_k|\mathbf{x})$ for each class C_k in the discriminative case, where an item $\mathbf{x}$ shall be assigned to C_i if

$$i = \underset{k}{\operatorname{argmax}}\, p(C_k|\mathbf{x})$$

- the class conditional distribution $p(\mathbf{x}|C_k)$ (and the prior distribution $p(C_k)$) for each class C_k in the generative (bayesian) case, where an item $\mathbf{x}$ shall be assigned to C_i if

$$i = \underset{k}{\operatorname{argmax}}\, p(\mathbf{x}|C_k)p(C_k)$$

Parametric approach

The type of probability distribution is assumed to be known: the value of a suitable set of coefficients must be derived. For example,

- $p(C_k|\mathbf{x})$ is assumed to be of the type $\frac{e^{\mathbf{w}_k^T\overline{\mathbf{x}}}}{\sum_i e^{\mathbf{w}_i^T\overline{\mathbf{x}}}}$ in the case of softmax (a discriminative method)
- $p(\mathbf{x}|C_k)$ is assumed to be of the type $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$ in the case of gaussian discriminant analysis (a generative method)

Parametric approach

In both case, an estimate of parameter values (either $\mathbf{w}_k$ or $\boldsymbol{\theta}_k$) is performed for all classes. Different approaches to parameter estimation:

Maximum likelihood :

- In the discriminative case, the likelihood of the target is considered $\mathbf{w}^{ML} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{t}|\mathbf{X}, \mathbf{w})$: prediction is performed as $\underset{k}{\operatorname{argmax}}\, p(C_k|\mathbf{x}, \mathbf{w}^{ML})$
- In the generative case, for each class C_k, the likelihood of the subset $\mathbf{X}_k$ of items belonging the class is instead maximized, that is $\boldsymbol{\theta}_k^{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{X}_k|\boldsymbol{\theta}_k)$: prediction is performed as $\underset{k}{\operatorname{argmax}}\, p(\mathbf{x}|\boldsymbol{\theta}_k^{ML})p(C_k)$

Parametric approach

Maximum a posteriori : Similar to the previous one:

- In the discriminative case, the posterior of the parameters wrt to training set $\mathbf{w}^{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{w}|\mathbf{X}, \mathbf{t})$: prediction is performed as $\underset{k}{\operatorname{argmax}}\, p(C_k|\mathbf{x}, \mathbf{w}^{MAP})$
- In the generative case, for each class C_k, the posterior of the parameters wrt the items in the class $\boldsymbol{\theta}_k^{MAP} = \underset{\boldsymbol{\theta}_k}{\operatorname{argmax}}\, p(\boldsymbol{\theta}_k|\mathbf{X}_k)$ is maximized: prediction is performed as $\underset{k}{\operatorname{argmax}}\, p(\mathbf{x}|\boldsymbol{\theta}_k^{MAP})p(C_k)$

Parametric approach

Bayesian estimate : This approach directly express the predictive distribution as

$$p(C_k|\mathbf{x}, \mathbf{X}, \mathbf{t}) = \int_{\mathbf{w}} p(C_k|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{t}) d\mathbf{w}$$

Non parametric approach

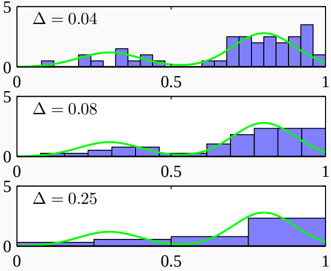
No knowledge whatsoever of the probabilities is assumed.

- The class distributions $p(\mathbf{x}|C_i)$ are directly from data.
- In previous cases, use of (parametric) models for a synthetic description of data in $\mathbf{X}, \mathbf{t}$
- In this case, no models (and parameters): training set items explicitly appear in class distribution estimates.
- Denoted as non parametric models: indeed, an unbounded number of parameters is used

Histograms

- Elementary type of non parametric estimate
- Domain partitioned into m d-dimensional intervals (bins)
- The probability $P_{\mathbf{x}}$ that an item belongs to the bin containing item $\mathbf{x}$ is estimated as $\frac{n(\mathbf{x})}{n}$, where $n(\mathbf{x})$ is the number of element in that bin
- The probability density in the interval corresponding to the bin containing $\mathbf{x}$ is then estimated as the ratio between the above probability and the interval width $\Delta(\mathbf{x})$ (tipically, a constant Δ)

$$p_H(\mathbf{x}) = \frac{\frac{n(\mathbf{x})}{N}}{\Delta(\mathbf{x})} = \frac{n(\mathbf{x})}{N\Delta(\mathbf{x})}$$



Kernel density estimators

- Probability that an item is in region $\mathcal{R}(\mathbf{x})$, containing $\mathbf{x}$

$$P_{\mathbf{x}} = \int_{\mathcal{R}(\mathbf{x})} p(\mathbf{z}) d\mathbf{z}$$

- Given n items $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, the probability that k among them are in $\mathcal{R}(\mathbf{x})$ is given by the binomial distribution

$$p(k) = \binom{n}{k} P_{\mathbf{x}}^k (1 - P_{\mathbf{x}})^{n-k} = \frac{n!}{k!(n-k)!} P_{\mathbf{x}}^k (1 - P_{\mathbf{x}})^{n-k}$$

- Since $E[k] = nP_{\mathbf{x}}$ and $\sigma_k^2 = nP_{\mathbf{x}}(1 - P_{\mathbf{x}})$, by the binomial distribution properties, we have that, for what concerns the ratio $r = \frac{k}{n}$,

$$E\left[r\right] = \frac{1}{n} E[k] = P_{\mathbf{x}} \qquad \sigma_r^2 = \frac{1}{n^2} \sigma_k^2 = \frac{P_{\mathbf{x}}(1 - P\mathbf{x})}{n}$$

- $P_{\mathbf{x}}$ is the expected fraction of items in $\mathcal{R}(\mathbf{x})$, and the ratio r is an estimate. As $n \to \infty$ variance decreases and r tends to $E[r] = P_{\mathbf{x}}$, we assume

$$r = \frac{k}{n} \simeq P_{\mathbf{x}}$$

Nonparametric estimates

- Let the volume of $\mathcal{R}(\mathbf{x})$ be sufficiently small. Then, the density $p(\mathbf{x})$ is almost constant in the region and

$$P_{\mathbf{x}} = \int_{\mathcal{R}(\mathbf{x})} p(\mathbf{z})d\mathbf{z} \simeq p(\mathbf{x})V$$

  where V is the volume of $\mathcal{R}(\mathbf{x})$
- since $P_{\mathbf{x}} \simeq \frac{k}{n}$, it then derives that $p(\mathbf{x}) \simeq \frac{k}{nV}$

Approaches to nonparametric estimates

Two alternative ways to exploit the relation $p(\mathbf{x}) \simeq \frac{k}{nV}$ to estimate $p(\mathbf{x})$ for any $\mathbf{x}$:

1. Fix V and derive k from data (kernel density estimation)
2. Fix k and derive V from data (K-nearest neighbor).

It can be shown that in both cases, under suitable conditions, the estimator tends to the true density $p(\mathbf{x})$ as $n \to \infty$.

Kernel density estimation: Parzen windows

- Region associated to a point $\mathbf{x}$: hypercube with edge length h (and volume h^d) centered on $\mathbf{x}$.
- Kernel function $k(\mathbf{z})$ (Parzen window) used to count the number of items in the unit hypercube centered on the origin $\mathbf{0}$

$$k(\mathbf{z}) = \begin{cases} 1 & |z_i| \leq 1/2 \qquad\qquad i = 1, \ldots, d \\ 0 & \text{otherwise} \end{cases}$$

- as a consequence, $k\left(\dfrac{\mathbf{x}-\mathbf{x}'}{h}\right) = 1$ iff $\mathbf{x}'$ is in the hypercube of edge length h centered on $\mathbf{x}$
- the number of items in the hypercube is then

$$K = \sum_{i=1}^{n} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

Kernel density estimation: Parzen windows

- The estimated density is

$$p(\mathbf{x}) = \frac{1}{nV}\sum_{i=1}^{n} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) = \frac{1}{nh^d}\sum_{i=1}^{n} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

- Since

$$k(\mathbf{z}) \geq 0 \qquad \text{and} \qquad \int k(\mathbf{z})d\mathbf{z} = 1$$

it derives

$$k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \geq 0 \qquad \text{and} \qquad \int k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) d\mathbf{x} = h^d$$

Kernel density estimation: Parzen windows

As a consequence, it results that $p_n(\mathbf{x})$ is a probability density. In fact,

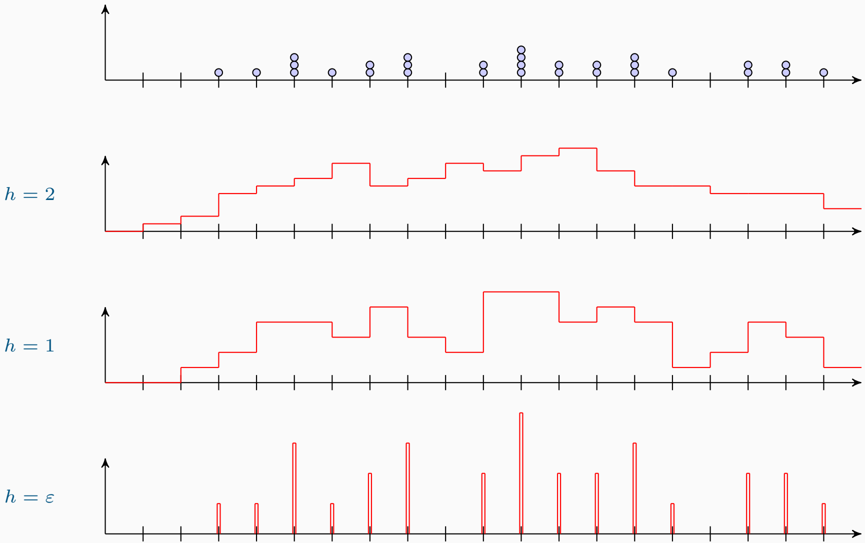
$$p(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h^d}k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \geq 0$$

and

$$\begin{aligned}\int p(\mathbf{x})d\mathbf{x} &= \int \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h^d}k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)d\mathbf{x}\\ &\frac{1}{nh^d}\int\sum_{i=1}^{n}k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)d\mathbf{x}\\ &\frac{1}{nh^d}\sum_{i=1}^{n}\int k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)d\mathbf{x} = \frac{1}{nh^d}nh^d = 1\end{aligned}$$

Clearly, the window size has a relevant effect on the estimate

Kernel density estimation: Parzen windows



Kernels and smoothing

Drawbacks

1. discontinuity of the estimates
2. items in a region centered on $\mathbf{x}$ have uniform weights: their distance from $\mathbf{x}$ is not taken into account

Solution. Use of smooth kernel functions $\kappa_h(u)$ to assign larger weights to points nearer to the origin.
Assumed characteristics of $\kappa_h(u)$:

$$\int \kappa_h(\mathbf{x})d\mathbf{x} = 1$$

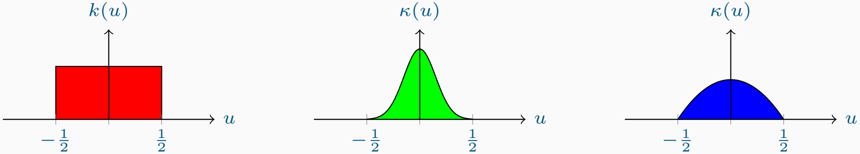
$$\int \mathbf{x}\kappa_h(\mathbf{x})d\mathbf{x} = 0$$

$$\int \mathbf{x}^2\kappa_h(\mathbf{x})d\mathbf{x} > 0$$

Kernels and smoothing

Usually kernels are based on smooth radial functions (functions of the distance from the origin)

1. gaussian $\kappa(u) = \dfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{u^2}{\sigma^2}}$, unlimited support
2. Epanechnikov $\kappa(u) = 3\left(\dfrac{1}{2} - u^2\right)$, $|u| \leq \frac{1}{2}$, limited support
3. $\cdots$

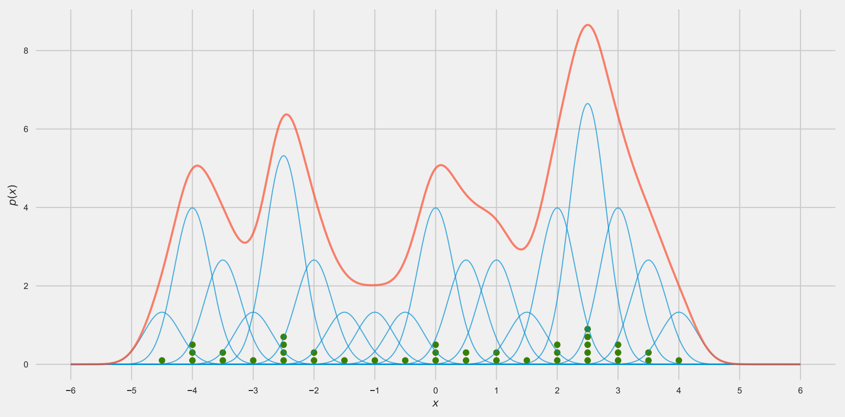


resulting estimate:

$$p(\mathbf{x}) = \frac{1}{nh}\sum_{i=1}^{n}\kappa\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) = \frac{1}{n}\sum_{i=1}^{n}\kappa_h\left(\mathbf{x}-\mathbf{x}_i\right)$$

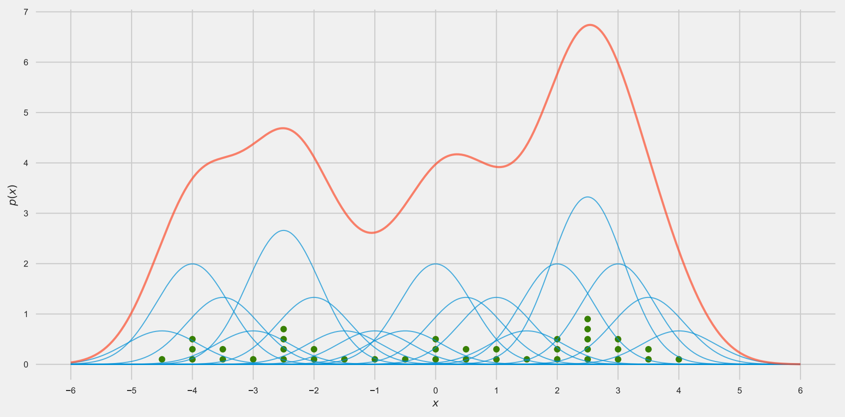
Kernels and smoothing

$h = 1$



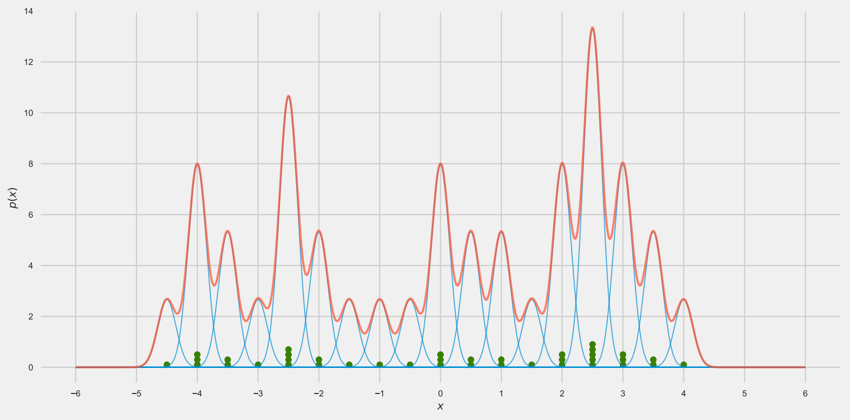
Kernels and smoothing

$h = 2$



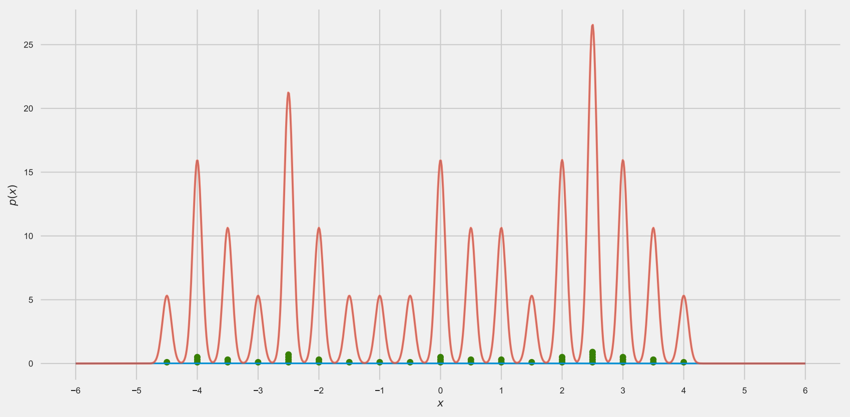
Kernels and smoothing

$h = .5$



Kernels and smoothing

$h = .25$



Parzen windows and classification

- Parzen windows provide a way to estimate $p(\mathbf{x})$ for any $\mathbf{x}$, given a set of points $\mathbf{X}$
- They can be applied to classify an item $\mathbf{x}$ by estimating $p(\mathbf{x}|C_k)$ for all classes, by referring to the sets $\mathbf{X}_1, \ldots, \mathbf{X}_k$ of items in the training set belonging to each class
- According to bayesian classification, $\mathbf{x}$ is predicted to the class with index

$$\begin{aligned}\operatorname*{argmax}_i p(\mathbf{x}|C_i)p(C_i) &= \operatorname*{argmax}_i \frac{1}{n_i h^d}\sum_{i=1}^{n_i} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)p(C_i) = \\ &= \operatorname*{argmax}_i \frac{1}{n h^d}\sum_{i=1}^{n_i} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \\ &= \operatorname*{argmax}_i \sum_{i=1}^{n_i} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\end{aligned}$$

- that is, an item is assigned to the class with most (weighted by the kernel) points near $\mathbf{x}$, that is in an hypercube of edge size h with center $\mathbf{x}$

Density estimation through kNN

- The region around $\mathbf{x}$ is extended to include k items
- The estimated density is

$$p(\mathbf{x}) \simeq \frac{k}{nV} = \frac{k}{nc_d r_k^d(\mathbf{x})}$$

where:

- c_d is the volume of the d-dimensional sphere of unitary radius
- $r_k^d(\mathbf{x})$ is the distance from $\mathbf{x}$ to the k-th nearest item (the radius of the smallest sphere with center $\mathbf{x}$ containing k items)

Classification through kNN

- To estimate $p(C_i|\mathbf{x})$ in order to classify $\mathbf{x}$, let us consider a hypersphere of volume V with center $\mathbf{x}$ containing k items from the training set
- Let k_i be the number of such items belonging to class C_i. Then, the following approximation holds:

$$p(\mathbf{x}|C_i) = \frac{k_i}{n_i V}$$

  where n_i is the number of items in the training set belonging to class C_i
- Similarly, for the evidence,

$$p(\mathbf{x}) = \frac{k}{nV}$$

- And, for the prior distribution,

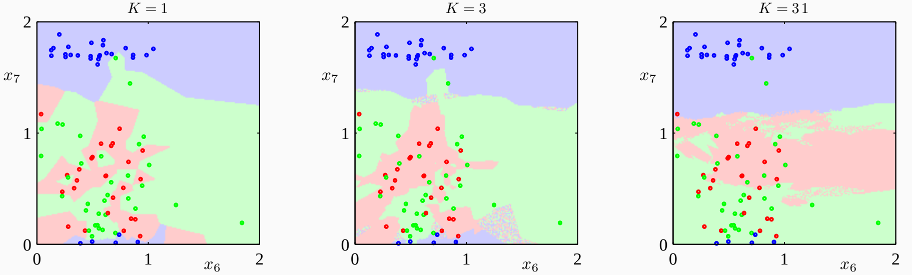
$$p(C_i) = \frac{n_i}{n}$$

- The class posterior distribution is then

$$p(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)p(C_i)}{p(\mathbf{x})} = \frac{\frac{k_i}{n_i V} \cdot \frac{n_i}{n}}{\frac{k}{nV}} = \frac{k_i}{k}$$

Classification through kNN

- Simple rule: an item is classified on the basis of similarity to near training set items
- To classify $\mathbf{x}$, determine the k items in the training nearest to it and assign $\mathbf{x}$ to the majority class among them
- A metric is necessary to measure similarity.



Classification through kNN

- kNN is a simple classifier which can work quite well, provided it is given a good distance metric and has enough labeled training data: it can be shown that it can result within a factor of 2 of the best possible performance as $n \to \infty$
- subject to the curse of dimensionality: due to the large sparseness of data at high dimensionality, items considered by kNN can be quite far away from the query point, and thus resulting in poor locality.