Non parametric models

Course of Machine Learning
Master Degree in Computer Science
University of Rome "Tor Vergata"
a.a. 2019-2020

Giorgio Gambosi

Probability distribution estimates

- The statistical approach to classification requires the (at least approximate) knowledge of $p(\mathcal{C}_i|\mathbf{x})$: in fact, an item $\mathbf{x}$ shall be assigned to the class $\mathcal{C}_i$ such that

$$i = \underset{k}{\text{argmax}}\, p(\mathcal{C}_k|\mathbf{x})$$

- The same holds in the regression case, where $p(y|\mathbf{x})$ has to be estimated.

Probability distribution estimates: hypotheses

What do we assume to know of class distributions, given a training set $\mathbf{X}, \mathbf{t}$?

- Case 1. The probabilities $p(\mathbf{x}|\mathcal{C}_i)$ are known: an item is assigned $\mathbf{x}$ to the class $\mathcal{C}_i$ such that

$$i = \underset{j}{\text{argmax}}\, p(\mathcal{C}_j|\mathbf{x})$$

where $p(\mathcal{C}_j|\mathbf{x})$ can be derived through Bayes' rule and prior probabilities, since $p(\mathcal{C}_k)|\mathbf{x}) \propto p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$

Probability distribution estimates: hypotheses

- Case 2. The *type* of probability distribution $p(\mathbf{x}|\boldsymbol{\theta})$ is known: an estimate of parameter values $\boldsymbol{\theta}_i$ is performed for all classes, taking into account for each class $\mathcal{C}_i$ the subset of $\mathbf{X}_i, \mathbf{t}_i$ of items belonging to the class, that is such that $t = i$. Different approaches to parameter estimation:

  1. Maximum likelihood: $\boldsymbol{\theta}_i^{ML} = \underset{\boldsymbol{\theta}}{\text{argmax}}\, p(\mathbf{X}_i, \mathbf{t}_i|\boldsymbol{\theta})$ is computed. Item $\mathbf{x}$ is assigned to class $\mathcal{C}_i$ if

  $$i = \underset{j}{\text{argmax}}\, p(\mathcal{C}_j|\mathbf{x}) = \underset{j}{\text{argmax}}\, p(\mathbf{x}|\boldsymbol{\theta}_j^{ML})p(\mathcal{C}_j)$$

  2. Maximum a posteriori: $\boldsymbol{\theta}_i^{MAP} = \underset{\boldsymbol{\theta}}{\text{argmax}}\, p(\boldsymbol{\theta}|\mathbf{X}_i, \mathbf{t}_i)$ is computed. Item $\mathbf{x}$ is assigned to class $\mathcal{C}_i$ if

  $$i = \underset{j}{\text{argmax}}\, p(\mathcal{C}_j|\mathbf{x}) = \underset{j}{\text{argmin}}\, p(\mathbf{x}|\boldsymbol{\theta}_j^{MAP})p(\mathcal{C}_j)$$

  3. Bayesian estimate: the distributions $p(\boldsymbol{\theta}|\mathbf{X}_i, \mathbf{t}_i)$ are estimated for each class and, from them,

  $$p(\mathbf{x}|\mathcal{C}_i) = \int_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X}_i, \mathbf{t}_i)d\boldsymbol{\theta}$$

Item $\mathbf{x}$ is assigned to class $\mathcal{C}_i$ if

$$\begin{aligned} i &= \underset{j}{\text{argmax}}\, p(\mathcal{C}_j|\mathbf{x}) = \underset{j}{\text{argmax}}\, p(\mathcal{C}_j)p(\mathbf{x}|\mathcal{C}_j) \\ &= \underset{j}{\text{argmax}}\, p(\mathcal{C}_j) \int_{\boldsymbol\theta} p(\mathbf{x}|\boldsymbol\theta)p(\boldsymbol\theta|\mathbf{X}_j, \mathbf{t}_j)d\boldsymbol\theta \end{aligned}$$

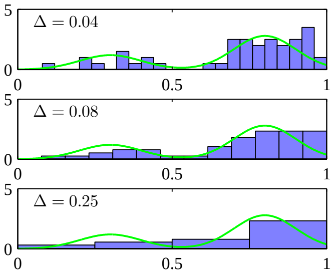
Probability distribution estimates: hypotheses

- Case 3. No knowledge of the probabilities assumed.
- The class distributions $p(\mathbf{x}|\mathcal{C}_i)$ 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 $\dfrac{n(\mathbf{x})}{n}$, 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)

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



Histograms: problems

- The density is a function of the position of the first bin. In the case of multivariate data, also from bin orientation.
- The resulting estimates is not continuous.
- Curse of dimensionality: the number of bins grows as a polynomial of order d: in high-dimensional spaces many bins may result empty, unless a large number of items is available.
- In practice, histograms can be applied only in low-dimensional datasets (1,2)

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

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

- Mean and variance of the ratio $r = \frac{k}{n}$ are

$$E[r] = P_{\mathbf{x}} \qquad \text{var}\,[r] = \frac{P_{\mathbf{x}}(1 - P_{\mathbf{x}})}{n}$$

- $P_{\mathbf{x}}$ 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 $E[r] = P_{\mathbf{x}}$. Hence, in general,

$$r = \frac{k}{n} \simeq P(\mathbf{x})$$

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

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 $k(\mathbf{u})$ (*Parzen window*) used to count the number of items in the unit hypercube centered on the origin $\mathbf{0}$

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

- as a consequence, $k\left(\frac{\mathbf{x}-\mathbf{x}'}{h}\right)=1$ 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

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

- Since

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

  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

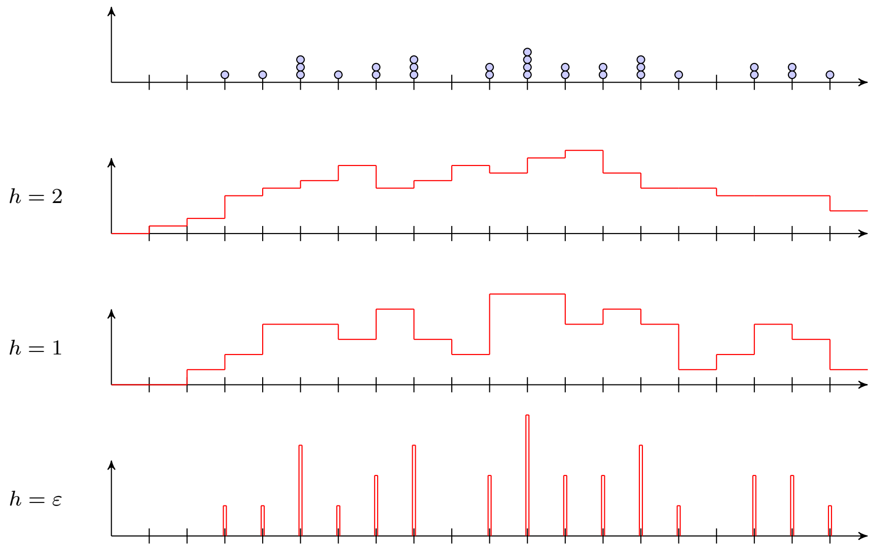
Kernel density estimation: Parzen windows

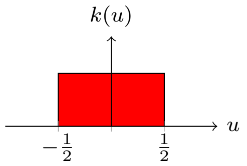
Kernels and smoothing

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 $\kappa_h(u)$:

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

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

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

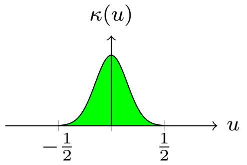
Kernels and smoothing

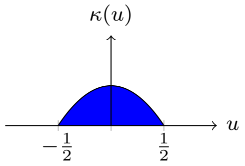
Usually kernels are based on smooth radial functions (functions of the distance from the origin)

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

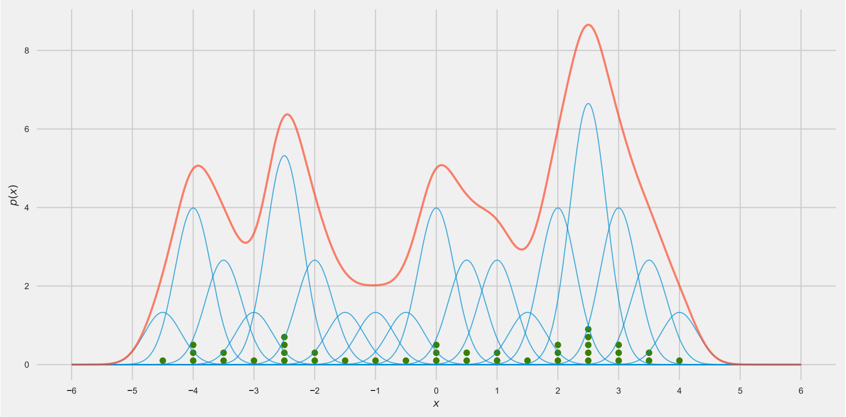
resulting estimate:

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



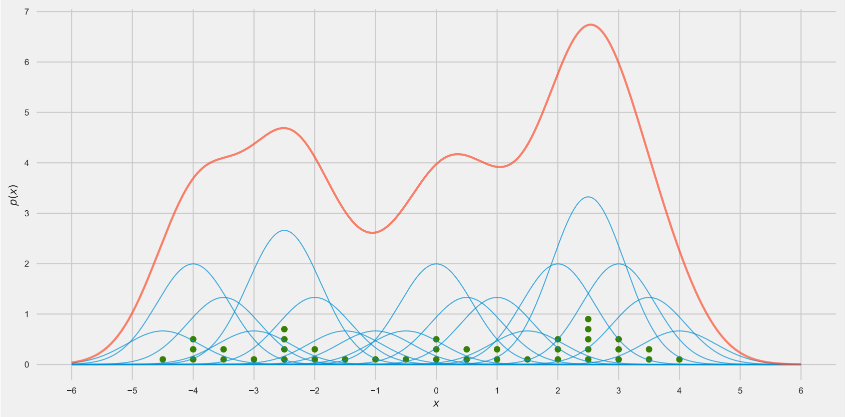


Kernels and smoothing



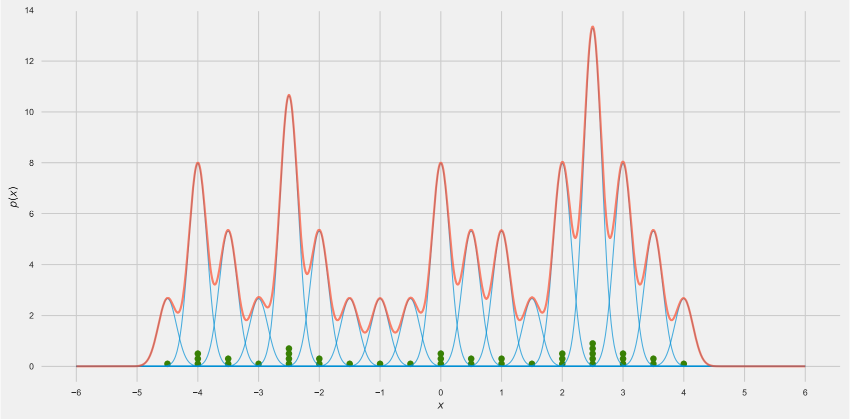
$h = 1$

Kernels and smoothing



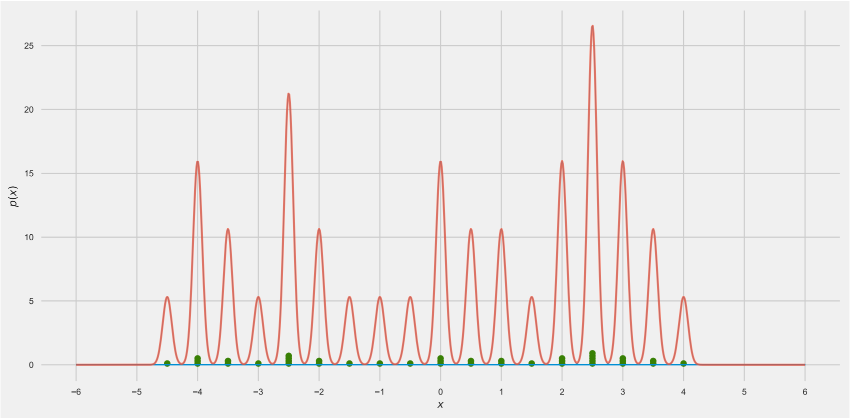
$h = 2$

Kernels and smoothing



$h = .5$

Kernels and smoothing



$h = .25$

Kernels regression

Kernel smoothers methods can be applied also to regression: in this case, the value corresponding to any item $\mathbf{x}$ is predicted by referring to items in the training set (and in particular to the items which are closer to $\mathbf{x}$).

In this case, the conditional expectation

$$f(\mathbf{x}) = \mathrm{E}[y|\mathbf{x}] = \int yp(y|\mathbf{x})dy = \int y\frac{p(\mathbf{x}, y)}{p(\mathbf{x})}dy = \frac{\int yp(\mathbf{x}, y)dy}{p(\mathbf{x})} = \frac{\int yp(\mathbf{x}, y)dy}{\int p(\mathbf{x}, y)dy}$$

should be returned.

Applying kernels, we have

$$p(\mathbf{x}, y) \approx \frac{1}{n}\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)\kappa_h(y - t_i)$$

Kernels regression

This results into

$$f(\mathbf{x}) = \frac{\int y\frac{1}{n}\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)\kappa_h(y - t_i)dy}{\int \frac{1}{n}\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)\kappa_h(y - t_i)dy} = \frac{\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)\int y\kappa_h(y - t_i)dy}{\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)\int \kappa_h(y - t_i)dy}$$

and, since $\displaystyle\int \kappa_h(y - t_i)dy = 1$ and $\displaystyle\int y\kappa_h(y - t_i)dy = y_i$, we get

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)t_i}{\sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)}$$

Kernels regression

By setting

$$w_i(\mathbf{x}) = \frac{\kappa_h(\mathbf{x} - \mathbf{x}_i)}{\sum_{j=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_j)}$$

we can write

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x})t_i$$

that is, the predicted value is computed as a linear combination of all target values, weighted by kernels (Nadaraya-Watson)

Locally weighted regression

In Nadaraya-Watson model, the prediction is performed by means of a weighted combination of constant values (target values in the training set).

Locally weighted regression improves that approach by referring to a weighted version of the sum of squared differences loss function used in regression.

If a value y has to be predicted for a provided item $\mathbf{x}$, a "local" version of the loss function is considered, with weight $\mathbf{w}_i$ dependent from the "distance" between $\mathbf{x}$ and $\mathbf{x}_i$.

$$L(\mathbf{x}) = \sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)(\mathbf{w}^T\overline{\mathbf{x}}_i - t_i)^2$$

Locally weighted regression

An instance of the weighted regression problem must be solved,

$$\hat{\mathbf{w}}(\mathbf{x}) = \underset{\mathbf{w}}{\text{argmin}} \sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)(\mathbf{w}^T\overline{\mathbf{x}}_i - t_i)^2$$

which has solution

$$\hat{\mathbf{w}}(\mathbf{x}) = (\overline{\mathbf{X}}^T\Psi(\mathbf{x})\overline{\mathbf{X}})^{-1}\overline{\mathbf{X}}^T\Psi(\mathbf{x})\mathbf{t}$$

where $\Psi(\mathbf{x})$ is a diagonal $n \times n$ matrix with $\Psi(\mathbf{x})_{ii} = \kappa_h(\mathbf{x} - \mathbf{x}_i)$.

The prediction is then performed as usual, as

$$y = \hat{\mathbf{w}}(\mathbf{x})^T\overline{\mathbf{x}}$$

The same considerations can be done if polynomial regression is applied

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 classify $\mathbf{x}_i$, 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 $\mathcal{C}_i$. Then, the following approximation holds:

  $$p(\mathbf{x}|\mathcal{C}_i) = \frac{k_i}{n_i V}$$

  where n_i is the number of items in the training set belonging to class $\mathcal{C}_i$
- Similarly, for the evidence,

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

- And, for the prior distribution,

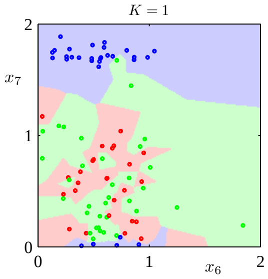
$$p(\mathcal{C}_i) = \frac{n_i}{n}$$

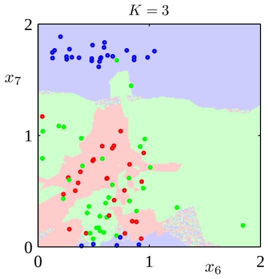
- The class posterior distribution is then

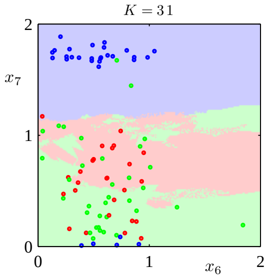
$$p(\mathcal{C}_i|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_i)p(\mathcal{C}_i)}{p(\mathbf{x})} = \frac{\frac{k_i}{n_i V} \cdot \frac{n_i}{n}}{\frac{k}{nV}} = \frac{k_i}{k}$$

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 is simple and 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.

Local logistic regression

The same approach applied in the case of local regression can be applied for classification, by defining a weighted loss function to be minimized, with weights dependent from the item whose target must be predicted.

In this case, a weighted version of the cross entropy function is considered, which has to be maximized

$$L(\mathbf{x}) = \sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)(t_i \log p_i - (1 - t_i) \log(1 - p_i))$$

with $p_i = \sigma(\mathbf{w}^T \overline{\mathbf{x}}_i)$, as usual.

A suitable modification of the IRLS algorithm for logistic regression can be applied here to compute

$$\hat{\mathbf{w}}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{w}} \sum_{i=1}^{n} \kappa_h(\mathbf{x} - \mathbf{x}_i)(t_i \log p_i - (1 - t_i) \log(1 - p_i))$$

Some properties of Gaussian distribution

In order to introduce Gaussian processes and how they can be exploited for regression, let us first provide a short reminder on some properties of multivariate gaussian distributions.

Consider a random vector $\mathbf{x} = (x_1, \ldots, x_n)^T$ with $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and let $\mathbf{x} = (\mathbf{x}_A, \mathbf{x}_B)$ be a partition of the components $\mathbf{x}$ such that:

- $\mathbf{x}_A = (x_1, \ldots, x_r)^T$
- $\mathbf{x}_B = (x_{r+1}, \ldots, x_n)^T$

Then, both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be partitioned as

$$\boldsymbol{\mu} = (\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)^T \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{AB}^T & \boldsymbol{\Sigma}_{BB} \end{pmatrix}$$

Clearly, $\boldsymbol{\mu}_A \in \mathbf{R}^r$, $\boldsymbol{\mu}_B \in \mathbf{R}^{n-r}$, $\boldsymbol{\Sigma}_{AA} \in \mathbf{R}^r \times \mathbf{R}^r$, $\boldsymbol{\Sigma}_{BB} \in \mathbf{R}^{n-r} \times \mathbf{R}^{n-r}$, $\boldsymbol{\Sigma}_{AB} \in \mathbf{R}^r \times \mathbf{R}^{n-r}$

Some properties of Gaussian distribution

Properties of $\mathbf{x}_A, \mathbf{x}_B$.

- Marginal densities are gaussian with

$$p(\mathbf{x}_A) = \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_{AA}) \qquad p(\mathbf{x}_B) = \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_{BB})$$

- Conditional densities are gaussian with

$$\begin{aligned} p(\mathbf{x}_A|\mathbf{x}_B) &= \mathcal{N}(\boldsymbol{\mu}_A + \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}(\mathbf{x}_B - \boldsymbol{\mu}_B), \boldsymbol{\Sigma}_{AA} - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA}) \\ p(\mathbf{x}_B|\mathbf{x}_A) &= \mathcal{N}(\boldsymbol{\mu}_B + \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}(\mathbf{x}_A - \boldsymbol{\mu}_A), \boldsymbol{\Sigma}_{BB} - \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}\boldsymbol{\Sigma}_{AB}) \end{aligned}$$

Gaussian processes

Multivariate Gaussian distributions: useful for modeling finite collections of real-valued variables. Nice analytical properties.

Gaussian processes: extension of multivariate Gaussians to infinite-sized collections of real-valued variables.

We may think of Gaussian processes as distributions not just over random vectors but over random functions.

Probability distributions over functions with finite domains

Let $\chi = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ be any finite array of elements, and let $\mathcal{H}$ be the set of functions from χ to $\mathbf{R}$

A function $f \in \mathcal{H}$ can be described by the array $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m))$ and any array $(y_1, \ldots, y_m)$ can be seen as the description of a function $f \in \mathcal{H}$ such that $f(\mathbf{x}_i) = y_i$.

The set $\mathcal{H}$ is then in 1-to-1 correspondence with the set of vectors in $\mathbb{R}^m$

A density distribution $p(f), f \in \mathcal{H}$ over functions $f \in \mathcal{H}$, corresponds then to a density distribution $p(\mathbf{x}), \mathbf{x} \in \mathbb{R}^m$

Gaussian distributions over functions with finite domains

If we assume that $p(f)$ is a multivariate Gaussian distribution centered on $\mathbf{0}$ and with diagonal covariance $\sigma^2\mathbf{I}$, it results

$$p(f) = \mathcal{N}(f|\mathbf{0}, \sigma^2\mathbf{I}) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{f(\mathbf{x}_i)^2}{2\sigma^2}}$$

This can be seen as a *prior* distribution of functions, with respect to the observation of any pair $(\mathbf{x}_j, \hat{y}_j)$, $1 \leq j \leq m$.

Gaussian distributions over functions with finite domains

Given a set $I \subset \{1, \ldots, m\}$ of indices, let $\mathbf{X} = \{(\mathbf{x}_j, \hat{y}_j), j \in I\}$ be a set of observed pairs. Then, according to the bayesian framework,

The posterior distribution $p(f|\mathbf{X})$ of functions (wrt to $\mathbf{X}$) can be defined and derived according to Bayes' rule, provided a likelihood model is defined such as $\hat{y} = t + \varepsilon$, $p(\varepsilon) = \mathcal{N}(\varepsilon|0, \beta)$.

The predictive distribution $p(y_k|\mathbf{x}_k, \mathbf{X}) = \displaystyle\int p(y_k|f, \mathbf{x}_k)p(f|\mathbf{X})df$ for any $k \notin I$ can be also derived.

Gaussian distributions over functions with infinite domains

In the case of infinite χ, we have to deal with an infinite collection of random variables.

In this case, the role of multidimensional distributions is covered by stochastic processes.

- A *stochastic process* is a collection of random variables, $\{f(\mathbf{x}) : \mathbf{x} \in \chi$, indexed by elements from some set χ, known as the index set.
- A *Gaussian process* is a stochastic process such that any finite subset of random variables has a multivariate Gaussian distribution.

Gaussian processes

In order to define a Gaussian process, both a mean and a covariance function must be defined.

- a mean function $m(\mathbf{x})$, mapping points in χ onto $\mathbb{R}$
- a kernel function $\kappa(\mathbf{x}_{,1}, \mathbf{x}_2)$, mapping pairs from χ^2 onto $\mathbb{R}$

Given a finite subset $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ of χ, the corresponding set of random variables $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m))$ distribution is given by

$$p(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m) = \mathcal{N}(f|\boldsymbol{\mu}(\mathbf{X}), \boldsymbol{\Sigma}(\mathbf{X}))$$

where $\boldsymbol{\mu}(\mathbf{X}) = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_m))^T$ and $\boldsymbol{\Sigma}(\mathbf{X})_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$.

That is, in a Gaussian process the expectation of $f(\mathbf{x})$ is provided by function $m(\mathbf{x})$ and the covariance between $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$, defined as $E_f[(f(\mathbf{x}_1) - m(\mathbf{x}_1))^2(f(\mathbf{x}_2) - m(\mathbf{x}_2))^2]$, is provided by the kernel function $\kappa(\mathbf{x}_1, \mathbf{x}_2)$.

It can be shown that κ must be such that for any subset $\mathbf{X}$ of χ, the corresponding covariance matrix $\boldsymbol{\Sigma}(\mathbf{X})$ is positive semidefinite (that is, for any $\mathbf{x}$ it must be $\mathbf{x}^T\boldsymbol{\Sigma}(\mathbf{X})\mathbf{x} \geq 0$)

Gaussian processes

Given a Gaussian process $\mathcal{GP}(m, \kappa)$, a function$f$ drawn from it can be intuitively seen as an extremely high-dimensional vector drawn from an extremely high-dimensional multivariate Gaussian.

Each dimension of the Gaussian corresponds to an element x from χ, and the corresponding component of the random vector represents the value of $f(x)$

Using the marginalization property for multivariate Gaussians, we can obtain the multivariate Gaussian density corresponding to any finite subset of variables.

RBF kernel

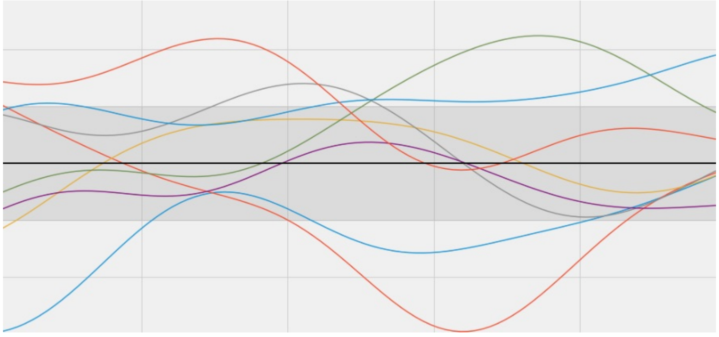
One of the most applied kernel is the RBF kernel

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 e^{-\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{2\tau^2}}$$

which tends to assign higher covariance between $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$ if $\mathbf{x}_1$ and $\mathbf{x}_2$ are nearby points.

Functions drawn from a Gaussian process with RBF kernel tend to be smooth (values computed for nearby points tend to have similar values). Smoothing is larger for larger τ.

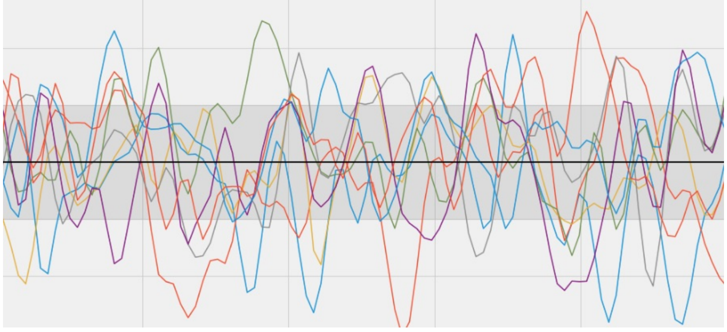
RBF kernel



Larger smoothing

RBF kernel

Smaller smoothing



Gaussian process regression

- Assume, as usual, that observed data derive from function f with an additional gaussian and independent noise.

$$y = f(\mathbf{x}) + \varepsilon \qquad\qquad p(\varepsilon) = \mathcal{N}(\varepsilon|\mathbf{0}, \sigma^2\mathbf{I})$$

  that is, $p(y|f, \mathbf{x}) = \mathcal{N}(y|f(\mathbf{x}), \sigma^2\mathbf{I})$

- A Gaussian process $p(f) = \mathcal{GP}(m, \kappa)$ provides a prior distribution on functions in $\mathcal{H}$.

Gaussian process regression

Given a set of observations $\{\mathbf{X} = \{\mathbf{x}_i\}, \mathbf{y} = \{y_i\}\}, i = 1, \ldots, m\}$ we could now derive a posterior distribution of functions.

It is possible to show that

$$p(f|\mathbf{X}, \mathbf{y}) = \mathcal{GP}(m_p, \kappa_p)$$

with

$$\begin{aligned} m_p(\mathbf{x}) &= k(\mathbf{x}, \mathbf{X})(\mathbf{\Sigma}(\mathbf{X}) + \sigma^2\mathbf{I})^{-1}\mathbf{y} \\ \kappa_p(\mathbf{x}, \mathbf{x}') &= \kappa(\mathbf{x}, \mathbf{x}') - k(\mathbf{x}, \mathbf{X})(\mathbf{\Sigma}(\mathbf{X}) + \sigma^2\mathbf{I})^{-1}k(\mathbf{X}, \mathbf{x}') \end{aligned}$$

where

$$\begin{aligned} k(\mathbf{x},\mathbf{X}) &= \left[\begin{array}{ccc} \kappa(\mathbf{x},\mathbf{x}_1) & \cdots & \kappa(\mathbf{x},\mathbf{x}_m) \end{array} \right] \\ k(\mathbf{X},\mathbf{x}') &= \left[\begin{array}{ccc} \kappa(\mathbf{x}_1,\mathbf{x}') & \cdots & \kappa(\mathbf{x}_m,\mathbf{x}') \end{array} \right]^T = k(\mathbf{x}',\mathbf{X})^T \end{aligned}$$

This makes it possible to sample functions from the posterior distribution

Gaussian process regression

In order to derive the predictive distribution of a new item $\mathbf{x}$, let us observe that for any function f drawn from $\mathcal{GP}(\mathbf{0},\kappa)$, the marginal distribution over any set of input points $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_m)$ is a multivariate Gaussian distribution.

$$p(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_m)) = \mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{\Sigma}(\mathbf{X}))$$

Let $\mathbf{X}_t$ be the set of points in the training set, and let $\mathbf{x}$ be a new item. Then,

$$(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_m),f(\mathbf{x})) \sim \mathcal{N}(\mathbf{0},\mathbf{\Sigma}(\mathbf{X}\cup\{\mathbf{x}\}))$$

where

$$\mathbf{\Sigma}(\mathbf{X}\cup\{\mathbf{x}\}) = \left[\begin{array}{cc} \mathbf{\Sigma}(\mathbf{X}) & k(\mathbf{X},\mathbf{x}) \\ k(\mathbf{x},\mathbf{X}) & \kappa(\mathbf{x},\mathbf{x}) \end{array} \right]$$

Gaussian process regression

By the assumption of gaussian noise

$$\begin{aligned} y_i &= f(\mathbf{x}_i) + \varepsilon \qquad\qquad i = 1,\ldots,m \\ y &= f(\mathbf{x}) + \varepsilon \end{aligned}$$

with $p(\varepsilon) = \mathcal{N}(\varepsilon|0,\sigma^2)$, it derives that

$$(y_1,\ldots,y_m,y) \sim (f(\mathbf{x}_1)+\varepsilon,\ldots,f(\mathbf{x}_m)+\varepsilon,f(\mathbf{x})+\varepsilon) \sim \mathcal{N}(\mathbf{0},\mathbf{\Sigma}(\mathbf{X}\cup\{\mathbf{x}\}+\sigma^2\mathbf{I})$$

From the properties of multivariate gaussian distributions, namely

$$p(\mathbf{x}_B|\mathbf{x}_A) = \mathcal{N}(\boldsymbol{\mu}_B + \mathbf{\Sigma}_{BA}\mathbf{\Sigma}_{AA}^{-1}(\mathbf{x}_A - \boldsymbol{\mu}_A), \mathbf{\Sigma}_{BB} - \mathbf{\Sigma}_{BA}\mathbf{\Sigma}_{AA}^{-1}\mathbf{\Sigma}_{AB})$$

it results

$$p(y|\mathbf{x},\mathbf{X},\mathbf{y}) = \mathcal{N}(y|\overline{\mu}(\mathbf{x}),\overline{\sigma}^2(\mathbf{x}))$$

with

$$\begin{aligned} \overline{\mu}(\mathbf{x}) &= k(\mathbf{x},\mathbf{X})(\mathbf{\Sigma}(\mathbf{X})+\sigma^2\mathbf{I})^{-1}\mathbf{y} = \sum_{i=1}^{m}\alpha(\mathbf{x}_i,\mathbf{x})y_i \\ \overline{\sigma}^2(\mathbf{x}) &= \kappa(\mathbf{x},\mathbf{x}) + \sigma^2 - k(\mathbf{x},\mathbf{X})(\mathbf{\Sigma}(\mathbf{X})+\sigma^2\mathbf{I})^{-1}k(\mathbf{x},\mathbf{X})^T \end{aligned}$$

Estimating kernel parameters

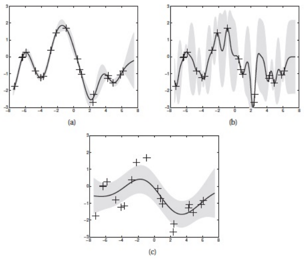
The predictive performance of gaussian processes depends exclusively on the suitability of the chosen kernel.

Let us consider the case of an RBF kernel. Then,

$$\kappa(\mathbf{x}_i,\mathbf{x}_j) = \sigma_f^2 e^{-\frac{1}{2}(\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i-\mathbf{x}_j)} + \sigma_y^2\delta_{ij}$$

M can be defined in several ways: the simplest one is $\mathbf{M} = l^{-2}\mathbf{I}$.

Even in this simple case, varying the values of σ_f,σ_y,l returns quite different results.



(figure from K.Murphy "Machine learning: a probabilistic perspective" p. 519, with (l, σ_f, σ_y) equal to $(1, 1, 0.1), (0.3, 1.08, 0.00005), (3.0, 1.16, 0.89)$)

Estimating kernel parameters

Kernel parameters can be estimated, as usual, through grid search and (cross-)validation.

A different, more efficient approach relies on maximizing the marginal likelihood

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{X}) p(\mathbf{f}|\mathbf{X}) d\mathbf{f} = \int \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{\Sigma}(\mathbf{X})) \prod_{i=1}^{m} \mathcal{N}(y_i | f_i, \sigma_y^2) d\mathbf{f}$$

It can be shown that

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}) &= \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{\Sigma}(\mathbf{X}) + \sigma_y^2 \mathbf{I}) \\ &= -\frac{1}{2} \mathbf{y}^T (\mathbf{\Sigma}(\mathbf{X}) + \sigma_y^2 \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{\Sigma}(\mathbf{X})| + \sigma_y^2 \mathbf{I} - \frac{n}{2} \log(2\pi) \end{aligned}$$

where the first term measures the fitting of the model to data, the second the complexity of the model, and the third + fourth ones are constants