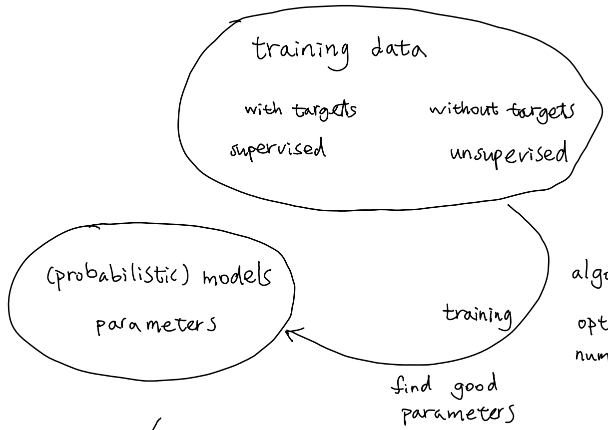


Problem

classification
regression
clustering

algorithms

optimization
numerical methods

Predict

test data

overfitting
regularization
reduce model complexity

Take a look at Appendix C

e.g.

$$(AB)^T = B^T A^T$$

$$AB B^{-1} A^{-1} = I \qquad (AB)^{-1} = B^{-1} A^{-1}$$

for square matrix

trace $\quad Tr(AB) = Tr(BA)$

$$\sum_i (AB)_{ii} = \sum_i \sum_j A_{ij} B_{ji} = \sum_j \sum_i B_{ji} A_{ij} = \sum_j (BA)_{jj}$$

$$Tr(ABC) = Tr(BCA) = Tr(CAB)$$

determinant

$$|A^{-1}| = \frac{1}{|A|}$$

calculus

$$\frac{\partial}{\partial A} Tr(AB) = \begin{array}{cc} & j \\ & \vdots \\ i & \left[\cdots \frac{\partial}{\partial A_{ij}} Tr(AB) \right] \end{array} = B^T$$

$$\frac{\partial}{\partial A_{ij}} Tr(AB) = \frac{\partial}{\partial A_{ij}} \sum_i \sum_j A_{ij} B_{ji} = B_{ji}$$

2009年9月17日
下午 03:47

PRML Ch 2.5

nonparametric methods → use data directly, assume some distance measure

vs. parametric methods → assume a model, find parameters

Central limit theorem

sum of a set of uniformly distributed random variables → Gaussian

See Fig 2.6 or run Matlab

1D histogram

x : continuous variable

partition x into distinct bins of width Δ_i

count the number n_i of observation of x falling in bin i

a normalized probability density is obtained by

$$p_i = \frac{n_i}{N \Delta_i}$$

Simplified $\Delta_i = \Delta$ fixed bin width

$$\lim_{\Delta \to 0} \sum_i p_i \Delta = \lim_{\Delta \to 0} \sum_i \frac{n_i}{N} = 1$$

$$\int p(x)\, dx = 1$$

2009年9月17日
下午 03:57

See Fig 2.24 or Run Matlab

small Δ → spiky

large Δ → smooth

histograms
① quantized
② visualization of data for 1D or 2D
③ discontinuity on bin edge
④ curse of dimensionality

two insights
① local neighborhood for estimating the probability density
"locality" who are your neighbors?

histogram : neighborhood is defined by the bins

② bin width Δ : smoothing parameters
too small or too large is not good
related to regularization , model complexity

Density Estimation (high dimensional)

assume data from some unknown probability density $p(\mathbf{x})$ in D-dimensional Euclidean space

Probability mass with region $\mathcal{R}$

$$P = \int_{\mathcal{R}} p(\mathbf{x})\, d\mathbf{x}$$

each data point has a probability P of falling within $\mathcal{R}$

total number K of points inside $\mathcal{R}$ is a binomial distribution

$$K \sim \mathrm{Bin}(K \mid N, P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K}$$

$$E[K] = \sum_{K=1}^{N} K \cdot \mathrm{Bin}(K \mid N, P) = NP$$

$E[K] = NP$

$$\sum_{K=1}^{N} \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K} = (P + (1-P))^N = 1$$

differentiate

$$\sum_{K=1}^{N} \frac{N}{K!(N-K)!} P^K (1-P)^{N-K} \left\{ \frac{K}{P} - \frac{N-K}{1-P} \right\} = 0$$

multiply $P(1-P)$

$$\sum_{K=1}^{N} \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K} \left\{ K(1-P) - P(N-K) \right\} = 0$$

$$\sum_{K=1}^{N} \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K} \left\{ K - PN \right\} = 0$$

so $E\left[\frac{K}{N}\right] = P$

similarly, variance: $\mathrm{Var}\left[\frac{K}{N}\right] = \frac{P(1-P)}{N}$ (differentiate again)

for large N, we get a distribution sharply peaked around the mean, so

$$K \simeq NP$$

assume $\mathcal{R}$ is small that the probability density $p(\mathbf{x})$ is roughly constant over the region

$P \simeq p(\mathbf{x}) V$ $\quad$ V is the volume of $\mathcal{R}$

we are interested in

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

we have two controdictory assumptions

① $\mathcal{R}$ should be sufficiently small that the density in $\mathcal{R}$ is constant

② $\mathcal{R}$ should be sufficiently large so that the number K of points is sufficient for the binomial distribution to be sharply peaked

$p(\mathbf{x}) = \frac{K}{NV}$ two different approaches

fix K : k-nearest-neighbor

fix V : kernel approaches

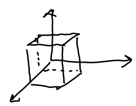
both converge to true probability density
in the limit $N \to \infty$, provided $V\downarrow$ with $N\uparrow$, $K\uparrow$ with $N\uparrow$

Kernel Density Estimation (K.D.E.)

consider the kernel function

$$k(\mathbf{u}) = \begin{cases} 1, & |u_i| < \frac{1}{2}, \quad i=1,\dots,D \\ 0, & \text{otherwise} \end{cases}$$

i.e. R is a unit cube
centered at the origin



The total number K of data points inside a cube of side h
centered on $\mathbf{x}$

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

Therefore

$$p(\mathbf{x}) = \frac{K}{NV} = \frac{1}{N h^D} \sum_{n=1}^{N} k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right)$$

$V = h^D$

artificial discontinuities across the cube boundary

usually we use a Gaussian-like kernel function

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

h is like the standard deviation in Gaussian

other kernel functions are allowable as long as

$$k(\mathbf{u}) \geq 0$$

$$\int k(\mathbf{u})\, d\mathbf{u} = 1$$

$\Rightarrow$ no "training" for kernel density estimation
but the computational cost for testing is hig

Try Fig 2.5 MATLAB

2009年9月17日
下午 05:41

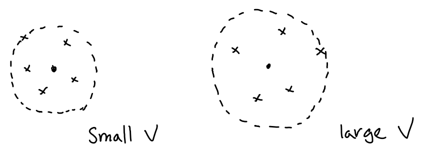
Nearest Neighbor Methods (KNN)

In K.D.E., optimal choice for h may be dependent on location

(there is an issue called "bandwidth selection" in K.D.E.)

we may fix K and use the data to find an appropriate value for V

e.g. $K=5$



KNN can be easily applied to multiclass classification problems

(Homework!)

2009年9月17日
下午 05:48

Consider N_m points in class C_m

$$\sum_{m=1}^{M} N_m = N$$

conditional $\quad p(\mathbf{x} \mid C_m) = \dfrac{K_m}{N_m V}$ $\quad$ (likelihood)

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

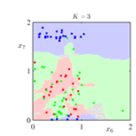
class priors are $\quad p(C_m) = \dfrac{N_m}{N}$

by Baye's theorem

the posterior is $\quad p(C_m \mid \mathbf{x}) = \dfrac{p(\mathbf{x} \mid C_m)\, p(C_m)}{p(\mathbf{x})} = \dfrac{K_m}{K}$

So the decision criterion is very simple :

To classify a new point, we find the K nearest neighbor points from the training data and assigned the new point to the class having the largest number of representatives among the k nearest neighbors (the largest K_m)



$N \to \infty$, the error rate is never more than twice minimum achievable error rate of an optimal classifier, i.e., one that uses the true class distributions.

① require entire training set to be stored

② the computational cost may be reduced by approximate nearest neighbor techniques