

CS4770

Pattern Recognition

Non-Parametric Methods

P. J. Narayanan
Monsoon 2006

Unknown Parameters

- We assumed the form of the likelihoods $p(\mathbf{x}|\omega)$ are known.
- This is not often the case in practice.
- Neither does real probabilities follow nice forms like Gaussians. They have complicated, irregular forms.
- But they **do have a form** which are expressed through experiments or sampling.
- Can we estimate likelihoods from training samples?
Can we directly estimate the posteriors?

Estimating Densities

- The probability that the random variable \mathbf{x} falls in a region R is: $P = \int_R p(\mathbf{x})dx$.
- If among n trials, k_n samples fall in the region R , we can approximate $P \approx \frac{k_n}{n}$.
- If V is the volume of the region, P/V is a good estimate of the probability density $p(\mathbf{x})$.
- We can write:

$$p(\mathbf{x}) = \lim_{V \rightarrow 0} \frac{k_n/n}{V} = \lim_{V \rightarrow 0} \frac{k_n}{nV}$$

Additional Constraints

- V can't be so small independently. Else k_n will be 0!
- n needs to go to ∞ and k_n needs to go to ∞ .
- Let us define regions R_1, R_2, \dots, R_n where R_i is the region after i training data are obtained.
- V_i is the corresponding volume and $p_n(\mathbf{x})$ is the estimate of $p(\mathbf{x})$. $p_n(\mathbf{x}) = \frac{k_n}{nV_n}$.
- Constraints for getting $p_n(\mathbf{x}) \rightarrow p(\mathbf{x})$ as $n \rightarrow \infty$ are:
$$V_n \rightarrow 0, \quad k_n \rightarrow \infty, \quad nV_n \rightarrow \infty.$$

Two Approaches

- Shrink V_n with n such that $nV_n \rightarrow \infty$.
For a given \mathbf{x} , fix a volume of V_n on \mathbf{x} , count the number of points k_n and calculate $p_i(\mathbf{x})$.

Parzen Windows

- Expand k_n with n such that $k_n/n \rightarrow 0$.
For a given \mathbf{x} , grow the volume around it till k_n points are included. Compute $p_n(\mathbf{x})$.

k_n -Nearest Neighbours

Parzen Windows

- Choose a series of **window functions** that define the volume V_n .
- Each training sample exerts its influence in this volume according to some drop-off formula. It describes the local probability distribution around the sample.
- Such volumes are centered around each sample and summed to give the estimate $p_n(\mathbf{x})$.
- Example: Region R_n is a d -dimensional hypercube of diminishing size. Region R_n is a Gaussian with zero mean and unit variance.

Hypercube Windows

- Let $V_n = h_n^d$ and $\varphi(\mathbf{u}) = 1$ within the unit hypercube centered at the origin.
- Now, $k_n = \sum_{i=1}^n \varphi(\mathbf{u}_n) d\mathbf{u}_n$ where $\mathbf{u}_n = (\mathbf{x} - \mathbf{x}_i)/h_n$.

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_n} \varphi(\mathbf{u}_n) = \frac{1}{n} \sum_{i=1}^n \delta_n(\mathbf{x} - \mathbf{x}_i)$$

- $\int \delta_n(\mathbf{x} - \mathbf{x}_i) d\mathbf{x} = \int \varphi(\mathbf{u}) d\mathbf{u} = 1$. $\delta_n \rightarrow \delta$ as $h_n \rightarrow 0$.
- h_n affects the width (through window φ) and the amplitude of δ_n through V_n .

- Large V_n : Each sample has large influence.
Small V_n : Effect of each sample pronounced on $p(\mathbf{x})$.
- Select an initial h_1 . Define h_n with respect to it, such as $h_n = h_1/\sqrt{n}$.

Gaussian Windows

- The window function could be: $\varphi(\mathbf{u}_n) = \frac{1}{\sqrt{2\pi}}e^{-\mathbf{u}_n^2/2}$
- If $\mathbf{u}_n = (\mathbf{x} - \mathbf{x}_i)/h_n$, the volume of the region is h_n .
- $p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$
- Select an initial h_1 . Define h_n with respect to it, such as $h_n = h_1/\sqrt{n}$.

Convergence Properties

- $p(\mathbf{x})$ is a sum of functions of random variables. We can define its mean $\bar{p}_n(\mathbf{x})$ and variance σ_n^2 .
- We can show:
$$\lim_{n \rightarrow \infty} \bar{p}_n(\mathbf{x}) = p(\mathbf{x}) \quad \text{and} \quad \lim_{n \rightarrow \infty} \sigma_n^2(\mathbf{x}) = 0$$
- Thus, $p(\mathbf{x})$ can be estimated by this technique.
- The $p()$ can have any shape, number of peaks etc.
- Will need a large number of samples for converging to the correct density.

Parzen Windows: Summary

- Nothing is known about the probability distribution except for a number of samples generated according to it!
- Each sample extends its influence over a small region of certain shape around it, the size of which depends on the number of samples).
- The net influence at a point is the sum of all influences from all samples. We can think of this as a functional form or convolution.
- As the number of samples increases, the volume of

influence reduces, approximating a δ function in the limiting case.

- Can approximate any arbitrary probability distribution in the asymptotic situation.

Probabilistic Neural Network (PNN)

- We can “train” a neural network using a simple procedure.
- A network with d input nodes, n pattern nodes and c category nodes.
- $w_{ij} = x_{ij}$, $1 \leq i \leq n$, $1 \leq j \leq d$. w_{ij} is the weight between input node j and intermediate node i . x_{ij} is the j th component of sample \mathbf{x}_i .
- $a_{ik} = 1$ iff $\mathbf{x}_i \in \omega_k$, $1 \leq i \leq n$, $1 \leq k \leq c$. a_{ik} is the weight between intermediate node i and output node k .

PNN Classification

- Each pattern node j computes its net α_j as $\mathbf{w}_j^T \mathbf{x}$ for input \mathbf{x} .
- If φ is a Gaussian, $\varphi((\mathbf{x} - \mathbf{x}_k)/h_n) = e^{-(\mathbf{x} - \mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k)/(2\sigma^2)}$. This equals $e^{(\mathbf{w}^T \mathbf{x} - 1)/\sigma^2}$ since \mathbf{x}, \mathbf{x}_i are normalized.
- Category node k accumulates $e^{(\alpha_j - 1)/\sigma^2}$ into g_k if $a_{kj} = 1$.
- Classify \mathbf{x} as belonging to class $\arg \max_k g_k(\mathbf{x})$.

k_n Nearest Neighbours

- We saw: $p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$. $p_n(\mathbf{x}) \rightarrow p(\mathbf{x})$ as $V_n \rightarrow 0$
- Parzen Windows: Use increasingly smaller volumes V_n with more samples.
- Let window size be data dependent. Increase V_n to contain k_n samples. Then use the formula for $p_n(\mathbf{x})$.
- We must have: $k_n \rightarrow \infty$ as $n \rightarrow \infty$ such that $nV_n \rightarrow \infty$.
- $k_n = \sqrt{n}$, $k_n = \ln n$, etc., will work.

Distances and Metrics

- What is a good distance to be used for such algorithms?
- **Metric:** If $D(\mathbf{a}, \mathbf{b})$ obeys: *non-negativity, reflexivity, symmetry and triangle inequality.*
- **Minkowski metric:** $L_k(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^d |a_i - b_i|^k \right)^{1/k}$
- $L_1 \equiv$ Manhattan distance, $L_2 \equiv$ Euclidean distance, etc.
- Different invariances for these distances.

Properties

- If the neighbourhood of \mathbf{x} is dense in samples, V_n would be small to contain k_n samples.
- Initial volume is not an arbitrary choice.
- What is k_n ? Can write $k_n = k_1\sqrt{n}$. Then the probability recovered depends on the initial value k_1 .

A *Posteriori* Probabilities

- Volume V_n contains k_n samples, of which k_i belongs to the class ω_i .
- $p_n(\mathbf{x}, \omega_i) = (k_i/n)/V_n$.
We can get:
$$P(\omega_i|\mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_{j=1}^c p_n(\mathbf{x}, \omega_j)} = \frac{k_i}{k_n}$$
- We can directly estimate the posterior probabilities!
- Posterior probability is the fraction of the samples within V that belongs to ω_i .

Nearest Neighbour Classifier

- What if $k_n = 1$? Choose the class for \mathbf{x} as the class of its closest neighbour \mathbf{x}' .
- This rule partitions space into *cells* of points which are close to a sample or prototype point.
- Partitions space as per **Vornoi tessellation**.
- $\Pr(\mathbf{x}' \in \omega_i) = P(\omega_i|\mathbf{x}')$. With large n , \mathbf{x}' and \mathbf{x} are very close. Hence $P(\omega_i|\mathbf{x}') = P(\omega_i|\mathbf{x})$.
- Thus, nearest neighbour rule tries to match probabilities.
- Bayesian rule selects m for which $P(\omega_m|\mathbf{x})$ is maximum.

Error in Nearest Neighbour

- Error $P_n(e|\mathbf{x})$ depends on the specific set of n samples. By averaging over all possible sets of n samples, we can get a conditional error $P(e|\mathbf{x})$.
- The total error $P_n(e)$ as $\int P(e|\mathbf{x})p(\mathbf{x})d\mathbf{x}$.
- Minimum (Bayesian) error: $P^*(e|\mathbf{x}) = 1 - P(\omega_m|\mathbf{x})$.
- Minimum possible $P(e)$ is given by $P^* = \int P^*(e|\mathbf{x})p(\mathbf{x})d\mathbf{x}$.
- Result: $P^* \leq P \leq 2P^*$, where $P = P_n(e)$ as $n \rightarrow \infty$.

- Maximum error is when $P(\mathbf{x}|\omega_i)$ are all equal.
- In the infinite sample case, a complicated decision rule will cut the error at best into half over a nearest neighbour rule.
- Thus, about half the classification information resides in the nearest neighbour.
- No guarantees on finite sample performance, however.

k -Nearest-Neighbour Rule

- Choose majority label from k nearest neighbours of \mathbf{x}
- Matching the probabilities is done more strongly with this rule than with the nearest neighbour rule.
- With nearest neighbour, $P(\omega_m|\mathbf{x})$ was the probability that the Bayesian label is selected.
- With k -NN rule, the probability of selecting ω_m is:

$$\sum_{i=(k+1)/2}^k C_i^k P(\omega_m|\mathbf{x})^i [1 - P(\omega_m|\mathbf{x})]^{k-i}$$

which increases with k .