(1) Spring 2014

Non-Parametric Learning

Note Title 10/28/2006

The previous lecture described learning parametric probability models. In particular, exponential models. — $P(x|\lambda) = \frac{1}{Z[\lambda]} \exp\langle \lambda \cdot \phi(\vec{x})\rangle$.

Now we consider non-parametric models.

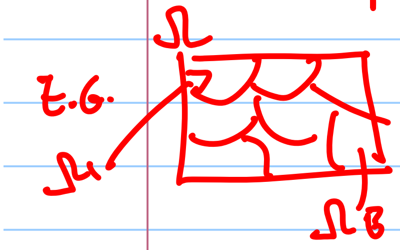
We have already disussed two non-parametric models:

(i) The empirical distribution : $f(x) = \frac{1}{N} \sum_{i=1}^{N} I(x-x_i)$
for dataset $\mathcal{X} = \{x_i : i = 1 \text{ to } N\}$

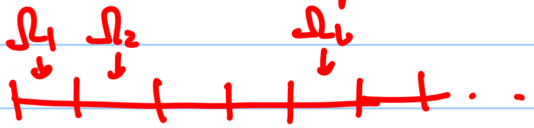
(ii) The histogram:

Divide X space into B domain $\Omega_1, .., \Omega_B$ s.t. $\bigcup_{i=1} \Omega_i = \Omega_x$ ← entire space → with $\Omega_i \cap \Omega_j = \phi$ for $i \neq j$

E.G.



or



Then $P(x) = n_b/n$, where n_b is no. counts in region Ω_b.
i.e. $n_b = \sum_{i=1}^{n} I(x_i \in \Omega_b)$.

Note: technically the histogram can also be thought as a parametric model. The counts n_b and the indicator values are sufficient statistics. It can be expressed as exponential.

More general : $P(x) = \frac{1}{n} \sum_{i=1}^{n} \text{win}(x-x_i)$ ← window.

eg. empirical distribution if $\text{win}(x-x_i) = I(x-x_i)$.

(2) Why non-parametric?

It is hard to develop parameterized probability models for some data.

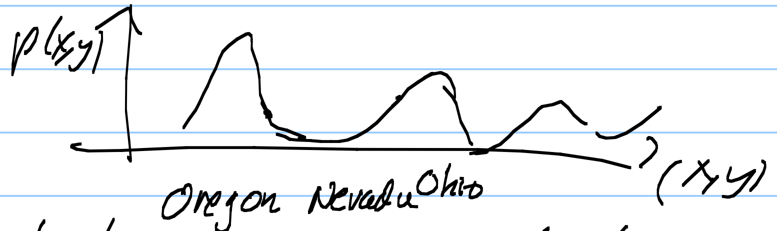
Example: estimate the distribution of the annual rainfall in the USA.

Goal – model p(x,y) – the probability that a raindrop hits a position (x,y)

(E.G. low in the Mohave desert, high in Hawaii)

It is hard to see how to model a multi-modal distribution like this

– Intuitively

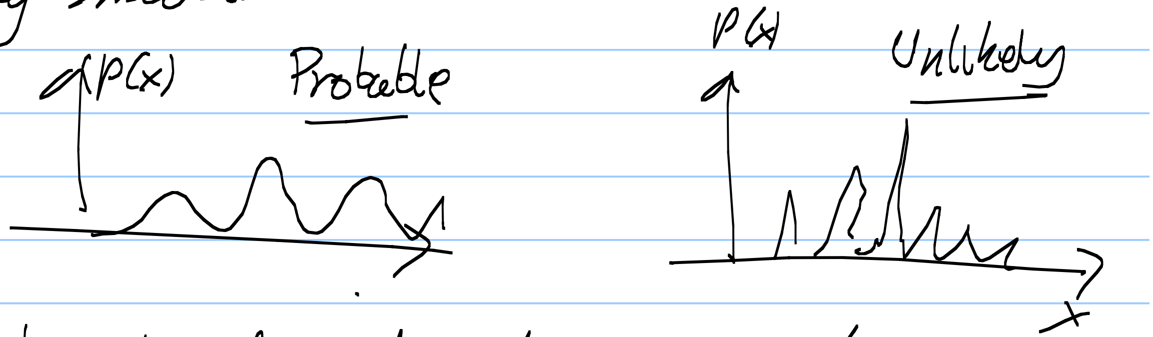


((But see later lecture on exponential models with hidden variables.)

(3.) Intuition for window function.

Assume that the probability distribution is locally smooth.

i.e.



Method 1: Windows based on points $\underline{x}$ in space.

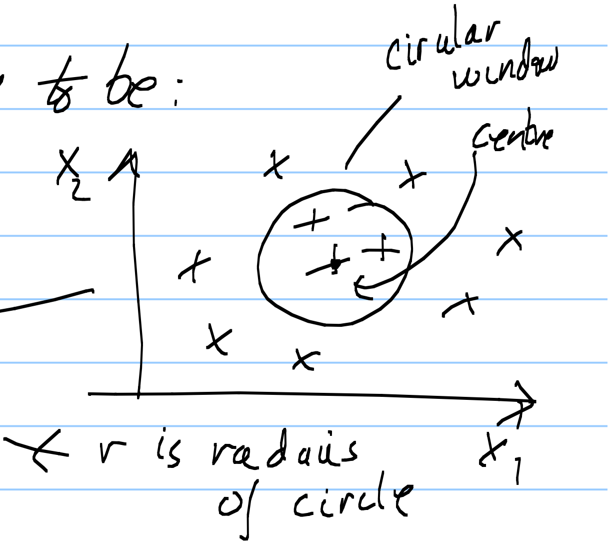
For each point $\underline{x}$, form a window centred on $\underline{x}$ with volume V_n. Count the number of samples k_n that fall in the window.

Estimate probability density to be:

$$p_n(x) = \frac{k_n}{nV_n}$$

(Smooth in scale of window.)

e.g. $k_n = 3$, $V_n = \pi r^2$



(4) Goal: to design a sequence of windows $V_1, \ldots V_n$ so that at each point x, $p_n(x) \rightarrow p(x)$ as $n \rightarrow \infty$

$\overset{\wedge}{\text{true distribution}}$ (under $p(x)$)

(recall, n is the no. samples)

Conditions for window design:

(i) Increasing spatial resolution
$$\lim_{n \to \infty} V_n = 0$$

(ii) Many samples at each point
$$\lim_{n \to \infty} k_n = \infty \quad , (\text{provided } p(x) \neq 0)$$

(iii) $$\lim_{n \to \infty} k_n/n = 0$$
i.e. k_n grows slower than n.

Two Design Methods.

(A) Parzen Windoos :

Fix the window size : $V_n = 1/\sqrt{n}$!

(B) K-NN :

Fix no. samples in window : $k_n = \sqrt{n}$!

(adaptive)

(A) Parzen Window

Uses a window function $\phi(\underline{u})$ s.t. $\phi(\underline{u}) \geq 0$, $\int \phi(\underline{u}) d\underline{u} = 1$.

Examples :

(i) Unit hypercube : $\phi(\underline{u}) = 1$, if $|\underline{u}| < 1/2$ and $\phi(\underline{u}) = 0$ otherwise.

(ii) Gaussian in d-dimensions.

$$\phi_G(\underline{u}) = \frac{1}{(2\pi)^{d/2}} e^{-\underline{u}^T \cdot \underline{u}/2}.$$

h_n is the scale factor.

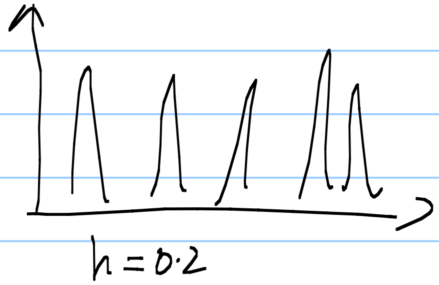
No. of samples in the hypercube centered on x is $k_n = \sum_{i=1}^{n} \phi\left(\frac{\underline{x} - x_i}{h_n}\right)$.

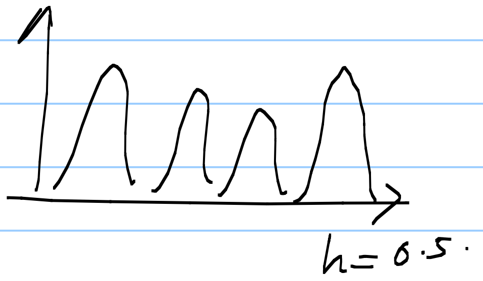
Volume $V_n = h_n^d$

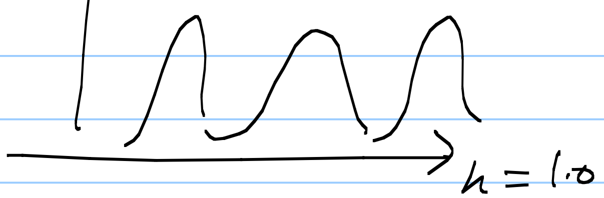
Estimated Density : $p_n(\underline{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \phi\left(\frac{\underline{x} - \underline{x}_i}{h_n}\right)$

(6) Parzen window example:

Gaussian window







True distribution has three models.
For small h, the Parzen window is too small and yields a distribution with too many modes.

Parzen Window Convergence Theorem

$$\lim_{n \to \infty} p_n(x) = p(x) \quad \text{(True Density)}$$

Hence the Parzen window estimator converges to the true density at each point x with increasing no. of samples.

Comment: It is good to have consistency as $n \to \infty$, but behaviour for small n is more important

$E\{.,.\}$ is expectation w.r.t. $p(\underline{x})$

(7) Proof of Convergence Theorem

Parzen density $p_n(\underline{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{V_n}\phi\left(\frac{\underline{x}-\underline{x}_i}{h_n}\right)$

is a random variable which depends on the observed samples $\underline{x}_1,\dots,\underline{x}_n$ from $p(\underline{x})$.

$$\hat{p}_n(\underline{x}) = E\{p_n(\underline{x})\} = \frac{1}{n}\sum_{i=1}^{n} E\left\{\frac{1}{V_n}\phi\left(\frac{\underline{x}-\underline{x}_i}{h_n}\right)\right\}$$

$$= \int dy\, p(\underline{y})\frac{1}{V_n}\phi\left(\frac{\underline{x}-\underline{y}}{h_n}\right)$$

As $n \to \infty$ $\frac{1}{V_n}\phi\left(\frac{\underline{x}-\underline{y}}{h_n}\right) \mapsto \delta(\underline{x}-\underline{y})$ ← Dirac delta function.

So $\lim_{n\to\infty} E\{p_n(\underline{x})\} = p(\underline{x})$ – consistency

To complete proof, must show that the variance of the estimate of $p_n(\underline{x})$ tends to zero as $n \to \infty$.

$$\sigma_n^2(\underline{x}) = \sum_{i=1}^{n} E\left\{\left(\frac{1}{nV_n}\phi\left(\frac{\underline{x}-\underline{x}_i}{h_n}\right) - \frac{1}{n}\hat{p}_n(\underline{x})\right)^2\right\}$$

Note: this is correct only in the limit as $n \to \infty$ →

$$= \frac{1}{n}\left\{E\left\{\frac{1}{V_n^2}\phi^2\left(\frac{\underline{x}-\underline{x}_i}{h_n}\right)\right\} - \left(E\{p_n(\underline{x})\}\right)^2\right\}$$

$$= \frac{1}{nV_n}\int \frac{1}{V_n}\phi^2\left(\frac{\underline{x}-\underline{y}}{h_n}\right)p(\underline{y})\,dy - \frac{1}{n}\left(E\{p_n(\underline{x})\}\right)^2$$

Note: upper bound the first term.

$$\leq \frac{\sup(\phi(.))\,\bar{p}_n(\underline{x})}{nV_n} \mapsto 0, \quad n \to \infty.$$

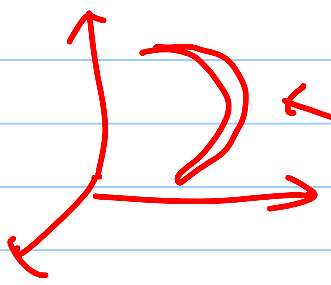
(8) Parzen Windows in Practice

In practice, we do not have an infinite number of samples

The choice of window shape and size is important. It interpolates the data.

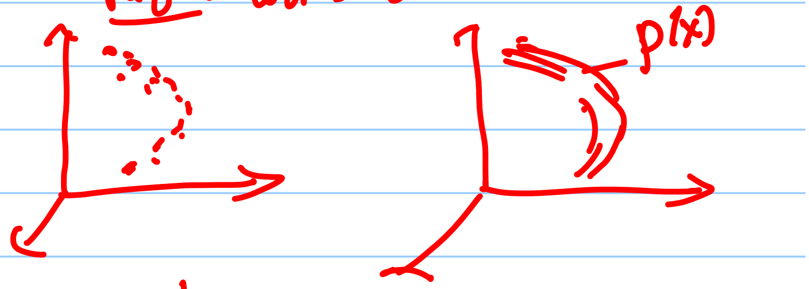
If the window shape and size fits the local structure of the true probability density, then Parzen windows are effective.

Example 1:

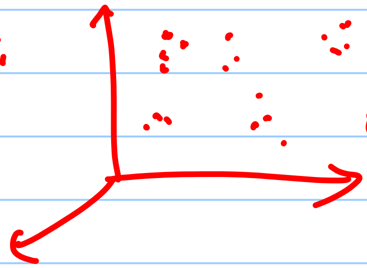


High-Dim space. Data lies on low-dim submanifold

Parzen works well.



Example 2:



← Parzen may be ineffective. Too few data points

Alternative Strategy.

K-Nearest Neighbours. K-NN.

Fix no. samples inside window

$k_n = \sqrt{n}$, $V_n = V_n(\underline{x})$ — function of $\underline{x}$

(vary size of window until it contains n samples)

$$p_n(\underline{x}) = \frac{k_n/n}{V_n} = \frac{1}{V_n(\underline{x})\sqrt{n}}$$

Advantages & Disadvantages.

Plus. The adaptive size of the window means that $p_n(\underline{x})$ will never be zero. This is an advantage in high dimensions.

Minus. Possibly enormous variation in window size. E.g. big in some parts of the space, but small in others.

Also distribution may not be normalizable. (i.e. can't have $\sum_x p(x) = 1$)

E.G for $n=1$, $p_n(\underline{x}) = \frac{1}{2|\underline{x}-\underline{x}_1|}$

$p_n(\underline{x})$ is not normalizable.

$p_n(\underline{x})$ will remain unnormalizable as the number of samples increases.

The Nearest Neighbour Decision Rule

Non-Parametric Classification

Suppose we have n samples $\mathcal{X} = \{\underline{x}_1, \ldots, \underline{x}_n\}$
and c classes $\omega_1, \omega_2, \ldots, \omega_c$

n_1 samples in class ω_1
n_2 " " " ω_2
n_c " " " ω_c

window V_n at $\underline{x}$ contains k_i samples in class ω_i

$$\sum_{i=1}^{c} n_i = n$$

Use non-parametric probabilities.

$$p(\underline{x}|\omega_i,\mathcal{X}) = \frac{k_i/n_i}{V_n}.$$

Total $k = \sum_i k_i$ samples in window.

The posterior probability

$$p(\omega_i|\underline{x},\mathcal{X}) = \frac{p(\underline{x}|\omega_i,\mathcal{X})\, p(\omega_i|\mathcal{X})}{\sum_{j=1}^{c} p(\underline{x}|\omega_j,\mathcal{X})\, p(\omega_j|\mathcal{X})}$$

Prior probabilities $p(\omega_i|\mathcal{X}) = n_i/n.$

Hene $P(\omega_i|\underline{x},\mathcal{X}) = k_i/k$

the fraction of samples within window that are labelled ω_i.

Bayes Decision Rule

$$\omega^*(x) = \underset{i}{\text{ARG MAX}} \{k_1, .., k_c\}$$

Bayes Decision Rule for Non-parametric distribution indicates that we can go directly for the decision rule – and byepass the estimation of $p(\omega_i|X)$ and $p(\underline{x}|\omega_i, X)$.

This gives the nearest neighbor NN decision rule.

Partition the space into c disjoint subspaces $\Omega = \bigcup_{i=1}^{c} \Omega_i$, $\Omega_i \cap \Omega_j = \phi$, $i \neq j$.

(The Ω_i's may not be simply connected).

NN decision rule:

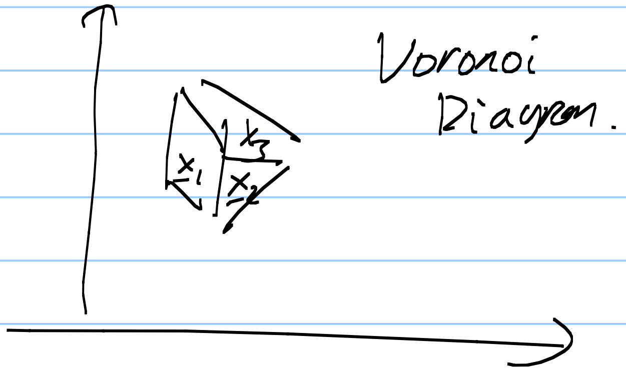
Let $\{(\underline{x}_1, \omega(\underline{x}_1)), \ldots (\underline{x}_n, \omega(\underline{x}_n))\}$ be the labelled samples.

$$\omega_{NN}(\underline{x}) = \omega(\underline{x}^*), \quad \underline{x}^* = \text{ARG MIN}_j \{|\underline{x}_i - \underline{x}_j| : j = 1 \text{ to } n\},$$

($\omega(\underline{x}^*)$ is the class of $\underline{x}^*$).

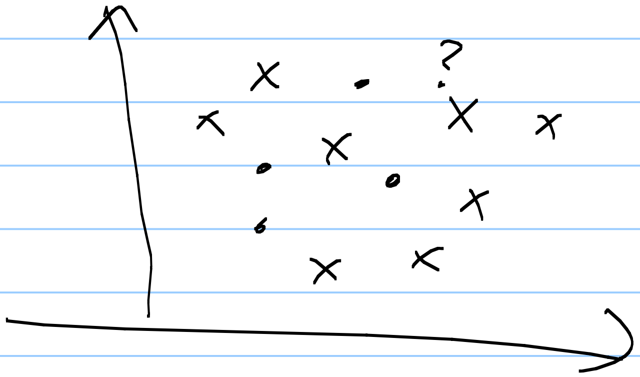
(12)

NN – partition the space into a Voronoi diagram., where each sample $\underline{x}_i$ occupies a cell



The NN decison rule is very intuitive

Label an unknown point ?, by the label of the closest data point.



To improve NN – go to k-NN

For $\underline{x}$, assign the labels that is most common of the k-nearest samples.

Find $k_1, .. k_c$ s.t. $\sum_{i=1}^{c} k_i = k$

counts of nearest samples in each class

Find $j = \underset{i}{\mathrm{ARGMAX}}\, k_i$

$\hat{\omega}(x) - \omega_j$

Asymptotic Analysis of NN

For large N, the performance of NN can be calculated. It is worse than the optimal Bayes classifier by a fixed amount.

Let $P_n(e|x)$ be the error rate at x based on a NN classifier with n samples.

Then $$P_n(e|x) = \int p_n(e, x^* | x)\, dx^*$$
$$= \int P_n(e | x^*, x)\, p(x^* | x)\, dx^*$$

where x^* is the point in the samples which is closest to x. x^* is a random variable which depends on the samples, so we must average over $p(x^*|x)$.

As $n \to \infty$ $p(x^*|x) = \delta(x - x^*)$, the nearest sample to x is arbitrarily close.

Now $$P_n(e | x^*, x) = 1 - \sum_{i=1}^{c} P(\omega_i | x^*)\, P(\omega_i | x)$$

(error occurs if x^* & x have different labels.)

We can write.

$$P_n(e|x) = \int \left\{ 1 - \sum_{i=1}^{c} P(\omega_i|x^*)\, P(\omega_i|x) \right\} P(x^*|x)\, dx^*$$

$$\lim_{n\to\infty} P_n(e|x) = \int \left[1 - \sum_{i=1}^{c} P(\omega_i|x^*)\, P(\omega_i|x) \right] \delta(x-x^*)\, dx^*$$

$$= 1 - \sum_{i=1}^{c} P^2(\omega_i|x).$$

The expected error rate is

$$P = \lim_{n\to\infty} \int P_n(e|x)\, p(x)\, dx$$

$$= \int \left\{ 1 - \sum_{i=1}^{c} P^2(\omega_i|x) \right\} p(x)\, dx$$

Now we want to bound this error in terms of the best (Bayes) error rate. P^*.

Claim: $P^* \le P \le P^*\left(2 - \frac{c}{c-1} P^*\right)$

To justify this claim,

let $\omega_m = \omega_{Bayes}(x)$

so $P^*(e|x) = 1 - P(\omega_m|x)$

(15)

Write.

$$\sum_{i=1}^{c} P^2(\omega_i|x) = P^2(\omega_m|\underline{x}) + \sum_{i\neq m} P^2(\omega_i|x)$$

$$= \{1-P^*(e|x)\}^2 + \sum_{i\neq m} P^2(\omega_i|x)$$

We bound this by minimizing $\sum_{i\neq m} P^2(\omega_i|x)$ subject to the constraint that $\sum_{i\neq m} P(\omega_i|x) = P^*(e|x)$.

This minimization occurs with $P(\omega_i|x) = \frac{P^*(e|x)}{c-1}$, for all i

<u>Hence</u> $\sum_{i=1}^{c} P^2(\omega_i|x) \geq (1-P^*(e|x))^2 + \frac{P^{*2}(e|x)}{c-1}$

which implies $1 - \sum_{i=1}^{c} P^2(\omega_i|x) \leq P^*(e|x) \left\{2 - \frac{c}{c-1} P^*(e|x)\right\}$

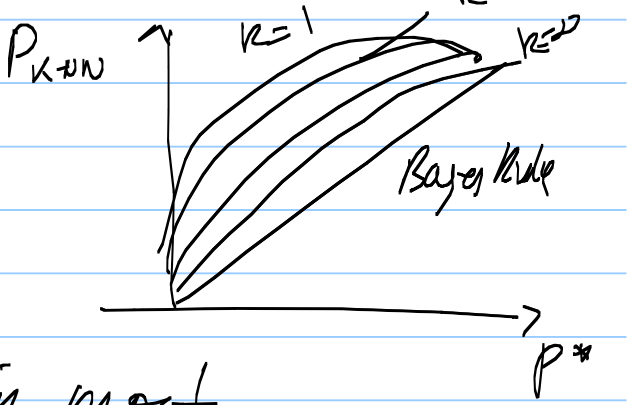
The claim follows after integrating. (using $\int (P^*(e|x))^2 p(x)dx \geq \left\{\int P^*(e|x)p(x)dx\right\}^2$

<u>Comment</u>, the error bound of NN-rule reaches P^* in two extreme cases:

(1) when $P = P^* = \frac{c-1}{c}$, No information

(2) when $P = P^* = 0$, No uncertainty.

The asymptotic performance of k-NN gets closer to the Bayes Risk as k increases.



But, one again, in most situations we do not have an infinite amount of data.

Performance for small n is important. Hard to analyse. Validate by comparison of performance on training and testing datasets.

Note: recent research concentrates on algorithms for efficiently finding the nearest neighbors. I.e. representing the dataset in such a way that this can be done rapidly. E.g. by coarse-to-fine search.

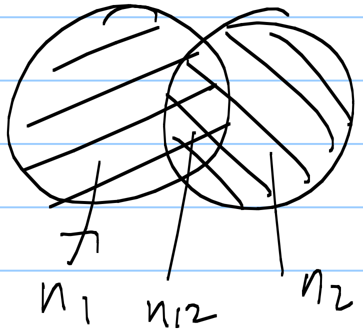
Distance Measures for NN.

Minkowski:

$$D(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^k \right)^{1/k}.$$

Tanimoto metric for sets.

$$D(S_1,S_2) = \frac{n_1 + n_2 - 2n_{12}}{n_1 + n_2 - n_{12}}$$



Transform Distance:

$\boxed{8}$ $\boxed{5}$ $\boxed{5}$

The $\boxed{5}$ may be closer to the $\boxed{8}$ than to the transformed $\boxed{5}$

Apply set of transformations G

$$D(x,y) = \min_{a \in G} \| f(x:a) - y \|$$

e.g. rotation
scaling
translation

Tangent Distance: $D(x,y) = \min_{a \in G} \| x + Ta - y \|$

linear expansion.

(18] How to Find Nearest Neighbors Quickly?

Suppose there is a lot of data in a high-dimensional space. i.e. $X_N = \{\underline{x}^1, \underline{x}^2, \dots \underline{x}^N\}$,

N is very big, 100's, 1000's ..

$\underline{x}^1 = (x^1_1, \quad , x^1_M)$, M is big (dim of space)

(common situation)

Then it can be really expensive to calculate the nearest neighbors — have to search over N datapoints

Solutions – organize the data efficiently so that the search can be done quickly (Google does this).

Example → ANN methods. See more in latex notes and references to online material.