STATISTICAL LINEAR REGRESSION - PART IV:

K-NN FOR REGRESSION

(Rafael Alcalá)

Bibliography:

PAPER para k-NN: Hechenbichler K. and Schliep K.P. (2004) Weighted k-Nearest-Neighbor Techniques and Ordinal Classification, Discussion Paper 399, SFB 386, Ludwig-Maximilians University Munich (http://www.stat.uni-muenchen.de/sfb386/papers/dsp/paper399.ps)

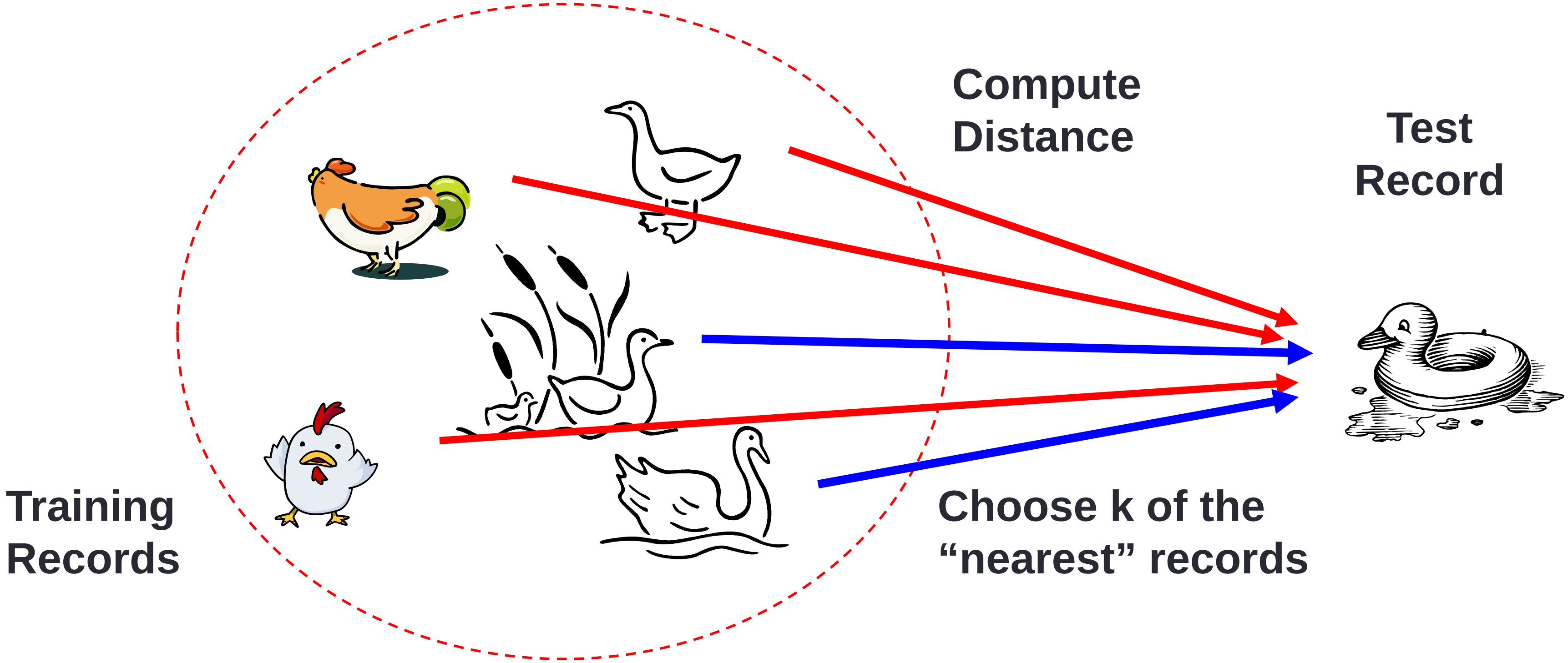
PAPER para k-NN: Samworth, R.J. (2012) Optimal weighted nearest neighbour classifiers. Annals of Statistics, 40, 2733-2763. (http://www.statslab.cam.ac.uk/~rjs57/Research.html)

Outline

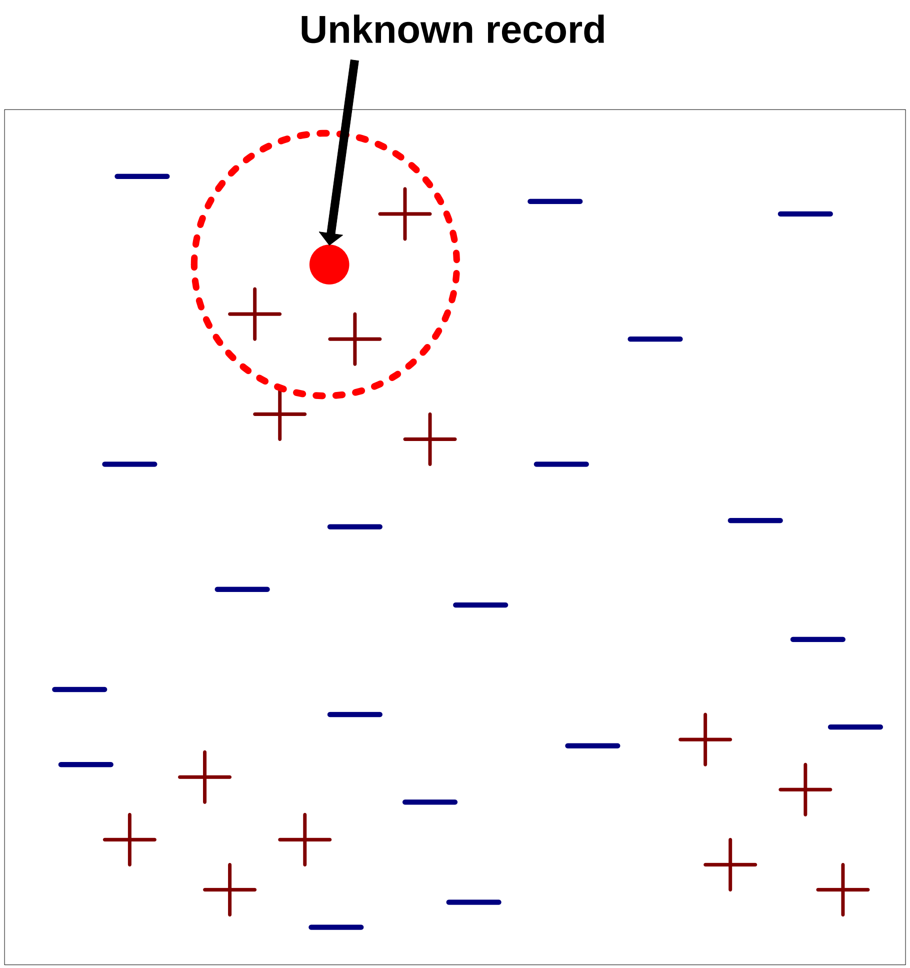
- The Basics of k-NN
  - k-NN for Classification (the initial idea)
  - k-NN for Regression (continuous target variables)
- Distance Weighted k-NN
- Common Kernel Types

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



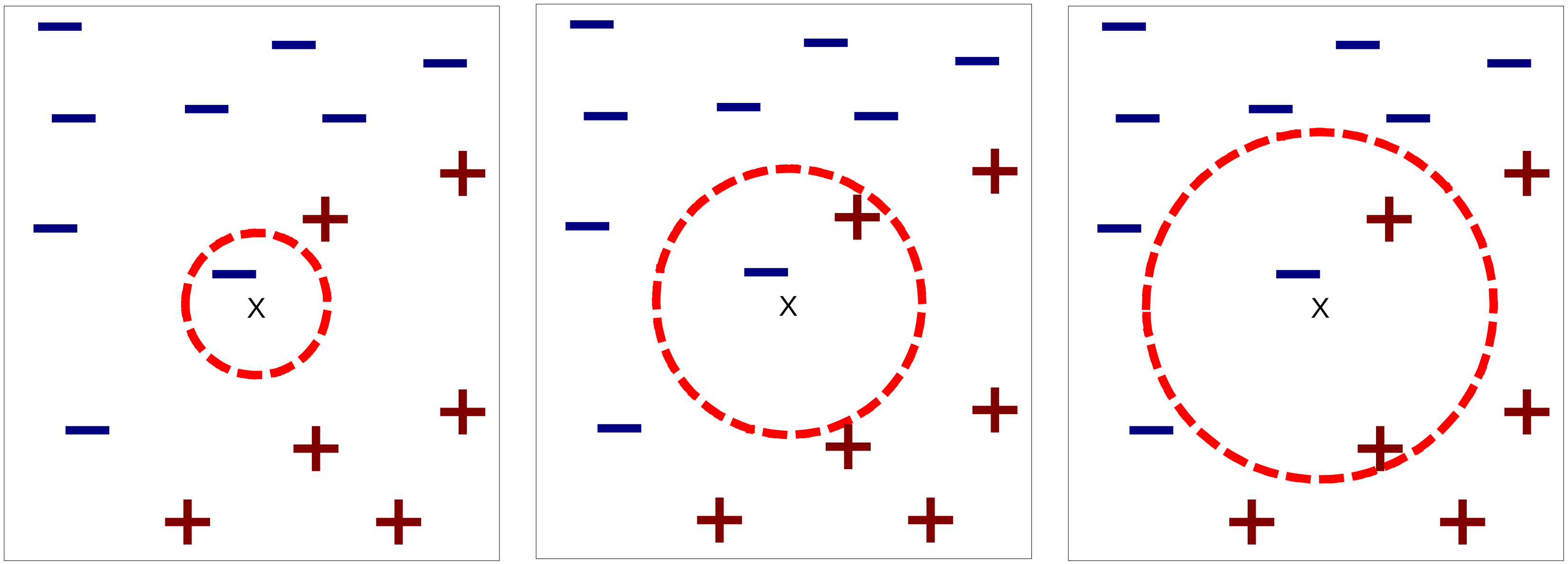
Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Basic k-NN algorithm (Discrete and Continuous target)

Key idea: just store all training examples $<x_i,f(x_i)>$

Nearest neighbor:

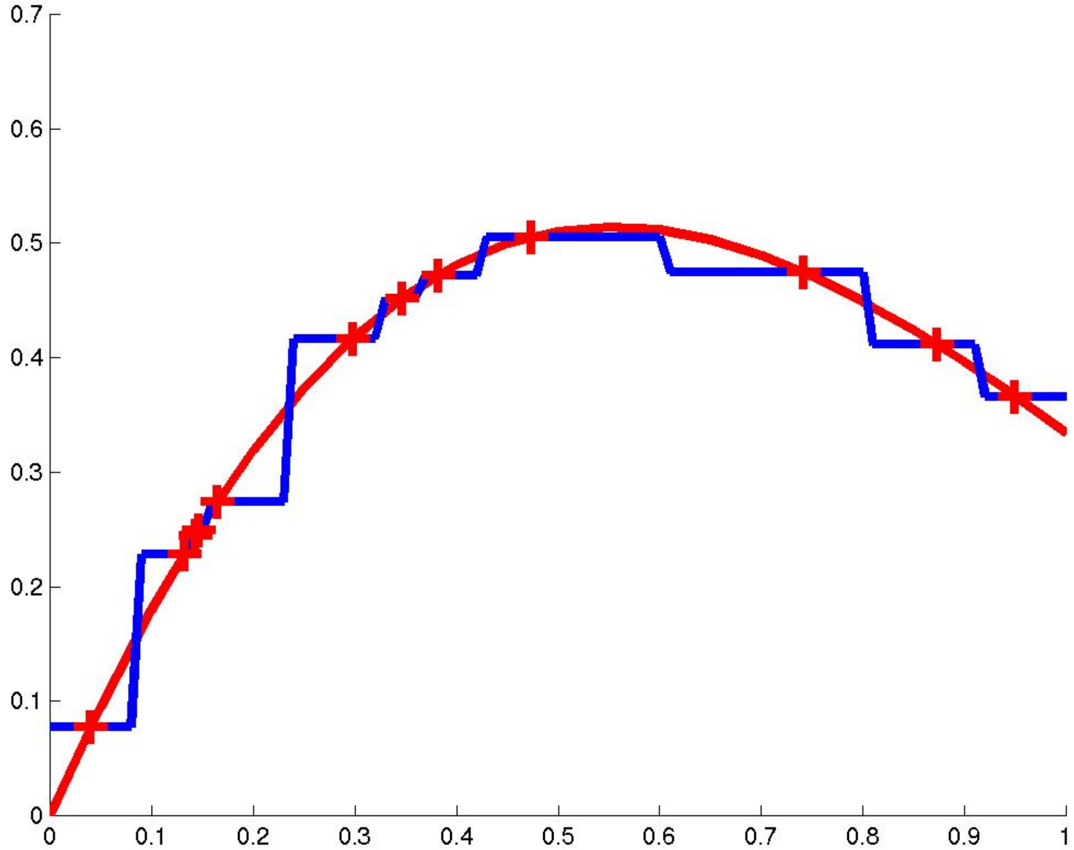
- Given query instance x_q, first locate nearest training example x_n, then estimate $f(x_q)=f(x_n)$

K-nearest neighbor:

- Given x_q, take vote among its k nearest neighbors (if discrete-valued target function)
- Take mean of f values of k nearest neighbors (if real-valued) $f(x_q)=\sum_{i=1}^{k} f(x_i)/k$

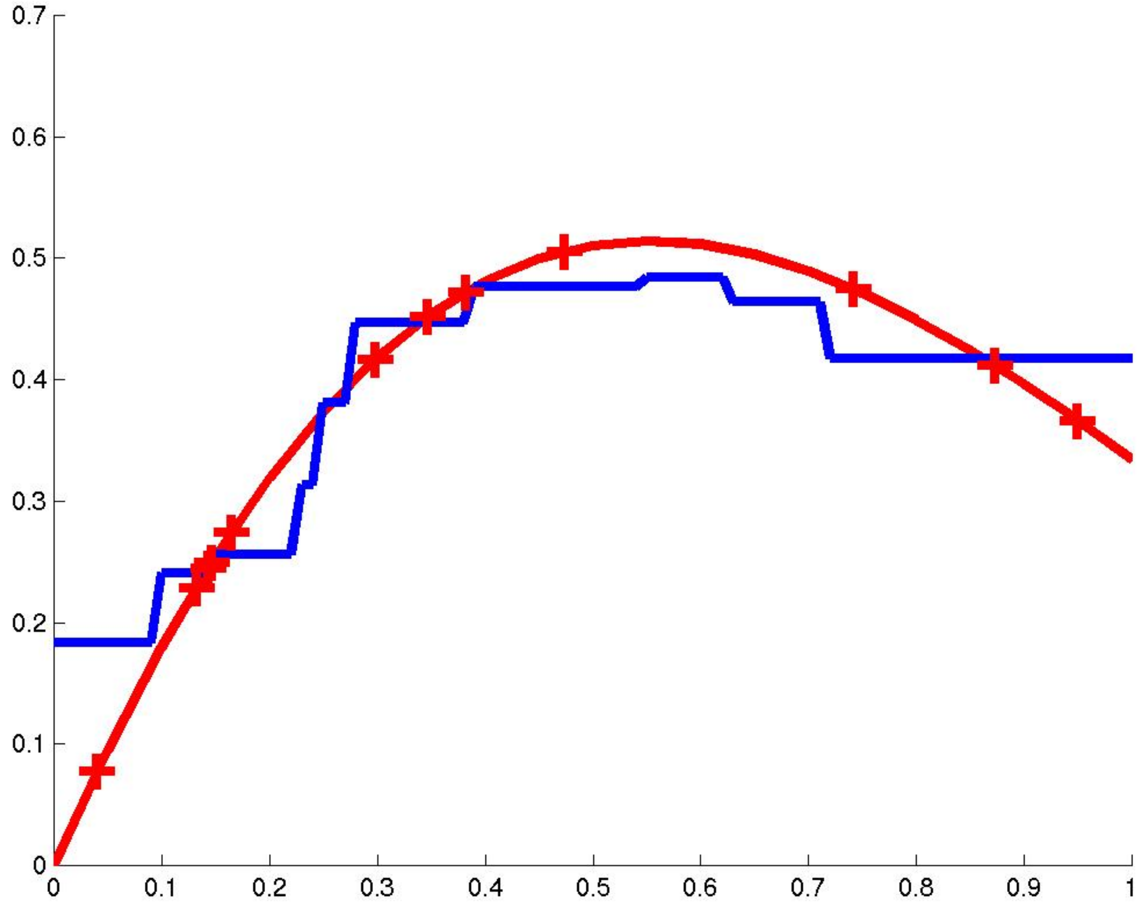
Nearest Neighbor (continuous)

1-nearest neighbor



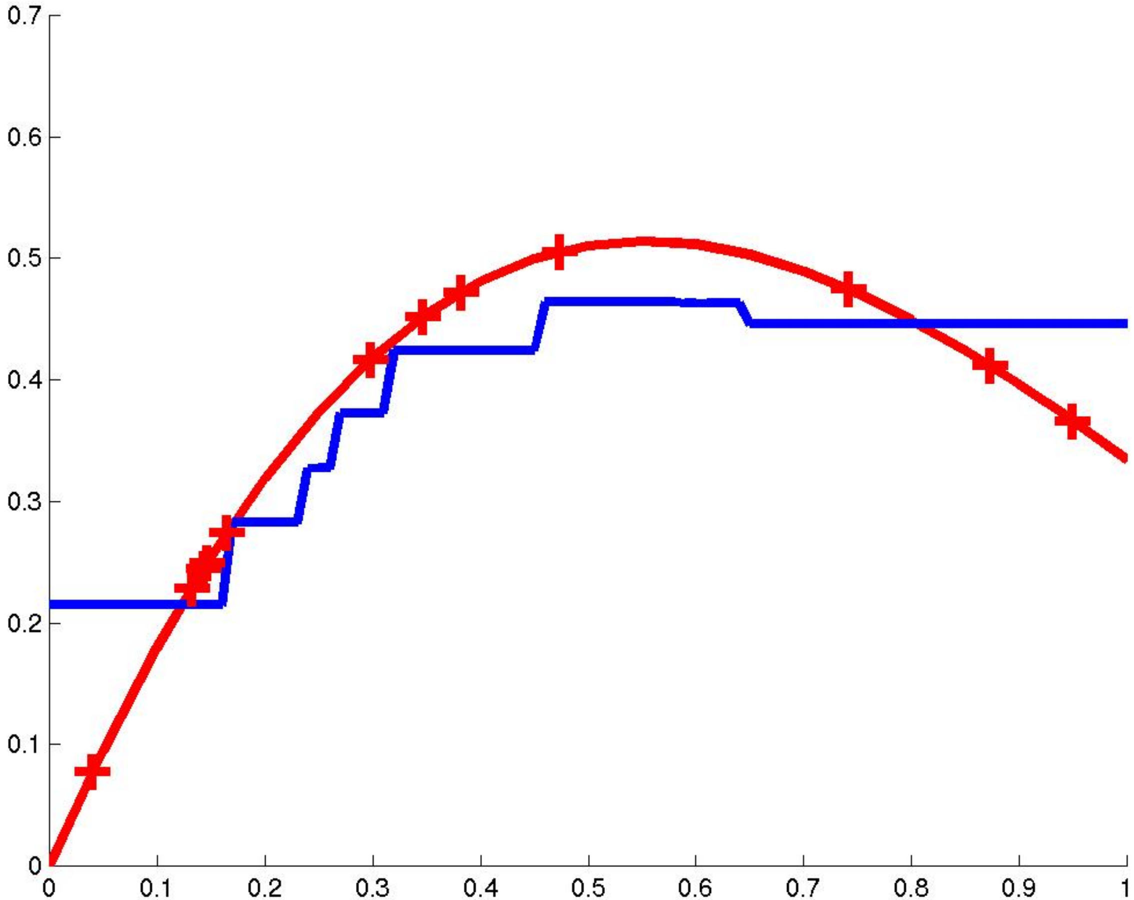
Nearest Neighbor (continuous)

3-nearest neighbor



Nearest Neighbor (continuous)

5-nearest neighbor



Nearest Neighbor Characteristics

- Compute distance between two points:
  - Usually the Euclidean distance

$$d(x_i, x_j) = \left(\sum_{s=1}^{p} (x_{is} - x_{js})^2 \right)^{\frac{1}{2}}$$

  - Minkowski distance is used in practical, which include Euclidean as a particular case

$$d(x_i, x_j) = \left(\sum_{s=1}^{p} |x_{is} - x_{js}|^q \right)^{\frac{1}{q}}$$

q=1 (Absolute or Manhattan); q=2 (Euclidean); etc.

Nearest Neighbor Characteristics

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include non-representative points
- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - income of a person may vary from $10K to $1M

      - Solution: Normalize the vectors (usually by sd)

Discussion on the *k*-NN Algorithm

- Robust to noisy data by averaging k-nearest neighbors
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant attributes.
  - To overcome it, axes stretch or elimination of the least relevant attributes.

- k-NN methods are lazy learners
  - It does not build models explicitly
  - Unlike eager learners such as decision tree induction and rule-based systems
  - Unknown records are relatively expensive to compute

Distance Weighted k-NN

- Regression means approximating a real-valued target function
- Residual is the error $\hat{f}(x) - f(x)$ in approximating the target function
- Kernel function is the function of distance that is used to determine the weight of each training example. In other words, the kernel function is the function K such that $w_i = K(d(x_i, x_q))$

Distance Weighted k-NN

Give more weight to neighbors closer to the query point

$f^{\wedge}(x_q) = \Sigma_{i=1}^{k} w_i f(x_i) / \Sigma_{i=1}^{k} w_i$

where $w_i=K(d(x_q,x_i))$

and $d(x_q,x_i)$ is the distance between x_q and x_i

Instead of only k-nearest neighbors use all training examples (Shepard's method)

Kernel Functions

- rectangular kernel $\frac{1}{2} \cdot \mathrm{I}(|d| \leq 1)$
- triangular kernel $(1 - |d|) \cdot \mathrm{I}(|d| \leq 1)$
- Epanechnikov kernel $\frac{3}{4}(1 - d^2) \cdot \mathrm{I}(|d| \leq 1)$
- quartic or biweight kernel $\frac{15}{16}(1 - d^2)^2 \cdot \mathrm{I}(|d| \leq 1)$
- triweight kernel $\frac{35}{32}(1 - d^2)^3 \cdot \mathrm{I}(|d| \leq 1)$
- cosine kernel $\frac{\pi}{4}\cos(\frac{\pi}{2}d) \cdot \mathrm{I}(|d| \leq 1)$
- Gauss kernel $\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{d^2}{2}\right)$
- inversion kernel $\frac{1}{|d|}$

Distance Weighted NN Example

$K(d(x_q,x_i)) = 1/ d(x_q,x_i)^2$