Nearest centroids, K-NN

Victor Kitov

v.v.kitov@yandex.ru

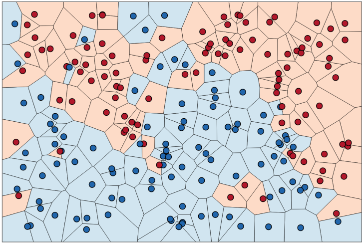


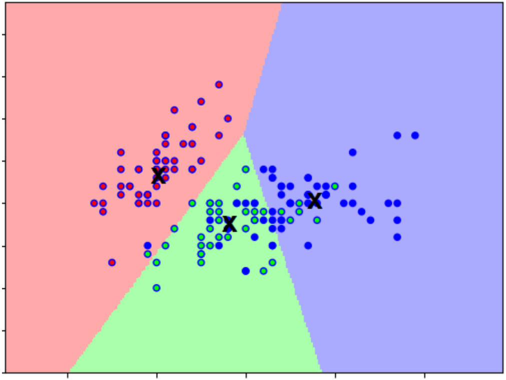
Table of Contents

Nearest centroids algorithm

- Consider training sample $(x_1, y_1), \ldots (x_N, y_N)$ with
  - N_1 representatives of 1st class
  - N_2 representatives of 2nd class
  - etc.
- **Training:**
  Calculate centroids for each class $c = 1, 2, \ldots C$:
  $$\mu_c = \frac{1}{N_C} \sum_{n=1}^{N} x_n \mathbb{I}[y_n = c]$$
- **Classification:**
  1. For object x find most close centroid:
  $$c = \arg\min_i \rho(x, \mu_i)$$
  2. Associate x the class of the most close centroid:
  $$\widehat{y}(x) = c$$

Illustration

Decision boundaries for 3-class nearest centroids



Questions

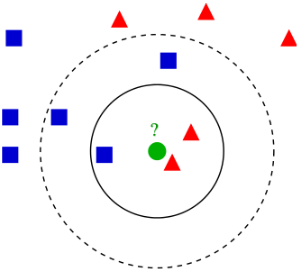
- What are discriminant functions $g_c(x)$ for nearest centroid?
- What is the complexity for:
  - training?
  - prediction?
- What would be the shape of class separating boundary?
- Can we use similar ideas for regression? Consider clustering.
- Is this method prone to the curse of dimensionality?

Table of Contents

K-nearest neighbours algorithm

Classification:

1. Find k closest objects to the predicted object x in the training set.
2. Associate x the most frequent class among its k neighbours.



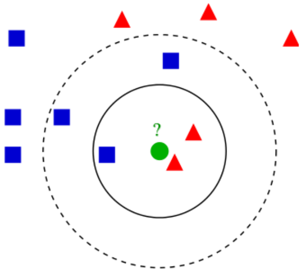
K-nearest neighbours algorithm

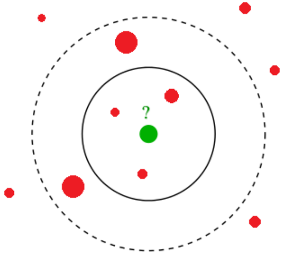
Classification:

1. Find k closest objects to the predicted object x in the training set.
2. Associate x the most frequent class among its k neighbours.

Regression:

1. Find k closest objects to the predicted object x in the training set.
2. Associate x average output of its k neighbours.



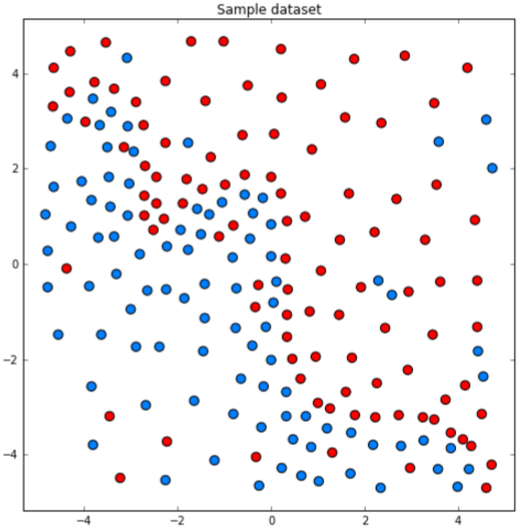


Comments

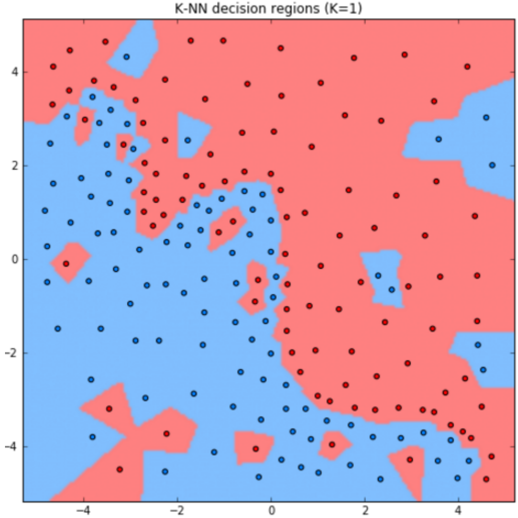
- K nearest neighbours algorithm is abbreviated as K-NN.
- $k = 1$: nearest neighbour algorithm[1]
- Base assumption of the method[2]:
    - similar objects yield similar outputs

[1] what will happen for $K = N$?
[2] what is simpler - to train K-NN model or to apply it?

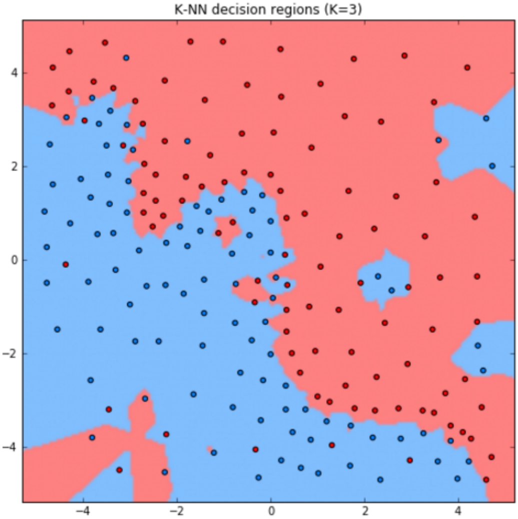
Sample dataset



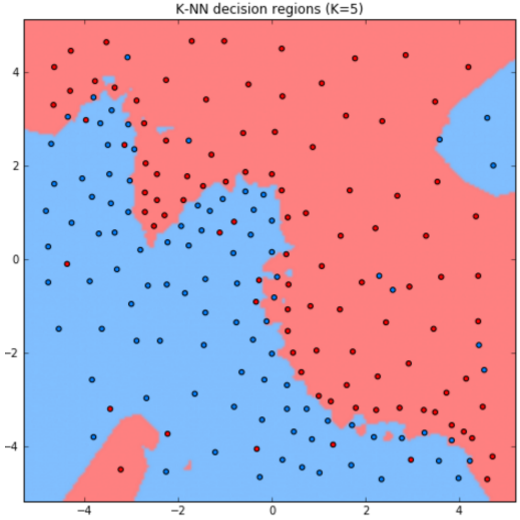
Example: K-NN classification



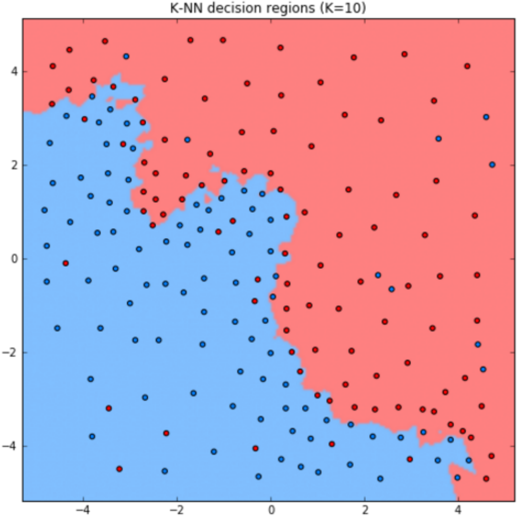
Example: K-NN classification



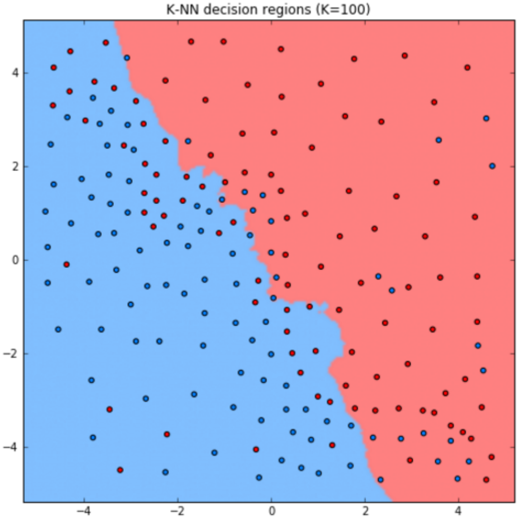
Example: K-NN classification



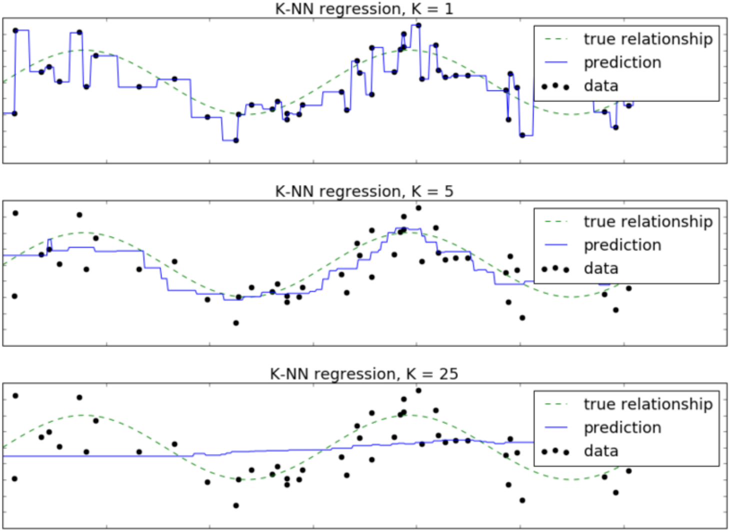
Example: K-NN classification



Example: K-NN classification



Example: K-NN regression



Dealing with similar rank

When several classes get the same rank, we can assign to class:

Dealing with similar rank

When several classes get the same rank, we can assign to class:

- with higher prior probability
- having closest representative
- having closest mean of representatives (among nearest neighbours)
- which is more compact, having nearest most distant representative

Parameters of the method

- Parameters:
  - the number of nearest neighbours K
  - distance metric $\rho(x, x')$
- Modifications:
  - forecast rejection option[3]
  - variable K[4]

---

[3] Propose a rule, under what conditions to apply rejection in a) classification b) regression

[4] Propose a method of K-NN with adaptive variable K in different parts of the feature space

Properties

- **Advantages:**
    - only similarity between objects is needed, not exact feature values.
        - so it may be applied to objects with arbitrary complex feature description
    - simple to implement
    - interpretable (case based reasoning)
    - does not need training
        - may be applied in online scenarios
        - Cross-validation may be replaced with LOO.
- **Disadvantages:**
    - slow classification with complexity $O(N)$
    - accuracy deteriorates with the increase of feature space dimensionality

Table of Contents

Normalization of features

- Feature scaling affects predictions of K-NN?

Normalization of features

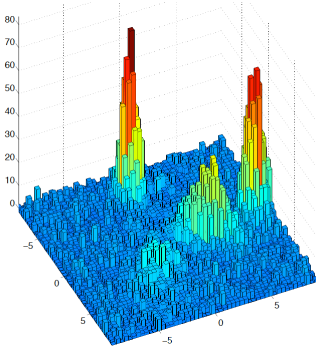
- Feature scaling affects predictions of K-NN?
    - yes, so normalize them
- Equal scaling - equal impact of features
- Non-equal scaling - non-equal impact of features
- Typical normalizations:

Name	Transformation	Properties
Standardization	$\frac{x_j - \mu_j}{\sigma_j}$	zero mean, unit variance.
Mean norm	$\frac{x_j - \mu_j}{\max(x_j) - \min(x_j)}$	zero mean, $[0, 1]$ interval.
Range scaling	$\frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}$	min=0,max=1, $[0, 1]$ interval.

- Which type of scaling is more robust to outliers?
- What type of scaling preserves the sparsity property? (many zero values)

The curse of dimensionality

- The curse of dimensionality: with growing D data distribution becomes sparse and insufficient.
- Example: histogram estimation[5]



[5] At what rate should training size grow with increase of D to compensate curse of dimensionality?

Curse of dimensionality

- Case of K-nearest neighbours:
  - assumption: objects are distributed uniformly in feature space
  - ball of radius R has volume $V(R) = CR^D$, where $C = \frac{\pi^{D/2}}{\Gamma(D/2+1)}$.
  - ratio of volumes of balls with radius $R - \varepsilon$ and R:

$$\frac{V(R-\varepsilon)}{V(R)} = \left(\frac{R-\varepsilon}{R}\right)^D \xrightarrow{D\to\infty} 0$$

  - most of volume concentrates on the border of the ball, so there lie the nearest neighbours.
  - nearest neighbours stop being close by distance
- Good news: in real tasks the true dimensionality of the data is often less than D and objects belong to the manifold with smaller dimensionality.

Table of Contents

Equal voting

- Define K nearest neighbors: $(z_1, y_1), (z_2, y_2), ...(z_K, y_K)$.

$$\rho(x, z_1) \le \rho(x, z_2) \le ... \le \rho(x, z_K)$$

- Regression:

$$\widehat{y}(x) = \frac{1}{K}\sum_{k=1}^{K} y_k$$

- Classification:

$$g_c(x) = \sum_{k=1}^{K} \mathbb{I}[y_k = c], \quad c = 1, 2, ...C.$$
$$\widehat{y}(x) = \arg\max_c g_c(x)$$

Weighted voting

- Weighted regression:

$$\widehat{y}(x)=\frac{\sum_{k=1}^{K}w(k,\ \rho(x,z_k))y_k}{\sum_{k=1}^{K}w(k,\ \rho(x,z_k))}$$

Weighted voting

- Weighted regression:

$$\widehat{y}(x)=\frac{\sum_{k=1}^{K}w(k,\ \rho(x,z_k))y_k}{\sum_{k=1}^{K}w(k,\ \rho(x,z_k))}$$

- Weighted classification:

$$g_c(x)=\sum_{k=1}^{K}w(k,\ \rho(x,z_k))\mathbb{I}[y_k=c],\quad c=1,2,...C.$$
$$\widehat{y}(x)=\arg\max_c g_c(x)$$

Commonly chosen weights

Index dependent weights:

$$w_k = \alpha^k, \quad \alpha \in (0,1)$$

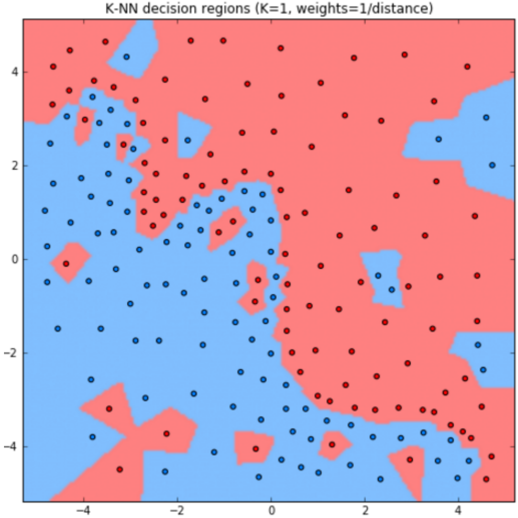
$$w_k = \frac{K+1-k}{K}$$

Distance dependent weights:

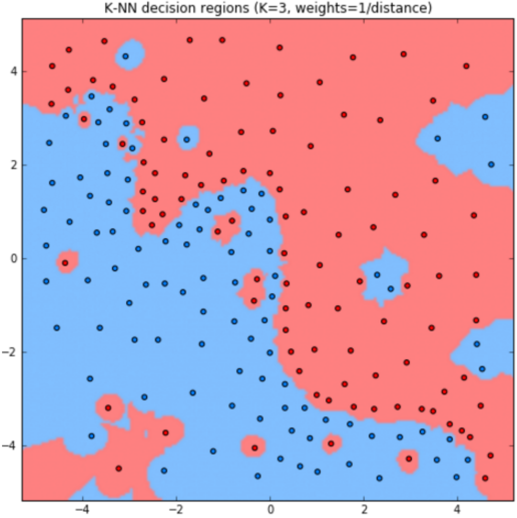
$$w_k = \begin{cases} \frac{\rho(z_K,x)-\rho(z_k,x)}{\rho(z_K,x)-\rho(z_1,x)}, & \rho(z_K,x) \neq \rho(z_1,x) \\ 1 & \rho(z_K,x) = \rho(z_1,x) \end{cases}$$

$$w_k = \frac{1}{\rho(z_k,x)}$$

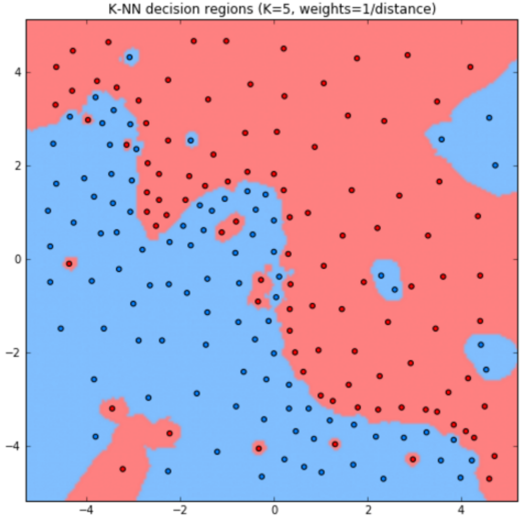
Example: K-NN classification with weights



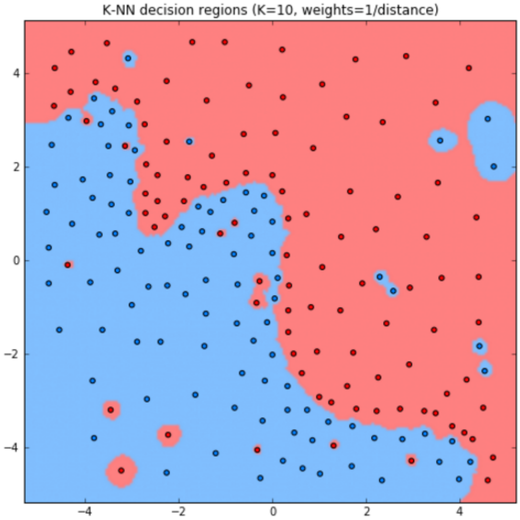
Example: K-NN classification with weights



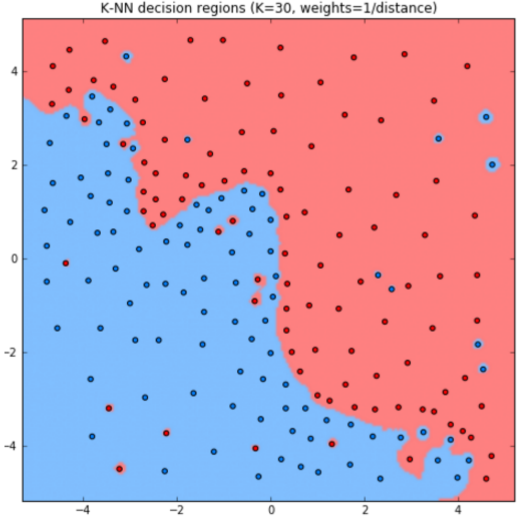
Example: K-NN classification with weights



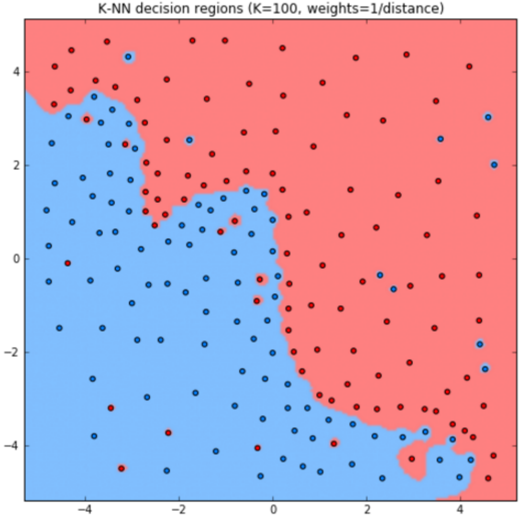
Example: K-NN classification with weights



Example: K-NN classification with weights



Example: K-NN classification with weights



Example: K-NN regression with weights

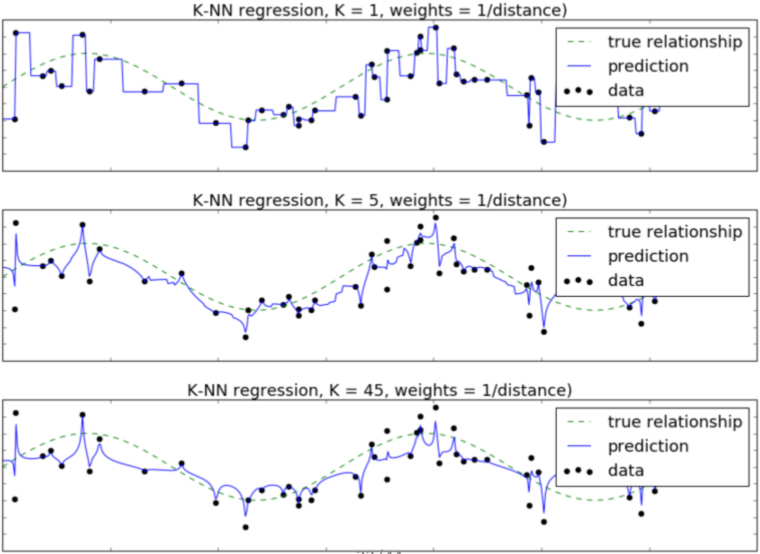


Table of Contents

Popular distance measures[6]

Название	$\boldsymbol{\rho(x,z)}$
Euclidean	$\sqrt{\sum_{i=1}^{D}(x^i - z^i)^2}$
L_p	$\sqrt[p]{\sum_{i=1}^{D}(x^i - z^i)^p}$
L_∞	$\max_{i=1,2,...D} \lvert x^i - z^i \rvert$
L_1	$\sum_{i=1}^{D} \lvert x^i - z^i \rvert$
Canberra (macro avg.)	$\frac{1}{D}\sum_{i=1}^{D}\frac{\lvert x^i - z^i\rvert}{\lvert x^i\rvert + \lvert z^i\rvert}$
Lance-Williams (micro avg.)	$\frac{\sum_{i=1}^{D}\lvert x^i - z^i\rvert}{\sum_{i=1}^{D}\lvert x^i + z^i\rvert}$

If have $S(x,z)$, then $\rho(x,z) = K(S(x,z))$ for $\downarrow K$, e.g.

$$\rho(x,z) = 1 - S(x,z) \qquad \rho(x,z) = \frac{1}{S(x,z)}$$

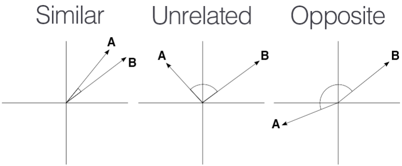
[6] Build circles with radius 1 for L_1, L_2, L_∞ distances.

Cosine measure

- Cosine measure: objects are close if the angle between them is small.

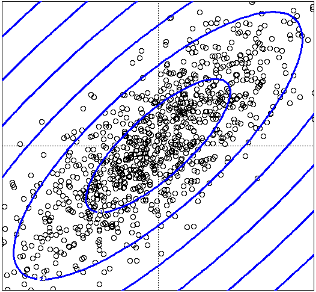
$$sim(x,z)=\frac{x^Tz}{\|x\|\,\|z\|}=\frac{\sum_{i=1}^{D}x^iz^i}{\sqrt{\sum_{i=1}^{D}(x^i)^2}\sqrt{\sum_{i=1}^{D}(z^i)^2}}$$

  - $\langle x,z\rangle=x^Tz=\|x\|\,\|z\|\,cos(\alpha)$, where α - angle between x and z.



  - measure $\in[-1,1]$, invariant to $\|x\|,\|z\|$.
    - convenient for text representations=word counts.

Dependent features: Mahalanobis distance



- Objects along $y = x$ are more similar than along $y = -x$.
- Mahalanobis distance=Euclidean distance in decorrelated feature space (for decorrelated features).

Table of Contents

Minimum squared error estimate

For training sample $(x_1, y_1), \dots (x_N, y_N)$ consider finding constant $\widehat{y} \in \mathbb{R}$:

$$L(\widehat{y}) = \sum_{i=1}^{N} (\widehat{y} - y_i)^2 \to \min_{\widehat{y} \in \mathbb{R}}$$

$$\frac{\partial L}{\partial \widehat{y}} = 2 \sum_{i=1}^{N} (\widehat{y} - y_i) = 0, \text{ so } \widehat{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$$

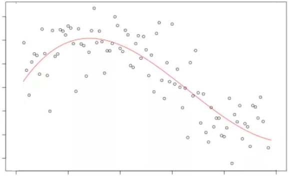
Minimum squared error estimate

For training sample $(x_1, y_1), ...(x_N, y_N)$ consider finding constant $\widehat{y} \in \mathbb{R}$:

$$L(\widehat{y}) = \sum_{i=1}^{N} (\widehat{y} - y_i)^2 \to \min_{\widehat{y} \in \mathbb{R}}$$

$$\frac{\partial L}{\partial \widehat{y}} = 2 \sum_{i=1}^{N} (\widehat{y} - y_i) = 0, \text{ so } \widehat{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$$

We need to model general curve $y(x)$:



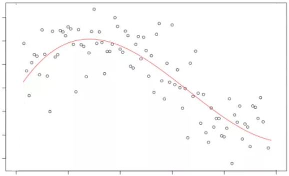
Minimum squared error estimate

For training sample $(x_1, y_1), \ldots (x_N, y_N)$ consider finding constant $\widehat{y} \in \mathbb{R}$:

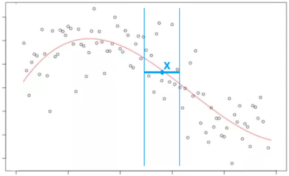
$$L(\widehat{y}) = \sum_{i=1}^{N} (\widehat{y} - y_i)^2 \to \min_{\widehat{y} \in \mathbb{R}}$$

$$\frac{\partial L}{\partial \widehat{y}} = 2 \sum_{i=1}^{N} (\widehat{y} - y_i) = 0, \text{ so } \widehat{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$$

We need to model general curve $y(x)$:



Nadaraya-Watson regression - localized averaging approach.



Nadaraya-Watson regression

- Find locally constant prediction for each x.

$$\widehat{y}(x) = \arg\min_{\widehat{y}\in\mathbb{R}} \sum_{i=1}^{N} w_i(x)(\widehat{y} - y_i)^2 = \frac{\sum_{i=1}^{N} y_i w_i(x)}{\sum_{i=1}^{N} w_i(x)}$$

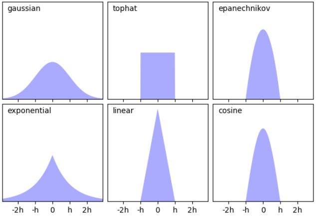
- Weights should $\downarrow$ as $\rho(x, x_i) \uparrow$ die to $\downarrow K(\cdot)$, called kernel.

$$w_i(x) = K\left(\frac{\rho(x, x_i)}{h}\right)$$

- $h(x)$ - some ≥ 0 function called bandwidth.
    - Intuition: "window width", consider $h(x) = h$, $K(u) = \mathbb{I}[u \leq 1]$.
- Equivalent names: local constant regression, kernel regression.

Функция ядра

Kernel $K(u)$	Formula
top-hat	$\mathbb{I}[\lvert u\rvert < 1]$
linear	$\max\{0, 1 - \lvert u\rvert\}$
Epanechnikov	$\max\{0, 1 - u^2\}$
exponential	$e^{-\lvert u\rvert}$
Gaussian	$e^{-\frac{1}{2}u^2}$
quartic	$(1 - u^2)^2\mathbb{I}[\lvert u\rvert < 1]$



Comments

- Weight enables non-linearity but should be recalculated for every x.
- Under general conditions $\widehat{y}(x) \xrightarrow{P} E[y|x]$
- Particular selection of $K(u)$ does not infuence accuracy as much as h.
- $K(u)$ affects continuity, smoothness and comp. efficiency.
- Can select $h(x)$ adaptively.
  - $h(x)$ lower for higher local density of points,
  - e.g. $h(x)$ - distance to K-th nearest neighbor of x.[7]

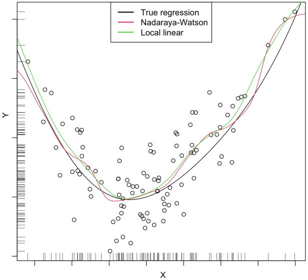
[7] What choice of $h(x)$ and $K(u)$ yield K-NN?

Local linear regression

Instead of a local constant, you can optimize locally linear regression::

$$\sum_{i=1}^{N} w_i(x)(x^T\beta - y_i)^2 \to \min_{\beta\in\mathbb{R}}; \quad \widehat{y}(x) = x^T\beta$$

It is more stable, better approximating regions of low object density, but computationally more difficult..



Summary

- Important parameters of K-NN:
  - K: controls model complexity
  - $\rho(x, x')$
- Output depends on feature scaling.
  - scaling to equal / non-equal scatter possible.
- Prone to curse of dimensionality.
- Fast training but long prediction.
  - some efficiency improvements are possible though
- Weighted account for objects possible.
- Nearest centroid has different properties.