K-nearest neighbours

Victor Kitov

Skoltech
Skolkovo Institute of Science and Technology

November-December 2015 г.

Table of Contents

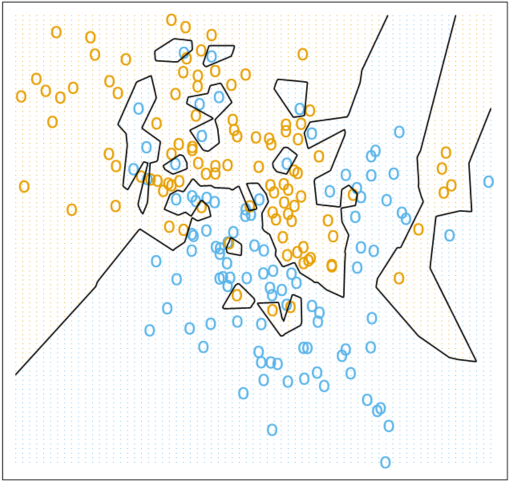
K-nearest neighbours

Classification using k nearest neighbours

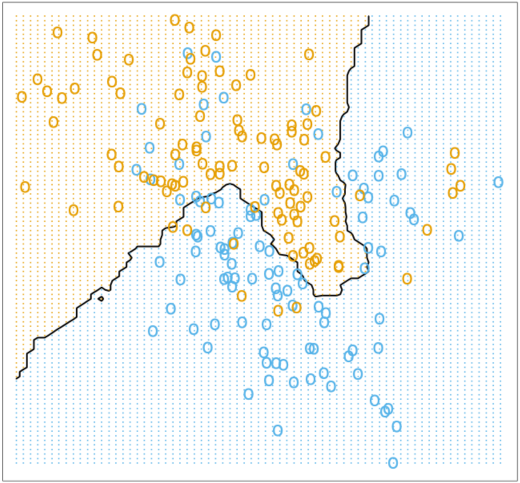
1. Find k closest objects to the predicted object x in the training set.
2. Associate x the most frequent class among its k neighbours.

- $k = 1$: nearest neighbour algorithm
- $k = N$: constant prediction with the most frequent class in the training set
- Regression case: targets of nearest neighbours are averaged
- Base assumption of the method:
    - similar objects yield similar outputs

Example: classification with 1 nearest neighbour



Example: classification with 15 nearest neighbours



Parameters of the method

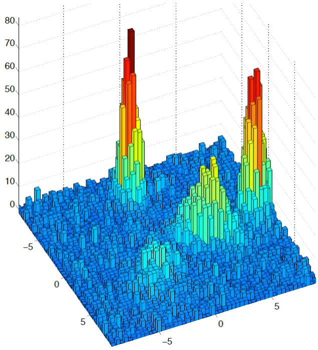
- Parameters:
  - the number of nearest neighbours k
  - distance metric $\rho(x, y)$
- Variants:
  - rejection option:
    - classification: when classes are uniformly distributed
    - regression: when variance is high
  - k is locally adapted to yield maximum accuracy.
  - random subspace modification

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
    - K-CV may be replaced with LOO.
- Disadvantages:
  - slow classification with complexity $O(N)$
  - accuracy deteriorates with the increase of feature space dimensionality

The curse of dimensionality

- The curse of dimensionality in ML is a situation when the number of training samples needs to grow exponentially when D grows linearly to guarantee certain level of accuracy.
- Example: histogram estimation



Curse of dimensionality

- Case of K-nearest neigbours:
  - 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.

Dealing with similar rank

When several classes get the same rank, we can:

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

Table of Contents

Distance metric selection

- Baseline case - Euclidean metric
- Necessary to normalize features.
    - Define μ_j, σ_j, L_j, U_j to be mean value, standard deviation, minimum and maximum value of the j-th feature.

Name	**Transformation**	**Properties of resulting feature**
Autoscaling	$x_j' = \frac{x_j - \mu_j}{\sigma_j}$	zero mean and unit variance.
Range scaling	$x_j' = \frac{x_j - L_j}{U_j - L_j}$	belongs to $[0, 1]$ interval.

Normalization of features

- Non-linear transformations incorporating features with rare large values:
  - $x_i' = \log(x_i)$
  - $x_i' = x^p$, $0 \leq p < 1$
- For $F_i(\alpha) = P(x^i \leq \alpha)$ transformation $\tilde{x}^i \to F_i(x^i)$ will give feature uniformly distributed on $[0, 1]$.

Distance metric selection

Metric	$\boldsymbol{d(x, z)}$
Евклидова	$\sqrt{\sum_{i=1}^{D}(x^i - z^i)^2}$
L_p	$\sqrt[p]{\sum_{i=1}^{D}(x^i - z^i)^p}$
L_∞	$\max_{i=1,2,\dots D} \lvert x^i - z^i \rvert$
L_1	$\sum_{i=1}^{D} \lvert x^i - z^i \rvert$
Canberra	$\frac{1}{D} \sum_{i=1}^{D} \frac{\lvert x^i - z^i \rvert}{x^i + z^i}$
Lance-Williams	$\frac{\sum_{i=1}^{D} \lvert x^i - z^i \rvert}{\sum_{i=1}^{D} x^i + z^i}$
Cosine	$\frac{\sum_{i=1}^{D} x^i z^i}{\sqrt{\sum_{i=1}^{D}(x^i)^2}\sqrt{\sum_{i=1}^{D}(z^i)^2}}$

Whitening transformation

- x is distributed with mean μ and non-degenerate covariance Σ ($\mu \in \mathbb{R}^D$, $\Sigma \in \mathbb{R}^{DxD}$)
- Whitening transformation: $z = \Sigma^{-1/2}(x - \mu)$ gives new feature vector with mean 0 and covariance $I \in \mathbb{R}^{DxD}$, where I is the identity matrix.
- Proof:

$$\mathbb{E}z = \mathbb{E}\left\{\Sigma^{-1/2}(x-\mu)\right\} = \Sigma^{-1/2}\mathbb{E}\left\{x-\mu\right\} = \mathbf{0} \in \mathbb{R}^D$$

$$\begin{aligned} cov[z] &= \mathbb{E}(z - \mathbb{E}z)(z - \mathbb{E}z)^T = \mathbb{E}zz^T \\ &= \mathbb{E}\left\{\Sigma^{-1/2}(x-\mu)(x-\mu)^T(\Sigma^{-1/2})^T\right\} = \\ &= \Sigma^{-1/2}\mathbb{E}\left\{(x-\mu)(x-\mu)^T\right\}(\Sigma^{-1/2})^T = \\ &= \Sigma^{-1/2}\Sigma\Sigma^{-1/2} = I \end{aligned}$$

Distance between normalized feature vectors

- Distance between normalized x and x' is equal to Euclidean distance between $z = \Sigma^{-1/2}(x - \mu)$ and $z' = \Sigma^{-1/2}(x' - \mu)$:

$$\begin{aligned} \rho_M(x, x') &= \rho_E(z, z') = \sqrt{(z - z')^T (z - z')} = \\ &= \sqrt{(x - x')^T \Sigma^{-1/2} \Sigma^{-1/2} (x - x')} \\ &= \sqrt{(x - x')^T \Sigma^{-1} (x - x')} \end{aligned}$$

- This is known as *Mahalonobis distance*.
- Special case when features are uncorrelated and $Var[x^i] = \sigma_i^2$:
$\rho_M(x, \tilde{x}) = \sqrt{\sum_i \frac{(x^i - \tilde{x}^i)^2}{\sigma_i^2}}$

Mahalanobis distance - illustration

(A): object in initial feature space with Mahalonobis sphere $G_\alpha = \{x : \rho_M(x, \mu) = \alpha\}$. (B): the image of objects and sphere in normalized space $(Im[G_\alpha] = \{z : \rho_E(z, 0) = \alpha\}$.

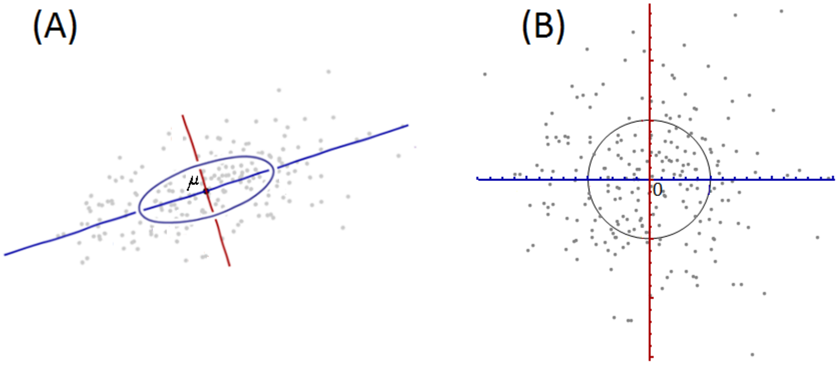


Table of Contents

Weighted voting

Let training set $x_1, x_2, \dots x_N$ be rearranged to $x_{i_1}, x_{i_2}, \dots x_{i_N}$ by increasing distance to the test pattern x:
$d(x, x_{i_1}) \leq d(x, x_{i_2}) \leq \dots \leq d(x, x_{i_N})$.
Define $z_1 = x_{i_1}$, $z_2 = x_{i_2}, \dots z_K = x_{i_K}$.
Usual K-NN algorithm can be defined, using C discriminant functions:

$$g_c(x) = \sum_{k=1}^{K} \mathbb{I}[z_k \in \omega_c], \quad c = 1, 2, \dots C.$$

Weighted K-NN algorithm uses weighted voting scheme:

$$g_c(x) = \sum_{k=1}^{K} w(k,\ d(x, z_k))\mathbb{I}[z_k \in \omega_c], \quad c = 1, 2, \dots C.$$

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:

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

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

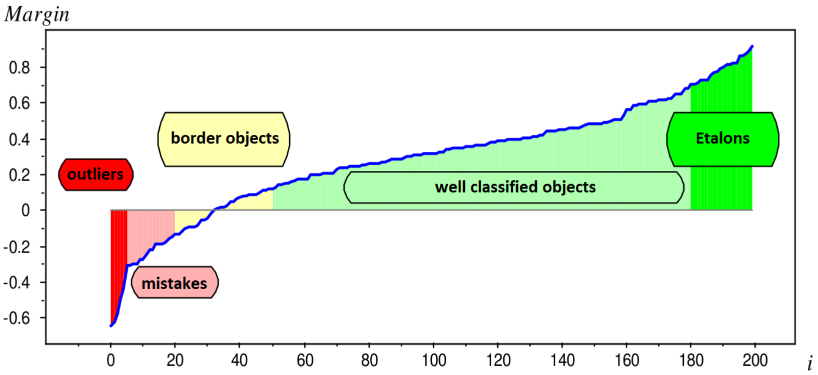
Margin definition

- Consider the training set: $(x_1, c_1), (x_2, c_2), ...(x_N, c_N)$, where c_i is the correct class for object x_i, and $\mathbf{C} = \{1, 2, ...C\}$ - is the set of all classes.
- Define the margin:

$$M(x_i, c_i) = g_{c_i}(x_i) - \max_{c \in \mathbf{C} \setminus \{\mathbf{c_i}\}} g_c(x_i)$$

  - margin is negative <=> object x_i was incorrectly classified
  - the value of margin shows how much the classifier is inclined to vote for class c_i

Categorization of objects based on margin



Good classifier should:

- minimize the number of negative margin region
- classify correctly with high margin