Nearest Neighbor classification

- example: $Y \in \{\text{small, medium, tall}\}$ X: height in m

threshold classifier

$$Y = \begin{cases} \text{small} & \text{if } X < 1.6 \text{ m} \leftarrow t_{12} \\ \text{medium} & 1.6 \le X < 2.0 \text{ m} \\ \text{tall} & X \ge 2.0 \text{ m} \end{cases} \qquad 2.0 \uparrow t_{23}$$

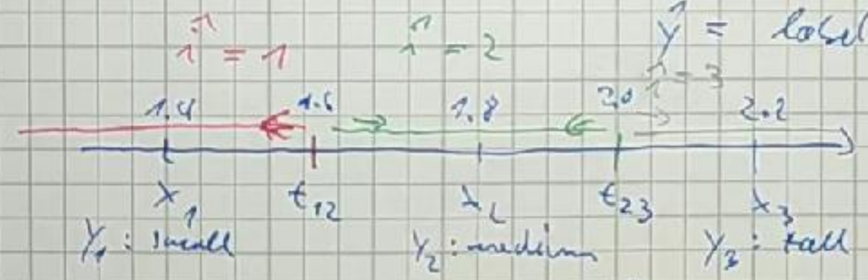
alternatively: define typical representatives for each class:

$X_{small} = 1.4 = X_1$, $X_{medium} = 1.8 \text{ m} = X_2$, $X_{tall} = 2.2 \text{ m} = X_3$

$\hat{Y}$ = label of the closest representative for a given X

two stages: $\hat{i} = \arg\min_{i \in \text{representatives}} |X - X_i|$ ← nearest neighbor

$\hat{Y}$ = label of representative $X_{\hat{i}}$



here: nearest neighbor and threshold rules are equivalent

- nearest neighbors work in any dimension, when a suitable distance function is available: $d(X, X')$ ($d(X,X) = 0$, $d(X,X') > 0$ if $X \neq X'$, $d(X,X') + d(X',X'') \ge d(X,X'')$ triangle inequal.)

⇒ given a set of representatives $TS = \{(X_i, Y_i)\}_{i=1}^{N}$ Y_i ≙ true label of instance i

nearest neighbor rule	$\hat{i} = \arg\min_i d(X, X_i)$ ↑ test feature	return $\hat{Y} = Y_{\hat{i}}$

- advantage: standard distance functions often work reasonably well, most popular:

  Euclidean distance $d(x, x') = \|x - x'\|_2 = \sqrt{\sum_{j=1}^{D} (x_j - x'_j)^2}$

  or weighted Euclidean distance

$$d(x, x') = \sqrt{\sum_{j=1}^{D} (x_j - x'_j)^2 \cdot \underbrace{w_j}_{\text{weight of feature } j}}$$

  intuitive meaning: measure features in units such that the magnitudes of different features are comparable

  example: body-mass-index: if height is measured in cm or km instead of m, it doesn't work

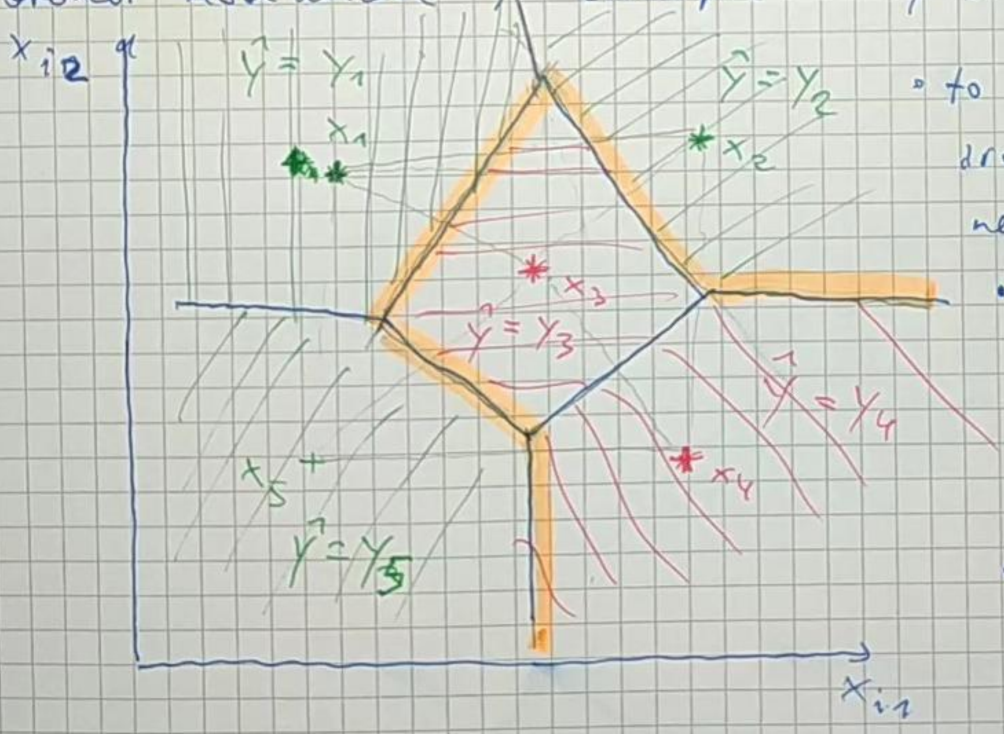
  standard weight: is inverse variance $w_j = \frac{1}{\sigma_j^2}$ ← variance of feature j

- but if standard distance do ~~~~ not work, defining a good distance $d(x, x')$ is hard (as hard as defining a good $z = g(x)$) ⇒ <u>metric learning</u>

Nearest Neighbor Classifier Recap:

- Training is trivial: just store ("memorize") the $TS = \{(X_i, Y_i)\}_{i=1}^{N}$
- Testing / inference:
  - need a suitable distance function for features $d(X, X')$
  - for test features X:
    - find index of the nearest (= most similar) training instance: $\hat{i} = \arg\min_i d(X, X_i)$
    - return the label of instance $\hat{i}$: $\hat{Y} = Y_{\hat{i}}$
- the training set partitions the feature space into regions, where each i is the winner of the argmin distance: Voronoi regions

=> Voronoi tesselation, example $Y \in \{0, 1\}$ $X_i \in \mathbb{R}^D$



- to construct Voronoi regions: draw bisector between neighbouring points
- merge all regions of the same label into one big ~~region~~ region per label "decision regions"
- the boundary between decision regions: "decision boundary"

· How good can the Nearest Neighbor rule perform ?

~~· three ways to analyze the error~~

· the training error of a NN cl. is meaningless: whenever query point X equals a training point X_i : $X = X_i$ for some i, then $\hat{Y} = Y_i$
⇒ the label of the response is always correct ⇒ training error $= 0$

· this doesn't imply that the test error / generalization error is also small

· three ways to analyse the generalization error
- compute an exact analytical formula - usually only possible for toy problems
- compute an analytic asymptotic formula, i.e. for infinitely large TS, $N \to \infty$
- estimate it empirically, either with separate test data set or cross validation (≈ fake test set)

Analytic error analysis for a toy problem

priors: $p(Y=0) = p(Y=1) = \frac{1}{2}$ $\quad Y \in \{0,1\}$

likelihoods $p(X \mid Y=0) = 2 - 2X$, $\quad X \in [0,1]$

$p(X \mid Y=1) = 2X$

evidence: $p(X) = 1$

posteriors: $p(Y=0 \mid X) = 1 - X$

$p(Y=1 \mid X) = X$

Reduce the NN cl. for $N=2$ to threshold classification

one training instance with either label

e.g. $TS = \{(X_1, Y_1=0), (X_2, Y_2=1)\}$

cases: - if $X_1 < X_2$: ⇒ type A classifier

- if $X_1 > X_2$: ⇒ type B anti-classifier

with threshold $t = \frac{X_1 + X_2}{2}$

⇒ calculate prob. of different TS ⇒ expected NN error

· How good can the Nearest Neighbor rule perform?

~~· three ways to analyze the error~~

· the training error of a NN cl. is meaningless: whenever query point x equals a training point x_i : $x = x_i$ for some i, then $\hat{Y} = Y_i$
$\Rightarrow$ the label of the response is always correct $\hat{=}$ training error $= 0$

· this doesn't imply that the test error / generalization error is also small

· three ways to analyse the generalization error
- compute an exact analytical formula - usually only possible for toy problems
- compute an analytic asymptotic formula, i.e. for infinitely large TS, $N \to \infty$
- estimate it empirically, either with separate test data set or cross validation ($\hat{=}$ fake test set)

Analytic error analysis for a toy problem

priors: $p(Y=0) = p(Y=1) = 1/2$ $\quad Y \in \{0,1\}$

likelihoods $p(X \mid Y=0) = 2 - 2x$, $x \in [0,1]$

$p(X \mid Y=1) = 2x$

evidence: $p(x) = 1$

posteriors: $p(Y=0 \mid X) = 1 - x$

$p(Y=1 \mid X) = x$

Reduce the NN cl. for $N=2$ to threshold classification

one training instance with either label

e.g. $TS = \{ (X_1, Y_1 = 0), (X_2, Y_2 = 1) \}$

cases: - if $x_1 < x_2$: $\Rightarrow$ type A classifier
- if $x_1 > x_2$: $\Rightarrow$ type B anti-classifier

with threshold $t = \frac{x_1 + x_2}{2}$

$\Rightarrow$ calculate prob. of different TS $\Rightarrow$ expected NN error

· probability of TS: pair $(x_1, y_1 = 0)$ has prob $p(x_1 \mid Y = 0)$

pair $(x_2, y_2 = 1)$ has prob. $p(x_2 \mid Y = 1)$

$$\mathbb{E}_{\substack{x_1, x_2 \\ x_1 < x_2}} \left[p\left(\text{error} \mid \text{rule} = A,\ t = \frac{x_1 + x_2}{2}\right)\right]$$

$$= \int_0^1 \underbrace{p(x_1 \mid Y=0)}_{=2-2x_1} \cdot \int_{x_1}^1 \underbrace{p(x_2 \mid Y=1)}_{=2x_2} \underbrace{p\left(\text{error} \mid \text{rule} = A,\ t = \frac{x_1+x_2}{2}\right)}_{=\left(\frac{x_1+x_2}{2} - \frac{1}{2}\right)^2 + 1/4} \, dx_2 \, dx_1$$

$= \frac{83}{360}$ (as per Wolfram Cloud)

$$\mathbb{E}_{\substack{x_1, x_2 \\ x_1 > x_2}} \left[p\left(\text{error} \mid \text{rule} = B,\ t = \frac{x_1 + x_2}{2}\right)\right]$$

$$= \int_0^1 p(x_1 \mid Y=0) \int_0^{x_1} p(x_2 \mid Y=1) \, p\left(\text{error} \mid \text{rule} = B,\ t = \frac{x_1+x_2}{2}\right) dx_2 \, dx_1$$

$= \frac{43}{360}$ → expected error of the NN-rule with $N = 2$:

error: $\frac{83}{360} + \frac{43}{360} = \frac{7}{20} = 35\,\%$

if you repeat the experiment "draw a TS of $N=2$, test the resulting NN rule" infinitely often, the average error is 35% (compare Bayes: 25%, guessing: 50%)

Asymptotic analytic error analysis: $N \to \infty$ (infinitely many training instances)

Definition: a classification algorithm is called consistent, if it converges to the Bayes classifier when $N \to \infty$

Result: NN rule is not consistent, but only a factor 2 off

i.e. $p_{N \to \infty}(\text{error NN}) \leq 2 \cdot \underset{\text{Bayes}}{p^*}(\text{error Bayes})$) .

[full derivation: Duda, Hart, Stork, chapter 4.5]

$p(\text{error} \mid \underset{\substack{\uparrow \\ \text{query} \\ \text{feature}}}{X}, \underset{\substack{\uparrow \\ \text{nearest} \\ \text{training instance}}}{X'})$ $\qquad X' = \arg\min_{X_i} d(X, X_i)$

define probability $p(X' \mid X)$ that X' is the nearest training point of X

(1) $p(\text{error} \mid X) = \int p(\text{error} \mid X, X')\, p(X' \mid X)\, dX'$

"marginalization over X'" $\hat{=}$ eliminate the unknown variable X'

[observation: when $N \to \infty$, a training example is at every X

$\Rightarrow p(X' \mid X) \xrightarrow[N \to \infty]{} \delta(X - X')$ if $p(X)$ is smooth]

(2) $p(\text{error} \mid X, X') = 1 - p(\text{correct} \mid X, X') = 1 - \sum_{k=1}^{C} p(\underset{\substack{\uparrow \\ \text{truth}}}{Y=k}, \underset{\substack{\uparrow \\ \text{prediction}}}{Y'=k} \mid X, X')$

assumption: query point is selected independently of the TS, but from same distribution

$\Rightarrow p(Y=k, Y'=k \mid X, X') = p(Y=k \mid X) \cdot p(Y'=k \mid X')$

$$p(\text{error} \mid x, x') = 1 - \sum_{k=1}^{C} p(Y=k \mid x)\, p(Y'=k \mid x')$$

(3) $$p_{N\to\infty}(\text{error NN} \mid x) = \lim_{N\to\infty} p(\text{error NN} \mid x)$$

$$= \int \left(1 - \sum_{k=1}^{C} p(Y=k \mid x) \cdot p(Y'=k \mid x')\right) \underbrace{\left(\lim_{N\to\infty} p(x' \mid x)\right)}_{=\delta(x-x')} dx'$$

$$\boxed{p_{N\to\infty}(\text{error NN} \mid x) = \underbrace{1 - \sum_{k=1}^{C} p(Y=k \mid x)^2}}$$

$\Rightarrow$ Gini impurity of the label distribution at x

extreme cases: - best case: only one label can occur for feature x

$$p(Y = Y^* \mid x) = 1, \quad p(Y \neq Y^* \mid x) = 0$$

$$\Rightarrow p_{N\to\infty}(\text{error NN} \mid x) = 0 \quad \Rightarrow \text{perfect classification}$$

- worst case: all labels are equally ~~likely at~~ likely at feature x:

$$p(Y=k \mid x) = \frac{1}{C} \quad \text{for all } k$$

$$\Rightarrow p_{N\to\infty}(\text{error NN} \mid x) = 1 - C \cdot \frac{1}{C^2} = \frac{C-1}{C} \geq \frac{1}{2} \quad \text{pure guessing}$$

typical case: for a given feature, the label probabilities are all different $p(Y=k \mid x)$ arbitrary (but positive!) $\sum_k p(Y=k \mid x) = 1$

⇒ predicted labels are random in the sense that we do not know which training label we happened to get at point X

⇒ lengthy calculation to get the average error over all x

$$P_{N\to\infty}(\text{error NN}) = \mathbb{E}_X\left[P_{N\to\infty}(\text{error NN}(X))\right]$$

<u>Result</u>:

$$\boxed{\underset{\substack{\uparrow\\ \text{Bayes error}}}{p^*(\text{error})} \le P_{N\to\infty}(\text{error NN}) \le p^*\left(2 - \frac{C}{C-1}p^*\right)}$$

special cases: - $p^* = 0$ ⇒ $p^*\left(2 - \frac{C}{C-1}p^*\right) = 0$ ⇒ $P_{N\to\infty}(\text{error NN}) = 0$

best case

- worst case: $p^* = \frac{C-1}{C}$ (pure guessing, non-informative features, equal class priors)

$$\Rightarrow p^*\left(2 - \frac{C}{C-1}p^*\right) = \frac{C-1}{C}\left(2 - \underbrace{\frac{C}{C-1}\cdot\frac{C-1}{C}}_{=1}\right) = \frac{C-1}{C} \text{ pure guessing}$$

$$P_{N\to\infty}(\text{error NN}) = \frac{C-1}{C} \text{ pure guessing}$$

- typical case: Bayes classifier works well $p^* = \varepsilon \ll 1$

always true $\frac{C}{C-1} \le 2$ ⇒ $P_{N\to\infty}(\text{error NN}) \le \varepsilon\left(2 - \underbrace{\frac{C}{C-1}\varepsilon}_{\ll 1}\right) \approx 2\varepsilon$

$$\boxed{P_{N\to\infty}(\text{error NN}) \le 2\varepsilon}$$ $\varepsilon \ll 1$ error of a good Bayes classif.