Regression (Theory!)

Definition 3.1 (Regression) Let $\boldsymbol{X}$ be a D-dimensional input and $\boldsymbol{Y}$ be a K-dimensional quantitative output with joint density $\rho(\boldsymbol{x}, \boldsymbol{y})$. In **regression** we seek, given a training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, for a function $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}^K$ that minimizes the error between $\boldsymbol{Y}$ and its prediction $\hat{\boldsymbol{Y}} = \boldsymbol{f}(\boldsymbol{X})$.

$X: \Omega \to \mathbb{R}^D$
$Y: \Omega \to \mathbb{R}$
$\rho(x,y) = ?$
$\Rightarrow \rho(y|x) = ?$

Idea: $Y \approx \hat{Y} := f(X)$
$f(X) = ?$

Definition 3.4 (Expected (squared) predition error in regression) Let $\boldsymbol{X}, \boldsymbol{Y}$ be input and quantitative output with joint density $\rho(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}^K$ a function to predict $\boldsymbol{Y}$ from $\boldsymbol{X}$. For the squared loss, the **expected (squared) prediction error** or **risk of $\boldsymbol{f}$**, $\mathrm{EPE}(\boldsymbol{f})$, is defined as

$$\begin{aligned} \mathrm{EPE}(\boldsymbol{f}) &= \mathrm{E}\left(L_2(\boldsymbol{Y}, \boldsymbol{f}(\boldsymbol{X}))\right) \\ &= \int_{\mathbb{R}^K} \int_{\mathbb{R}^D} L_2(\boldsymbol{y}, \boldsymbol{f}(\boldsymbol{x})) \rho(\boldsymbol{x}, \boldsymbol{y}) \, d\boldsymbol{x} d\boldsymbol{y} \,. \end{aligned}$$

"prediction error of f":
$EPE(f)$

Warning! Did not use $\{(x_i, y_i)\}_{i=1}^N$!

Theorem 3.1 Let $\boldsymbol{X}$, $\boldsymbol{Y}$ be input and quantitative output with joint density $\rho(\boldsymbol{x}, \boldsymbol{y})$. The function $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}^K$ that minimizes (under appropriate conditions) the expected (squared) prediction error $\mathrm{EPE}(\boldsymbol{f})$ is given by

$$\boldsymbol{f}(\boldsymbol{x}) = \underset{\boldsymbol{g}(\boldsymbol{x})}{\arg\min}\, \mathrm{EPE}(\boldsymbol{g}) = \mathrm{E}\left(\boldsymbol{Y} | \boldsymbol{X} = \boldsymbol{x}\right)$$

$E(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x})$ is often called **regressor** / **regression function**.

Which is "best" f?

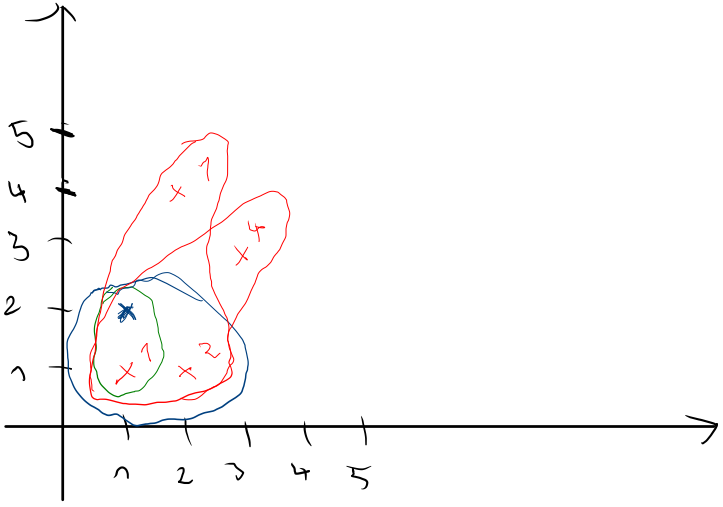
$f(x) = E(Y|X=x)$

2 kNN regression

We are given the training data

$$\mathcal{T} = \left\{ \left((1,1)^\top, 1\right), \left((2,1)^\top, 2\right), \left((3,3)^\top, 4\right), \left((2,4)^\top, 1\right) \right\}.$$

Use kNN regression with $k = 1,2,3$ to do a prediction for $\boldsymbol{x} = (1,2)$.



$k = 1 \Rightarrow \hat{f}\left((1,2)^\top\right) = 1$

$k = 2 \Rightarrow \hat{f}\left((1,2)^\top\right) = 1{,}5$

$k = 3 \Rightarrow \hat{f}\left((1,2)^\top\right) = 1\frac{1}{3}$

$\hat{f}\left((1,2)^\top\right) = 2\frac{1}{3}$

Classification

Definition 3.2 (Classification) Let $\boldsymbol{X}$ be a D-dimensional input and G be a one-dimensional qualitative output with density $\rho(\boldsymbol{x}, g)$ and range R_G. In **classification** we seek, given a training set $\{(\boldsymbol{x}_i, g_i)\}_{i=1}^N$, for a function $f : \mathbb{R}^D \to R_G$ that minimizes the error between G and its prediction $\hat{G} = f(\boldsymbol{X})$.

Definition 3.5 (Expected (0-1) predition error in classification) Let $\boldsymbol{X}$, G be input and a one-dimensional output with mixed joint density $\rho(\boldsymbol{x}, g)$ and $f : \mathbb{R}^D \to R_G$ a function to predict G from $\boldsymbol{X}$. For the 0-1 loss, the **expected prediction error of** f, $\mathrm{EPE}(f)$, is defined as

$$\begin{aligned}\mathrm{EPE}(f) = \mathrm{E}\left(L_{0-1}(G, f(\boldsymbol{X}))\right) &= \sum_{g \in R_G} \int_{\mathbb{R}^D} L_{0-1}(g, f(\boldsymbol{x}))\rho(\boldsymbol{x}|g)d\boldsymbol{x}\, p_G(g) \\ &= \int_{\mathbb{R}^D} \sum_{g \in R_G} \left(L_{0-1}(g, f(\boldsymbol{x}))p(g|\boldsymbol{x})\right) \rho_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}\,.\end{aligned}$$

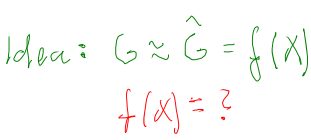
Theorem 3.2 Let $\boldsymbol{X}$ be a continuous D-dimensional input and G be the qualitative discrete output ($R_G = \{1, \ldots, r\}$) with mixed joint density $\rho(\boldsymbol{x}, g)$. The function $f : \mathbb{R}^D \to R_G$ that minimizes (under appropriate conditions) the expected prediction error $\mathrm{EPE}(f)$ with respect to the $0-1$ loss is given by

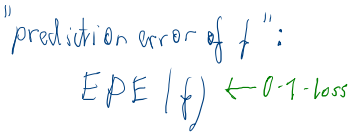
$$f(\boldsymbol{x}) = \arg\min_{g \in R_G} [1 - p(g|\boldsymbol{x})]\,,$$

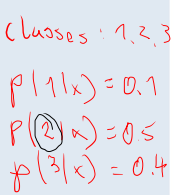
which can be simplified to

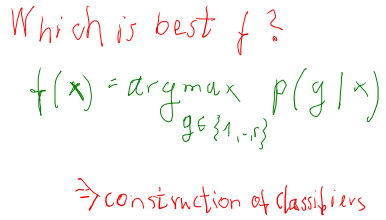
$$f(\boldsymbol{x}) = \mathfrak{g} \quad \text{if} \quad p(\mathfrak{g}|\boldsymbol{x}) = \max_{g \in R_G} p(g|\boldsymbol{x})$$

This minimizer is the **Bayes classifier**.









3 kNN classification

Previously, we studied the the well-known Iris data set and made it available in a Jupyter notebook. Moreover, we had an example Jupyter notebook in which we performed classification by the kNN classification method.

Combine both notebooks such that you carry out kNN classification for the Iris data set. For now, you just load all training data, fit the kNN model and predict the labels for a subset of the training data. Finally, you compare the results of the kNN model to the actual labels associated to the inputs.