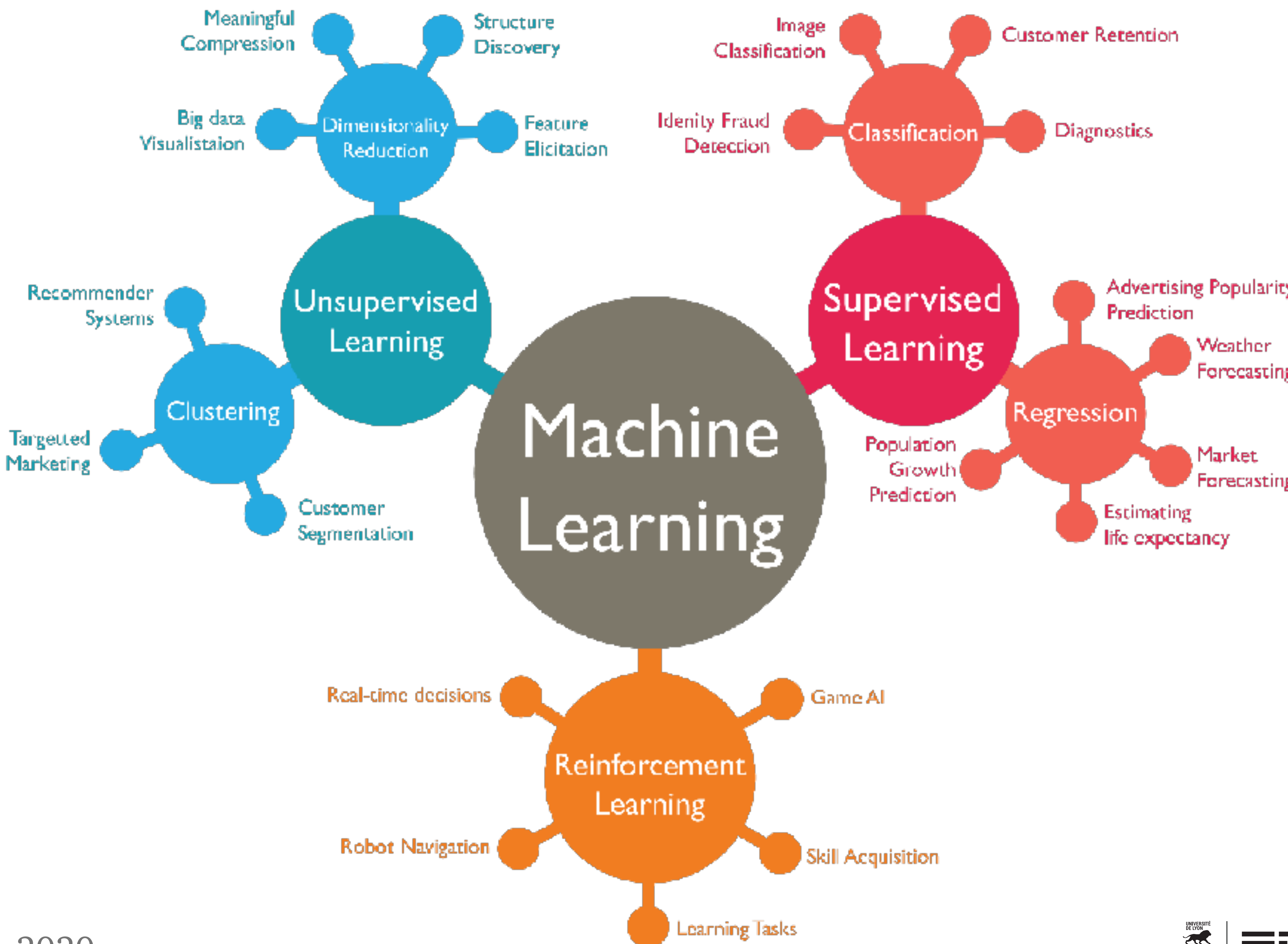
Lecture 4: Supervised Learning

Yohann De Castro & Aurélien Garivier

UNIVERSITÉ DE LYON

ENS DE LYON

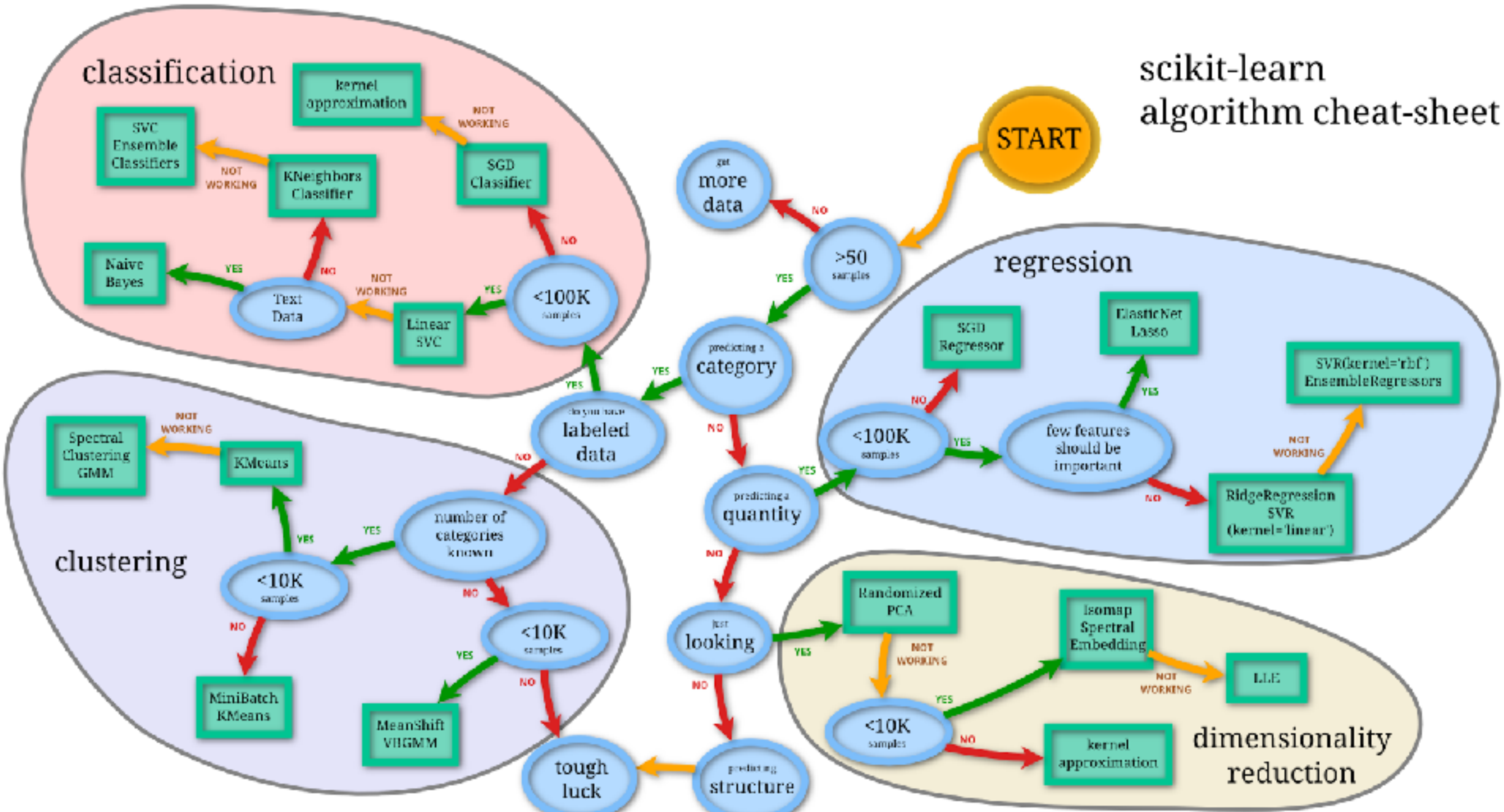
The ML family



- Supervised Learning:
  - Goal: Learn a function f predicting a variable Y from an individual $\mathbf{X}$.
  - Data: Learning set $(\mathbf{X}_i, Y_i)$

- Supervised Learning:
  - Goal: Learn a function f predicting a variable Y from an individual $\mathbf{X}$.
  - Data: Learning set $(\mathbf{X}_i, Y_i)$
- Unsupervised Learning:
  - Goal: Discover a structure within a set of individuals $(\mathbf{X}_i)$.
  - Data: Learning set $(\mathbf{X}_i)$

A Python Library: Scikit-Learn



UNIVERSITÉ DE LYON
ENS DE LYON

Supervised Learning

Decision Theory and Bias-Variance Decomposition, the quest for optimality

Supervised Learning Framework

- Input measurement $\mathbf{X} \in \mathcal{X}$
- Output measurement $Y \in \mathcal{Y}$.
- $(\mathbf{X}, Y) \sim \mathbf{P}$ with $\mathbf{P}$ unknown.
- Training data : $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)

Supervised Learning Framework

- Input measurement $\mathbf{X} \in \mathcal{X}$
- Output measurement $Y \in \mathcal{Y}$.
- $(\mathbf{X}, Y) \sim \mathbf{P}$ with $\mathbf{P}$ unknown.
- Training data : $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)

- Often
  - $\mathbf{X} \in \mathbb{R}^d$ and $Y \in \{-1, 1\}$ (classification)
  - or $\mathbf{X} \in \mathbb{R}^d$ and $Y \in \mathbb{R}$ (regression).
- A classifier is a function in $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y} \text{ meas.}\}$

Supervised Learning Framework

- Input measurement $\mathbf{X} \in \mathcal{X}$
- Output measurement $Y \in \mathcal{Y}$.
- $(\mathbf{X}, Y) \sim \mathbf{P}$ with $\mathbf{P}$ unknown.
- Training data : $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)

- Often
  - $\mathbf{X} \in \mathbb{R}^d$ and $Y \in \{-1, 1\}$ (classification)
  - or $\mathbf{X} \in \mathbb{R}^d$ and $Y \in \mathbb{R}$ (regression).
- A classifier is a function in $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y} \text{ meas.}\}$

Goal

- Construct a good classifier $\widehat{f}$ from the training data.

- Need to specify the meaning of good.
- Classification and regression are almost the **same** problem!

Loss function for a generic predictor

- Loss function : $\ell(Y, f(\mathbf{X}))$ measures the goodness of the prediction of Y by $f(\mathbf{X})$
- Examples:
  - Prediction loss: $\ell(Y, f(\mathbf{X})) = \mathbf{1}_{Y \neq f(\mathbf{X})}$
  - Quadratic loss: $\ell(Y, \mathbf{X}) = |Y - f(\mathbf{X})|^2$

Loss function for a generic predictor

- Loss function : $\ell(Y, f(\mathbf{X}))$ measures the goodness of the prediction of Y by $f(\mathbf{X})$
- Examples:
  - Prediction loss: $\ell(Y, f(\mathbf{X})) = \mathbf{1}_{Y \neq f(\mathbf{X})}$
  - Quadratic loss: $\ell(Y, \mathbf{X}) = |Y - f(\mathbf{X})|^2$

Risk function

- Risk measured as the average loss for a new couple:

$$\mathcal{R}(f) = \mathbb{E}_{(X,Y)\sim \mathbf{P}}\left[\ell(Y, f(\mathbf{X}))\right]$$

- Examples:
  - Prediction loss: $\mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{P}\left\{Y \neq f(\mathbf{X})\right\}$
  - Quadratic loss: $\mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}\left[|Y - f(\mathbf{X})|^2\right]$

- **Beware:** As $\widehat{f}$ depends on $\mathcal{D}_n$, $\mathcal{R}(\widehat{f})$ is a random variable!

Supervised Learning Framework

- Input measurement $\mathbf{X} \in \mathcal{X}$
- Output measurement $Y \in \mathcal{Y}$.
- $(\mathbf{X}, Y) \sim \mathbf{P}$ with $\mathbf{P}$ unknown.
- Training data : $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)

Supervised Learning Framework

- Input measurement $\mathbf{X} \in \mathcal{X}$
- Output measurement $Y \in \mathcal{Y}$.
- $(\mathbf{X}, Y) \sim \mathbf{P}$ with $\mathbf{P}$ unknown.
- Training data : $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)

Goal

- Learn a rule to construct a classifier $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. the risk $\mathcal{R}(\widehat{f})$ is small on average or with high probability with respect to $\mathcal{D}_n$.

- The best solution f^* (which is independent of $\mathcal{D}_n$) is

$$f^* = \arg\min_{f \in \mathcal{F}} R(f) = \arg\min_{f \in \mathcal{F}} \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{x}))\right]\right]$$

- The best solution f^* (which is independent of $\mathcal{D}_n$) is

$$f^* = \arg\min_{f \in \mathcal{F}} R(f) = \arg\min_{f \in \mathcal{F}} \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{x}))\right]\right]$$

Bayes Classifier (explicit solution)

- In binary classification with $0-1$ loss:

$$f^*(\mathbf{X}) = \begin{cases} +1 & \text{if} \quad \mathbb{P}\left\{Y = +1|\mathbf{X}\right\} \geq \mathbb{P}\left\{Y = -1|\mathbf{X}\right\} \\ & \quad \Leftrightarrow \mathbb{P}\left\{Y = +1|\mathbf{X}\right\} \geq 1/2 \\ -1 & \text{otherwise} \end{cases}$$

- In regression with the quadratic loss

$$f^*(\mathbf{X}) = \mathbb{E}\left[Y|\mathbf{X}\right]$$

Issue: Explicit solution requires to know $\mathbb{E}\left[Y|\mathbf{X}\right]$ for all values of $\mathbf{X}$!

Machine Learning

- Learn a rule to construct a classifier $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. the risk $\mathcal{R}(\widehat{f})$ is small on average or with high probability with respect to $\mathcal{D}_n$.

- In practice, the rule should be an algorithm!

Empirical Risk Minimizer

Machine Learning

- Learn a rule to construct a classifier $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. the risk $\mathcal{R}(\widehat{f})$ is small on average or with high probability with respect to $\mathcal{D}_n$.

- In practice, the rule should be an algorithm!

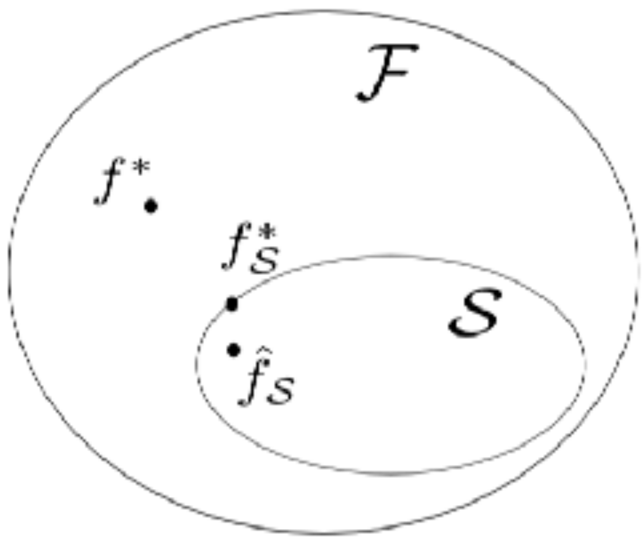
Canonical example: Empirical Risk Minimizer

- One restricts f to a subset of functions $\mathcal{S} = \{f_\theta, \theta \in \Theta\}$
- One replaces the minimization of the average loss by the minimization of the empirical loss

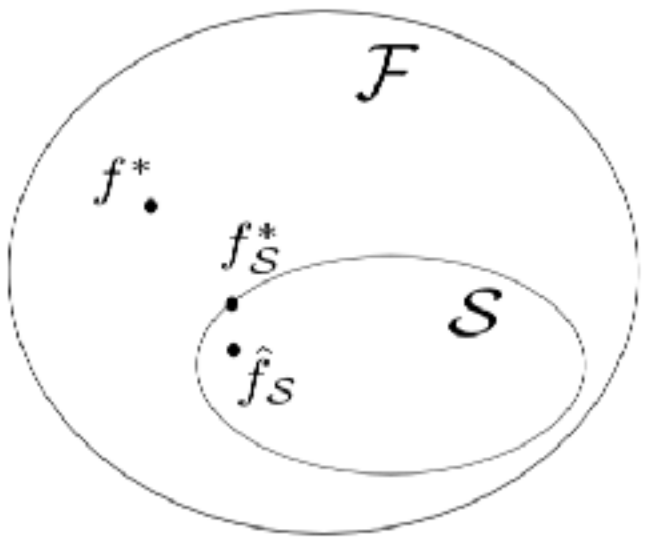
$$\widehat{f} = f_{\widehat{\theta}} = \operatorname*{argmin}_{f_\theta, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(\mathbf{X}_i))$$

- Example: univariate linear regression!

- General setting:
  - $\mathcal{F} = \{\text{measurable fonctions } \mathcal{X} \to \mathcal{Y}\}$
  - Best solution: $f^* = \text{argmin}_{f \in \mathcal{F}} \mathcal{R}(f)$
  - Class $\mathcal{S} \subset \mathcal{F}$ of functions
  - Ideal target in $\mathcal{S}$: $f^*_{\mathcal{S}} = \text{argmin}_{f \in \mathcal{S}} \mathcal{R}(f)$
  - Estimate in $\mathcal{S}$: $\widehat{f}_{\mathcal{S}}$ obtained with some procedure



- General setting:
  - $\mathcal{F} = \{\text{measurable fonctions } \mathcal{X} \to \mathcal{Y}\}$
  - Best solution: $f^* = \text{argmin}_{f \in \mathcal{F}} \mathcal{R}(f)$
  - Class $\mathcal{S} \subset \mathcal{F}$ of functions
  - Ideal target in $\mathcal{S}$: $f^*_{\mathcal{S}} = \text{argmin}_{f \in \mathcal{S}} \mathcal{R}(f)$
  - Estimate in $\mathcal{S}$: $\widehat{f}_{\mathcal{S}}$ obtained with some procedure

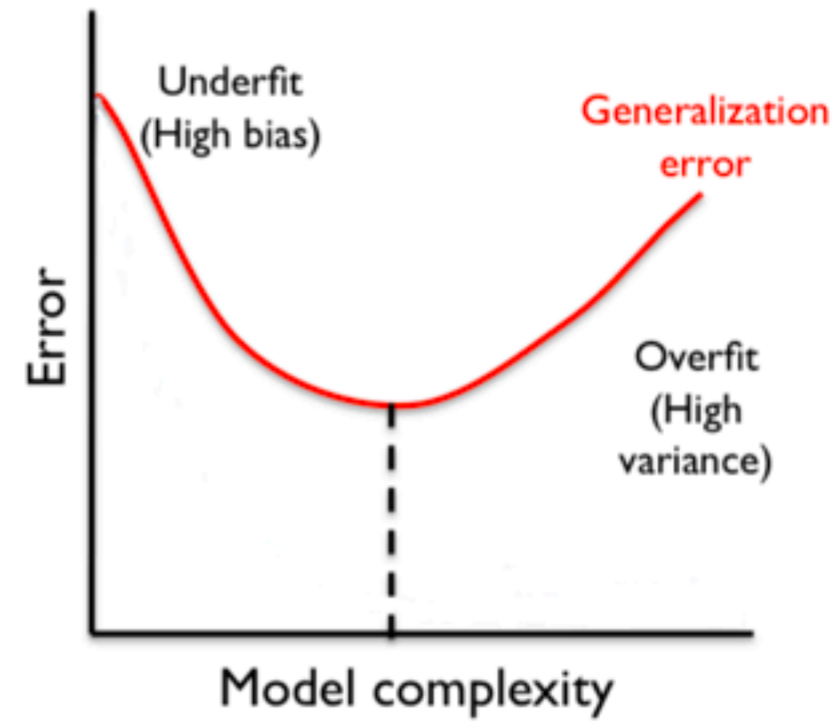


Approximation error and estimation error (Bias/Variance)

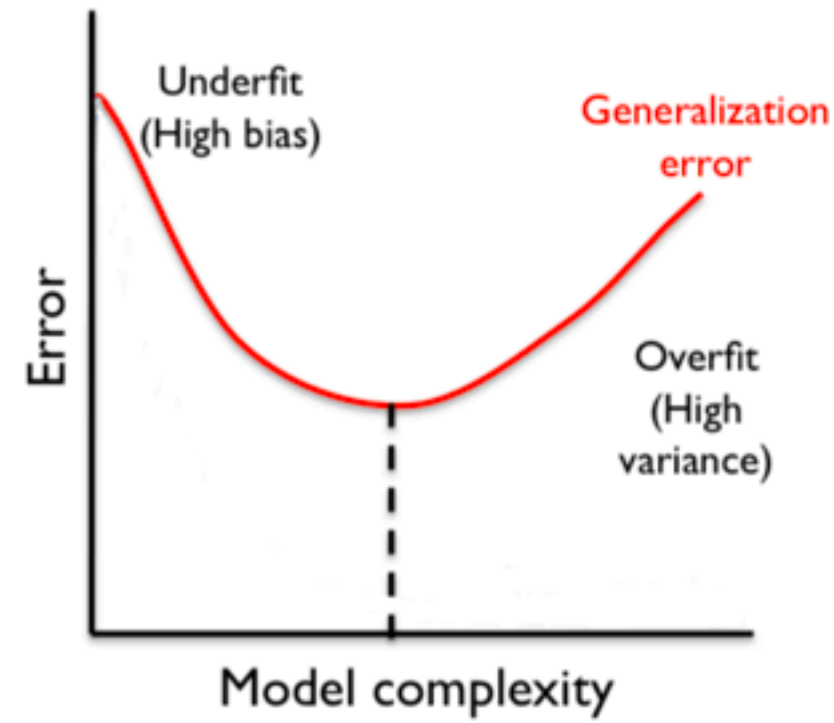
$$\mathcal{R}(\widehat{f}_{\mathcal{S}}) - \mathcal{R}(f^*) = \underbrace{\mathcal{R}(f^*_{\mathcal{S}}) - \mathcal{R}(f^*)}_{\text{Approximation error}} + \underbrace{\mathcal{R}(\widehat{f}_{\mathcal{S}}) - \mathcal{R}(f^*_{\mathcal{S}})}_{\text{Estimation error}}$$

- Approx. error can be large if the model $\mathcal{S}$ is not suitable.
- Estimation error can be large if the model is complex.

Bias-Variance Trade-off



- Different behavior for different model complexity
- Low complexity model are easily learned but the approximation error ("bias") may be large (Under-fit).
- High complexity model may contains a good ideal target but the estimation error ("variance") can be large (Over-fit)



- Different behavior for different model complexity
- Low complexity model are easily learned but the approximation error ("bias") may be large (Under-fit).
- High complexity model may contains a good ideal target but the estimation error ("variance") can be large (Over-fit)

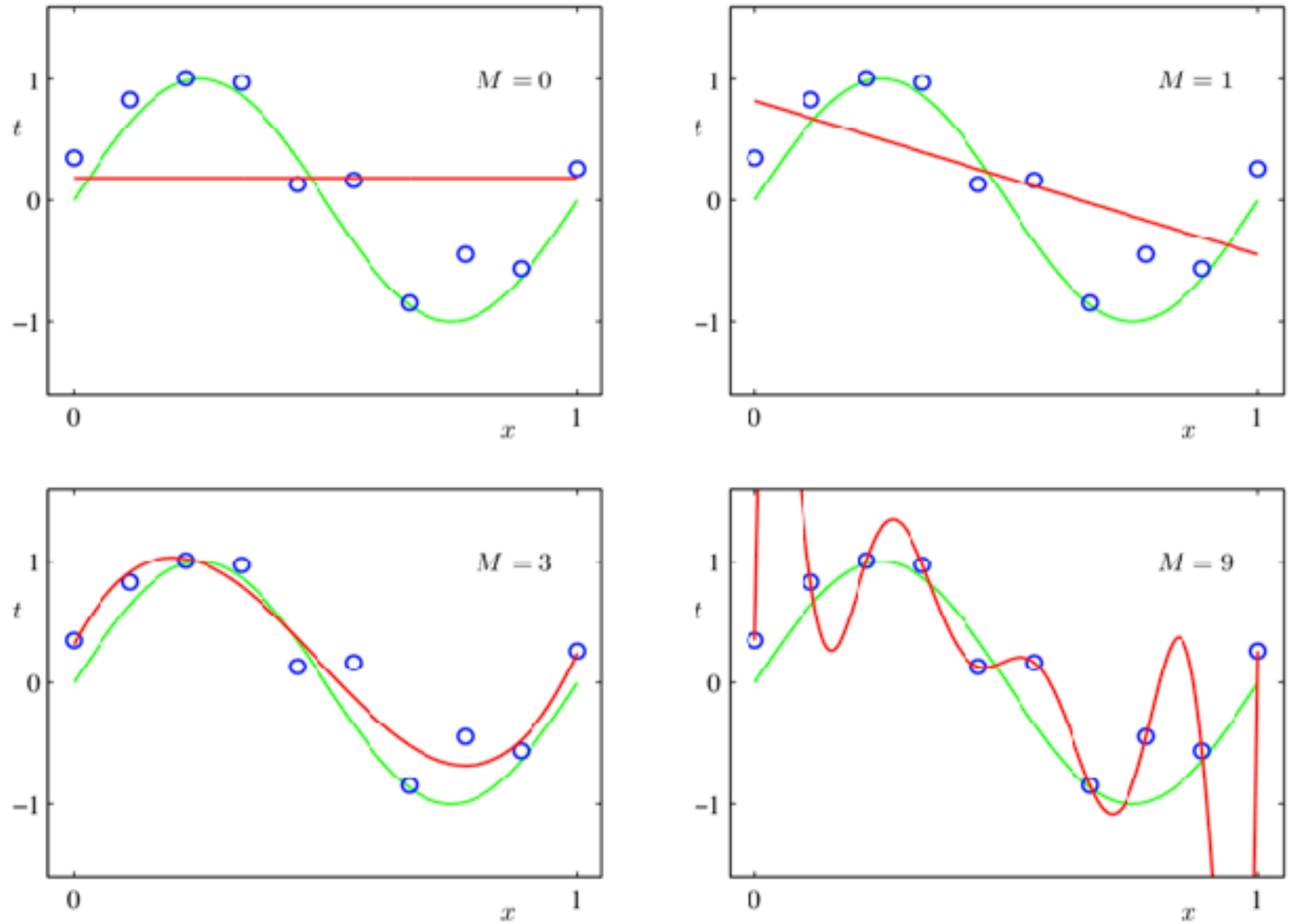
Bias-variance trade-off $\Longleftrightarrow$ avoid overfitting and underfitting

Agnostic approach

- No assumption (so far) on the law of $(\mathbf{X}, Y)$.

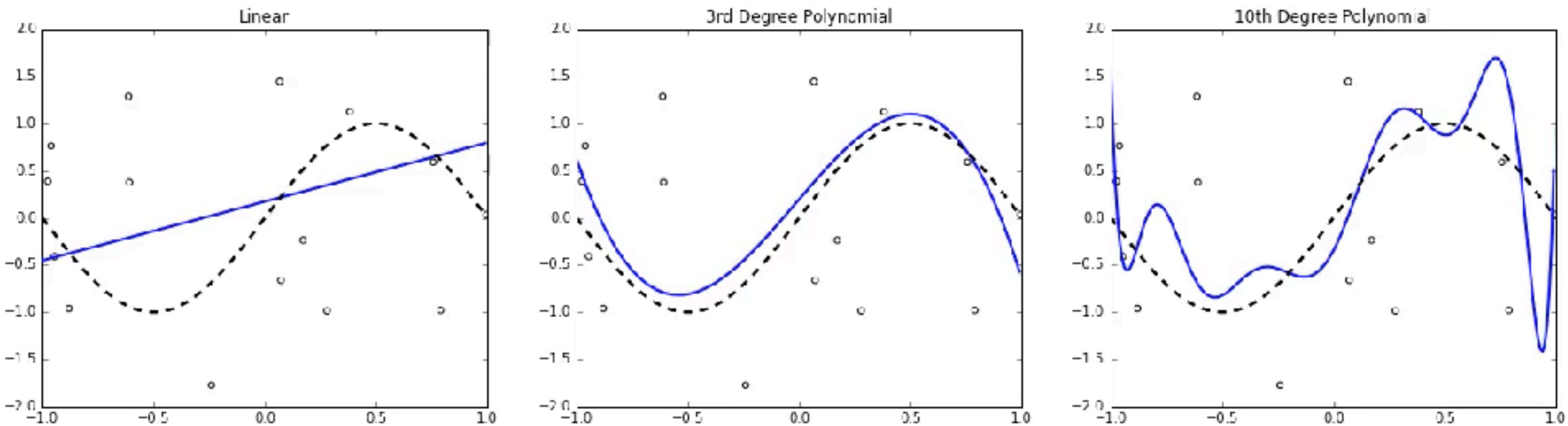
Model of the form $Y = w_0 + w_1 X + w_2 X^2 + \ldots + w_p X^p + \varepsilon$

$$\min_{\boldsymbol{w}} \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - (w_0 + w_1 x_i + w_2 x_i^2 + \ldots + w_p x_i^p)\right)^2$$



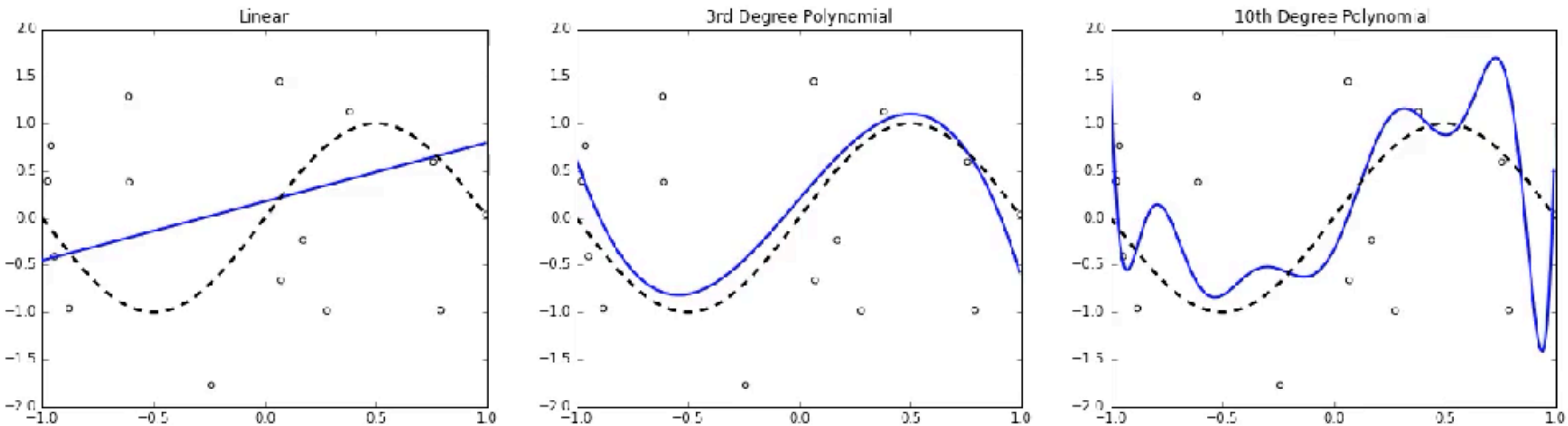
From large bias to overfitting ….

… a quest for optimality



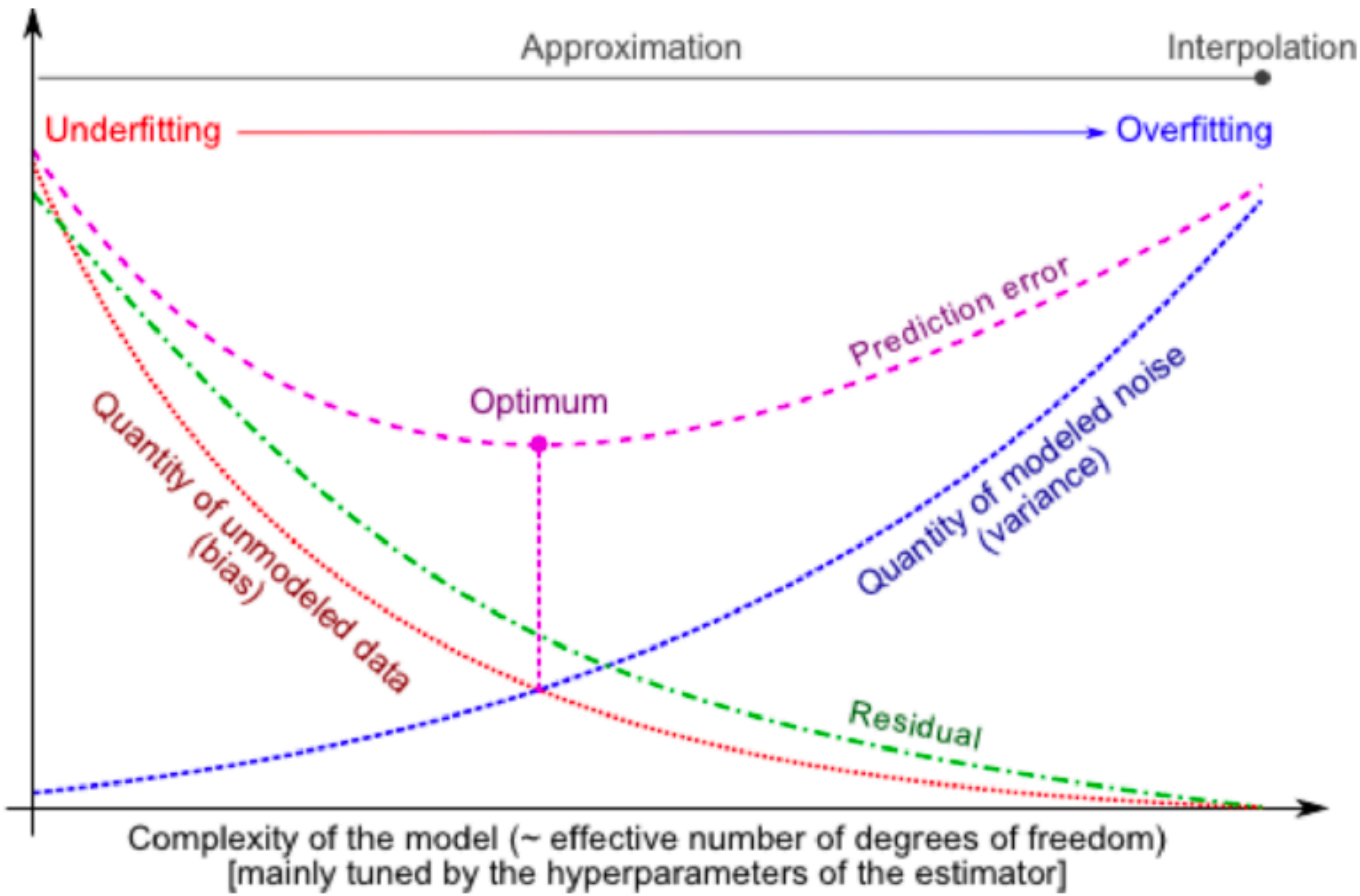
Empirical Risk Minimizer on different Models

… a quest for optimality



Empirical Risk Minimizer on different Models

From large bias to overfitting ....



Statistical Learning Analysis

- Error decomposition:

$$\mathcal{R}(\widehat{f}_S) - \mathcal{R}(f^*) = \underbrace{\mathcal{R}(f_S^*) - \mathcal{R}(f^*)}_{\text{Approximation error}} + \underbrace{\mathcal{R}(\widehat{f}_S) - \mathcal{R}(f_S^*)}_{\text{Estimation error}}$$

- Bound on the approximation term: approximation theory.
- Probabilistic bound on the estimation term: probability theory!
- **Goal:** Agnostic bounds, i.e. bounds that do not require assumptions on $\mathbf{P}$! (Statistical Learning?)

Approximation vs Estimation

Statistical Learning Analysis

- Error decomposition:

$$\mathcal{R}(\widehat{f}_S) - \mathcal{R}(f^*) = \underbrace{\mathcal{R}(f_S^*) - \mathcal{R}(f^*)}_{\text{Approximation error}} + \underbrace{\mathcal{R}(\widehat{f}_S) - \mathcal{R}(f_S^*)}_{\text{Estimation error}}$$

- Bound on the approximation term: approximation theory.
- Probabilistic bound on the estimation term: probability theory!
- **Goal:** Agnostic bounds, i.e. bounds that do not require assumptions on **P**! (Statistical Learning?)

- Often need mild assumptions on **P**... (Nonparametric Statistics?)

How to find a good function f that makes small

$$R(f) = \mathbb{E}\left[\ell(Y, f(X))\right] \quad ?$$

Canonical approach: $\widehat{f}_{\mathcal{S}} = \text{argmin}_{f \in \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(\mathbf{X}_i))$

Problems

- How to choose $\mathcal{S}$?
- How to compute the minimization?

How to find a good function f that makes small

$$R(f) = \mathbb{E}\left[\ell(Y, f(X))\right] \quad ?$$

Canonical approach: $\widehat{f}_{\mathcal{S}} = \operatorname{argmin}_{f \in \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(\mathbf{X}_i))$

Problems

- How to choose $\mathcal{S}$?
- How to compute the minimization?

Statistical Point of View

Solution: For $\mathbf{X}$, estimate $Y|\mathbf{X}$ plug this estimate in the Bayes classifier: (generalized) linear models, kernel methods, k-nn, naive Bayes...

Optimization Point of View

Solution: If necessary replace the loss ℓ by an upper bound ℓ' and minimize the empirical loss: SVR, SVM, Neural Network, Boosting

Supervised Learning

Linear Regression

Experience, Task and Performance measure

- Training data : $\mathcal{D} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- Predictor: $f : \mathbb{R}^d \to \mathbb{R}$ measurable
- Cost/Loss function : $\ell(Y, f(\mathbf{X})) = |f(\mathbf{X}) - Y|^2$ measure how well $f(\mathbf{X})$ "predicts" Y
- Risk:

$$\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{X}))\right]\right]$$

$$\mathbb{E}\left[|Y - f(\mathbf{X})|^2\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|\mathbf{X}}\left[|Y - f(\mathbf{X})|^2\right]\right]$$

Experience, Task and Performance measure

- Training data : $\mathcal{D} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- Predictor: $f : \mathbb{R}^d \to \mathbb{R}$ measurable
- Cost/Loss function : $\ell(Y, f(\mathbf{X})) = |f(\mathbf{X}) - Y|^2$ measure how well $f(\mathbf{X})$ "predicts" Y
- Risk:

$$\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{X}))\right]\right]$$

$$\mathbb{E}\left[|Y - f(\mathbf{X})|^2\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|\mathbf{X}}\left[|Y - f(\mathbf{X})|^2\right]\right]$$

Goal

- Learn a rule to construct a predictor $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. the risk $\mathcal{R}(\widehat{f})$ is small on average or with high probability with respect to $\mathcal{D}_n$.

Linear Model

- Prediction model:

$$f_\beta(\mathbf{X}) = \sum_{j=1}^{p} \beta_j \mathbf{X}_j = \langle \mathbf{X}, \beta \rangle$$

with an unknown parameter $\beta \in \mathbb{R}^p$

Losses

- Quadratic loss: $\ell(Y, f(\mathbf{X})) = \mathbb{E}\left[|Y - \langle \mathbf{X}, \beta \rangle|^2\right]$
- Empirical quadratic loss:

$$\frac{1}{n}\sum_{i=1}^{n} |Y_i - \langle \mathbf{X}_i, \beta \rangle|^2$$

Minimizer

- Loss minimizer:

$$\beta^\dagger = \operatorname{argmin} \mathbb{E}\left[|Y - \langle \mathbf{X}, \beta \rangle|^2\right]$$

- Empirical loss minimizer:

$$\widehat{\beta} = \operatorname{argmin} \frac{1}{n}\sum_{i=1}^{n} |Y_i - \langle \mathbf{X}_i, \beta \rangle|^2$$

- Empirical loss minimization: easy problem with an explicit

Optimization heuristic

- Minimizing the empirical loss

$$\frac{1}{n}\sum_{i=1}^{n}|Y_i - \langle \mathbf{X}_i, \beta\rangle|^2.$$

  is a good idea.
- This can easily be done here!

Optimization heuristic

- Minimizing the empirical loss

$$\frac{1}{n}\sum_{i=1}^{n}|Y_i - \langle \mathbf{X}_i, \beta\rangle|^2.$$

  is a good idea.
- This can easily be done here!

Statistical heuristic

- Estimating $\mathbb{E}[Y|X]$ is a good idea.
- A natural estimate (if we assume finite second order moments) is provided by the least squares approach (quadratic contrast minimization...)

- The two approaches does not always coincide. (classification!)

- Capitalize on $\langle \mathbf{X}, \beta \rangle = \mathbf{X}^t \beta$

Matrix rewriting

- Denoting

$$\mathbf{X}_{(n)} = \begin{pmatrix} \mathbf{X}_1^t \\ \vdots \\ \mathbf{X}_n^t \end{pmatrix} \quad \text{and} \quad \mathbf{Y}_{(n)} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

we obtain

$$\widehat{\beta} = \operatorname{argmin} \|\mathbf{Y}_{(n)} - \mathbf{X}_{(n)} \beta\|^2.$$

- Capitalize on $\langle \mathbf{X}, \beta \rangle = \mathbf{X}^t \beta$

Matrix rewriting

- Denoting

$$\mathbf{X}_{(n)} = \begin{pmatrix} \mathbf{X}_1^t \\ \vdots \\ \mathbf{X}_n^t \end{pmatrix} \quad \text{and} \quad \mathbf{Y}_{(n)} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

we obtain

$$\widehat{\beta} = \operatorname{argmin} \|\mathbf{Y}_{(n)} - \mathbf{X}_{(n)}\beta\|^2.$$

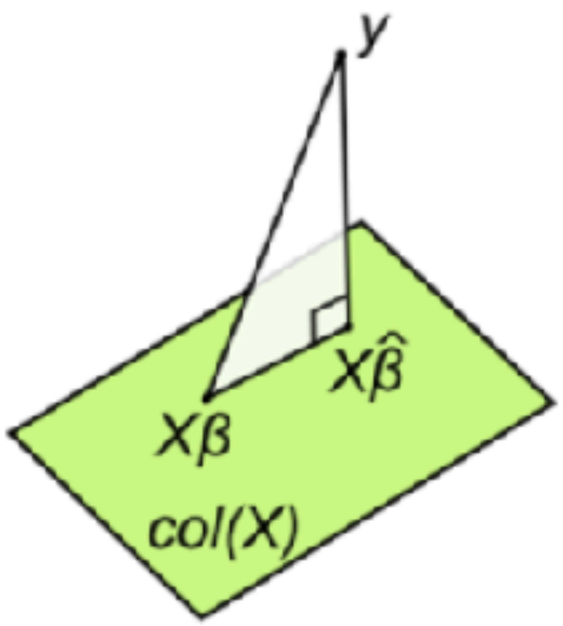
Optimization

- First order optimality condition:

$$2\mathbf{X}_{(n)}^t(\mathbf{Y}_{(n)} - \mathbf{X}_{(n)}\beta) = 0 \Leftrightarrow \mathbf{X}_{(n)}^t\mathbf{X}_{(n)}\beta = \mathbf{X}_{(n)}^t\mathbf{Y}_{(n)}$$

- If $\mathbf{X}_{(n)}^t\mathbf{X}_{(n)}$ is invertible, the unique solution is given by

$$\widehat{\beta} = (\mathbf{X}_{(n)}^t\mathbf{X}_{(n)})^{-1}\mathbf{X}_{(n)}^t\mathbf{Y}_{(n)}$$



Prediction = Projection

- $\mathbf{X}_{(n)}\widehat{\beta}$ is the orthonormal projection of $\mathbf{Y}_{(n)}$ onto the space spanned by the column of $\mathbf{X}_{(n)}$.

Non unique solution

- If $\mathbf{X}_{(n)}$ is not full rank, the minimizer is not unique but every solution yields the same prediction at the observation points.
- Beware: The predictions may differ on non observation points!

Best $f_{\mathcal{S}} \in \mathcal{S}$

- General case:

$$\mathbb{E}\left[\left|Y - f_{\mathcal{S}}(\mathbf{X})\right|^2\right] = \min_{f \in \mathcal{S}} \underbrace{\mathbb{E}\left[\left|f^\star(\mathbf{X}) - f(\mathbf{X})\right|^2\right]}_{\text{Approx. error}} + \underbrace{\mathbb{E}\left[|\varepsilon|^2\right]}_{\text{Variability}}$$

- Issue: the best choice requires the knowledge of both $f^\star(\mathbf{X})$ and the law of $\mathbf{X}$!

Linear prediction

- Model: $f_\beta(\mathbf{X}) = \langle \mathbf{X}, \beta \rangle$

$$\mathbb{E}\left[\left|Y - f_\beta(\mathbf{X})\right|^2\right] = \mathbb{E}\left[\left|f^\star(\mathbf{X}) - \langle \mathbf{X}, \beta \rangle\right|^2\right] + \mathbb{E}\left[|\varepsilon|^2\right]$$

- Best linear prediction: $f_{\beta^\dagger}$ with

$$\beta^\dagger = \operatorname*{argmin}_{\beta} \underbrace{\mathbb{E}\left[\left|f^\star(\mathbf{X}) - \langle \mathbf{X}, \beta \rangle\right|^2\right]}_{\text{Approx. error}} + \underbrace{\mathbb{E}\left[|\varepsilon|^2\right]}_{\text{Variability}}$$

Empirical Risk Minimizer Case

- $\hat{f} = \operatorname{argmin}_{f \in \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} |Y_i - f(\mathbf{X}_i)|^2$
- $R_n(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \hat{f}(\mathbf{X}_i)|^2$
- No independence between $\hat{f}$ and $(\mathbf{X}_i, Y_i)$!
- **Intuitively** $R_n(\hat{f})$ should be optimistic...:

$$\mathbb{E}\left[R_n(\widehat{f})\right] = \mathbb{E}\left[\inf_{f \in \mathcal{S}} R_n(f)\right] \leq \inf_{f \in \mathcal{S}} \mathbb{E}\left[R_n(f)\right] = \inf_{f \in \mathcal{S}} R(f) = R(f^\dagger)$$

Two directions

- Find a way to *correct* $R_n(\hat{f})$?
- *Estimate* $R(\hat{f})$ in a different way?

Find a way to *correct* $R_n(\hat{f})$

- **Bias correction**: Find a correction $\mathrm{cor}(\hat{f})$ such that

$$R(\hat{f}) \sim R_n(\hat{f}) + \mathrm{cor}(\hat{f}).$$

- **Rk**: An upper bound is already interesting.
- **Issue**:No easy way to construct such a bound without further assumptions...

Find a way to *correct* $R_n(\hat{f})$

- **Bias correction**: Find a correction $\mathrm{cor}(\hat{f})$ such that

$$R(\hat{f}) \sim R_n(\hat{f}) + \mathrm{cor}(\hat{f}).$$

- **Rk**: An upper bound is already interesting.
- **Issue**:No easy way to construct such a bound without further assumptions...

Estimate $R(\hat{f})$ in a different way

- **Naive idea**: use another sample to estimate the error...
- Impossible by definition!
- **Cross Validation**: split the sample in two, learn with one part and estimate the error with the other one.
- **Issue**: not exactly the same estimator (less data is used...)

Supervised Learning

Classification and Logistic Regression

- Input: a data set $\mathcal{D}_n$
  Learn $Y|x$ or equivalently $p_k(\mathbf{x}) = \mathbb{P}\{Y = k|\mathbf{X} = \mathbf{x}\}$ (using the data set) and plug this estimate in the Bayes classifier

- Output: a classifier $\hat{f} : \mathbb{R}^d \to \{-1, 1\}$

$$\hat{f}(\mathbf{x}) = \begin{cases} +1 & \text{if } \widehat{p}_{+1}(\mathbf{x}) \geq \widehat{p}_{-1}(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases}$$

- Three instantiations:
  1. Generative Modeling (Bayes method)
  2. Logistic modeling (parametric method)
  3. Nearest neighbors (kernel method)

Bayes formula

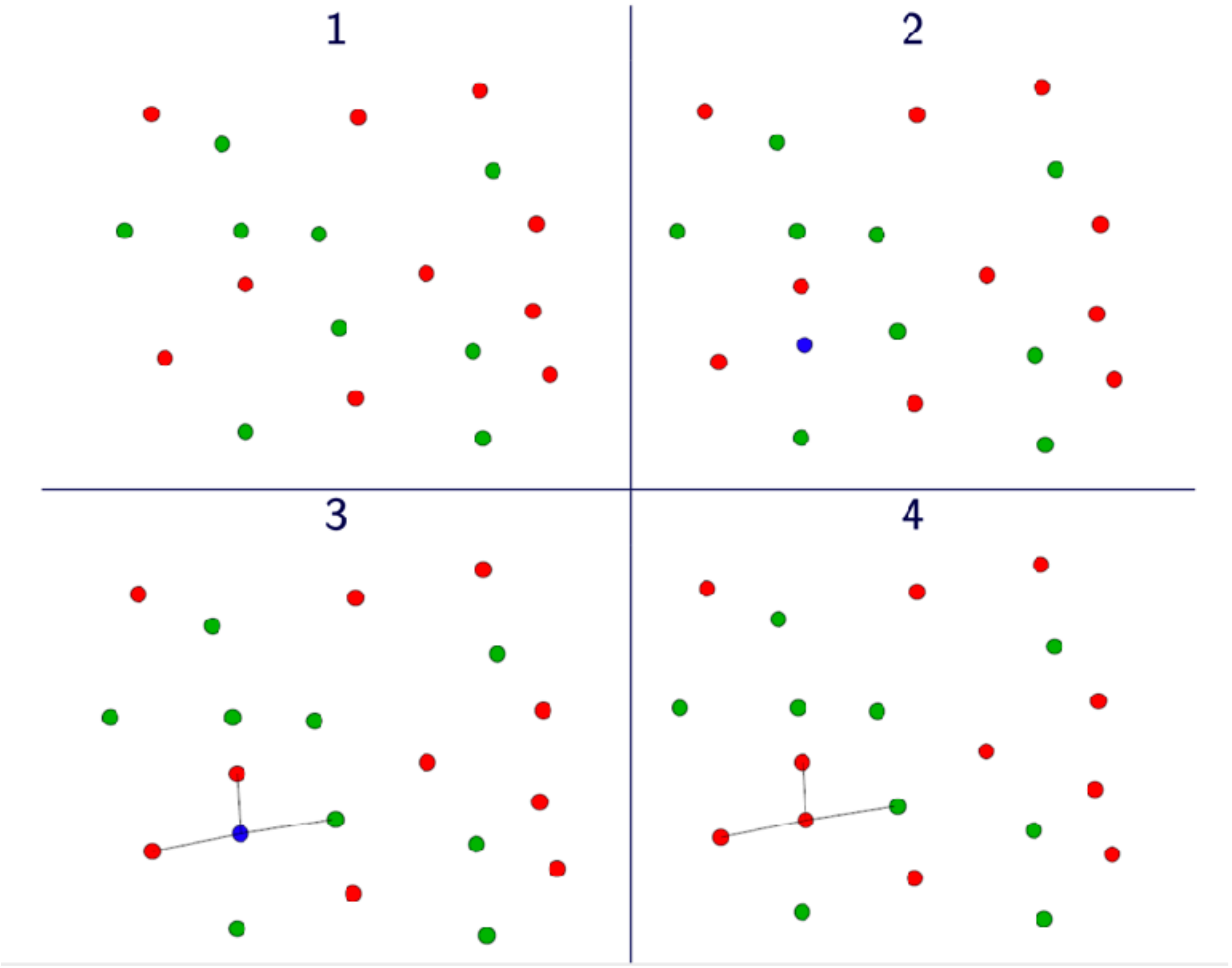
$$p_k(\mathbf{x}) = \frac{\mathbb{P}\left\{\mathbf{X} = \mathbf{x} \middle| Y = k\right\} \mathbb{P}\left\{Y = k\right\}}{\mathbb{P}\left\{\mathbf{X} = \mathbf{x}\right\}}$$

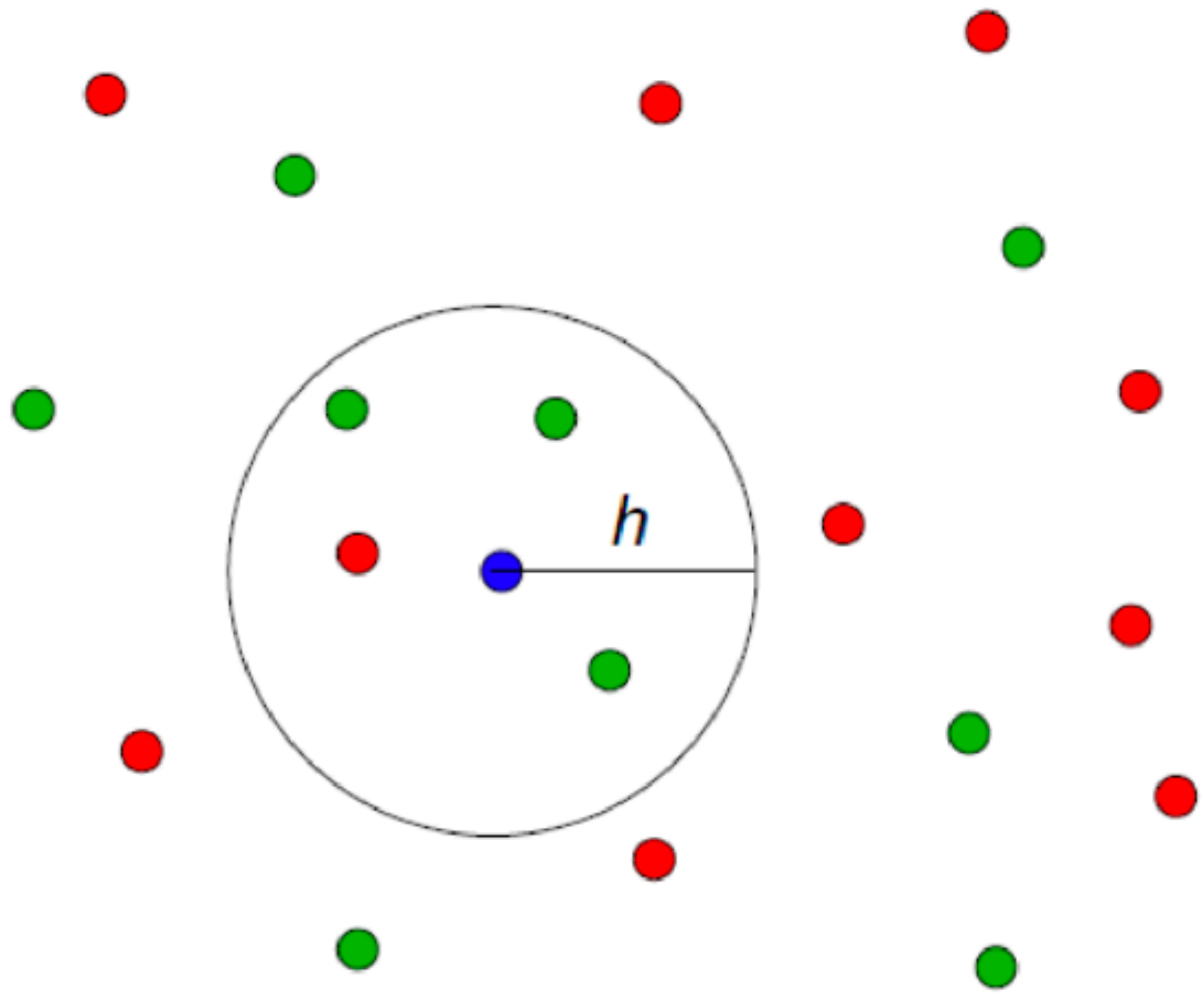
Remark: If one knows the law of (X, Y) or equivalently of X given y and of Y then everything is easy!

- Binary Bayes classifier (the best solution)

$$f^*(\mathbf{x}) = \begin{cases} +1 & \text{if } p_{+1}(\mathbf{x}) \geq p_{-1}(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases}$$

- **Heuristic**: Estimate those quantities and plug the estimations.
- By using different models for $\mathbb{P}\left\{\mathbf{X} \middle| Y\right\}$, we get different classifiers.
- **Remark:** You can also use your favorite density estimator...


1
2
3
4


h

- Neighborhood $\mathcal{V}_{\mathbf{x}}$ of $\mathbf{x}$: k closest from $\mathbf{x}$ learning samples.

k-NN as local conditional density estimate

$$\widehat{p}_{+1}(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in \mathcal{V}_{\mathbf{x}}} \mathbf{1}_{\{y_i=+1\}}}{|\mathcal{V}_{\mathbf{x}}|}$$

- KNN Classifier:

$$\widehat{f}_{KNN}(\mathbf{x}) = \begin{cases} +1 & \text{if } \widehat{p}_{+1}(\mathbf{x}) \geq \widehat{p}_{-1}(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases}$$

- **Remark:** You can also use your favorite kernel estimator...

Linear Classifier

- Classifier family:

$$\mathcal{S} = \{f_\theta : \mathbf{x} \mapsto \mathtt{sign}\{\beta^T \mathbf{x} + \beta_0\} \,/\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}\}$$

- Natural loss: $\ell^{0/1}(Y, f(x)) = \mathbf{1}_{y \neq f(x)}$

Plugin Classification

Linear Classifier

- Classifier family:

$$\mathcal{S} = \{f_\theta : \mathbf{x} \mapsto \texttt{sign}\{\beta^T \mathbf{x} + \beta_0\} / \beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}\}$$

- Natural loss: $\ell^{0/1}(Y, f(x)) = \mathbf{1}_{y \neq f(x)}$

Empirical Risk Minimization

- ERM Classifier:

$$\widehat{f} = f_{\widehat{\theta}} = \underset{f_\theta, \theta \in \Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{Y_i \neq f_\theta(\mathbf{x}_i))}$$

- Not smooth or convex $\Longrightarrow$ no easy minimization scheme!
- $\neq$ regression with quadratic loss case!

- How to go beyond?

Bayes Classifier and Plugin

- Best classifier given by

$$f^*(\mathbf{X}) = \begin{cases} +1 & \text{if } \mathbb{P}\{Y = +1|\mathbf{X}\} \geq \mathbb{P}\{Y = -1|\mathbf{X}\} \\ & \Leftrightarrow \mathbb{P}\{Y = +1|\mathbf{X}\} \geq 1/2 \\ -1 & \text{otherwise} \end{cases}$$

- Plugin classifier: replace $\mathbb{P}\{Y = +1|\mathbf{X}\}$ by a data driven estimate $\widehat{\mathbb{P}\{Y = +1|\mathbf{X}\}}$!

- Other strategies are possible (Risk convexification...)

Plugin Linear Discrimination

- Model $\mathbb{P}\{Y = +1|\mathbf{X}\}$ by $h(\beta^T\mathbf{X} + \beta_0)$ with h non decreasing.
- $h(\beta^T\mathbf{X} + \beta_0) > 1/2 \Leftrightarrow \beta^T\mathbf{X} + \beta_0 - h^{-1}(1/2) > 0$
- Linear Classifier: $\text{sign}(\beta^T\mathbf{X} + \beta_0 - h^{-1}(1/2))$

Plugin Linear Discrimination

- Model $\mathbb{P}\{Y=+1|\mathbf{X}\}$ by $h(\beta^T\mathbf{X}+\beta_0)$ with h non decreasing.
- $h(\beta^T\mathbf{X}+\beta_0) > 1/2 \Leftrightarrow \beta^T\mathbf{X}+\beta_0 - h^{-1}(1/2) > 0$
- Linear Classifier: $\operatorname{sign}(\beta^T\mathbf{X}+\beta_0 - h^{-1}(1/2))$

Plugin Linear Classifier Estimation

- Classical choice for h:

$$h(t) = \frac{e^t}{1+e^t} \qquad \text{logit or logistic}$$

$$h(t) = F_{\mathcal{N}}(t) \qquad \text{probit}$$

$$h(t) = 1 - e^{-e^t} \qquad \text{log-log}$$

- Choice of the *best* β from the data.

- Need to specify the quality criterion...

Logistic Regression and Odd

- Logistic model: $h(t) = \frac{e^t}{1+e^t}$ (most *natural* choice...)
- The Bernoulli law $\mathcal{B}(h(t))$ satisfies then

$$\frac{\mathbb{P}\{Y=1\}}{\mathbb{P}\{Y=-1\}} = e^t \Leftrightarrow \log \frac{\mathbb{P}\{Y=1\}}{\mathbb{P}\{Y=-1\}} = t$$

- Interpretation in term of odd.
- Logistic model: linear model on the logarithm of the odd.

Associated Classifier

- Plugin strategy:

$$f_\beta(x) = \begin{cases} 1 & \text{if } \frac{e^{x^t\beta}}{1+e^{x^t\beta}} > 1/2 \Leftrightarrow x^t\beta > 0 \\ -1 & \text{otherwise} \end{cases}$$

Likelikood Rewriting

- Opposite of the log-likelihood:

$$
\begin{aligned}
& -\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{1}_{y_i=1}\log(h(x_i^t\beta)) + \mathbf{1}_{y_i=-1}\log(1-h(x_i^t\beta))\right) \\
&= -\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{1}_{y_i=1}\log\frac{e^{x_i^t\beta}}{1+e^{x_i^t\beta}} + \mathbf{1}_{y_i=-1}\log\frac{1}{1+e^{x_i^t\beta}}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\log\left(1+e^{-y_i(x_i^t\beta)}\right)
\end{aligned}
$$

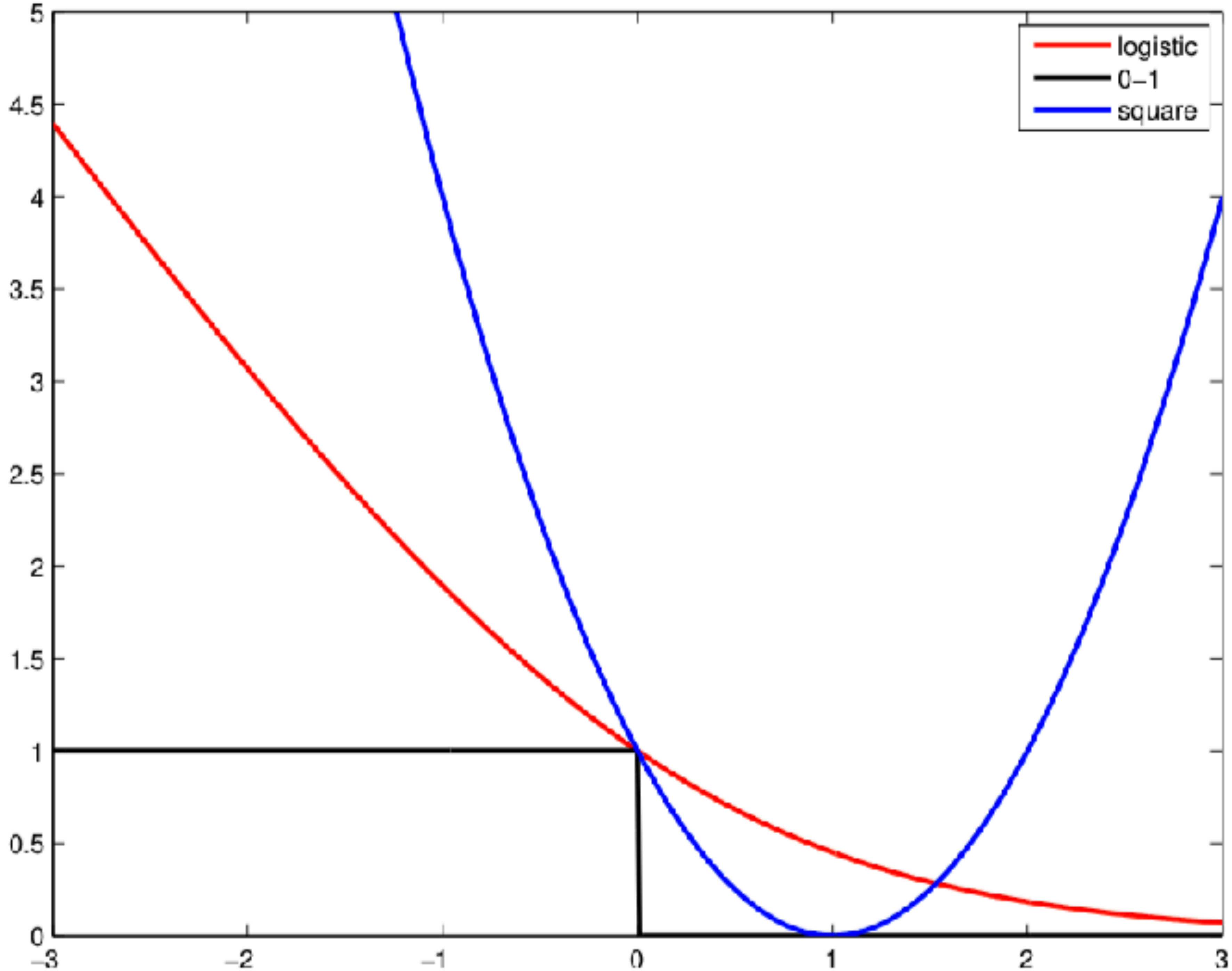
- Convex and smooth function of β
- Easy optimization.

Risk Convexification Heuristic

- **Prop:** $\ell^{0/1}(y_i, f_\beta(x_i)) = \mathbf{1}_{y_i(x_i^t\beta)<0} \leq \frac{\log\left(1 + e^{-y_i(x_i^t\beta)}\right)}{\log 2}$
- Link between the empirical prediction loss and the likelihood:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{y_i \neq f_\beta(x_i)} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{y_i(x_i^t\beta)<0} \leq \frac{1}{n\log 2}\sum_{i=1}^{n}\log\left(1 + e^{-y_i(x_i^t\beta)}\right)$$

- Logistic: easy minimization of the right hand instead of the untractable left hand side...



$\ell(a, 1)$ for several classification losses