# MACHINE LEARNING & DATA MINING

image from: theportalwiki.com

# Model selection and assessment

Model selection:

estimating the performance of a set of models to choose the best one.

Model assessment:

once having selected the best model, estimating its prediction error on new data.

# Model selection and assessment

Model selection:

estimating the performance of a set of models to choose the best one.

Model assessment:

once having selected the best model, estimating its prediction error on new data.

- How do you do it when you have plenty of data?

# The nature of the test error

Expected error at a given point $x_0$:

$$\mathrm{Err}(x_0) \equiv \mathop{\mathbb{E}}_{\tau, Y | X = x_0} \left[ L(Y, \hat{f}(x_0, \tau) \right]$$

$\tau$ — training dataset

# The nature of the test error

*Example: MSE*

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

$$L(y, \hat{y}) = (y - \hat{y})^2$$

$$\mathrm{Err}(x_0) = \mathop{\mathbb{E}}_{\tau, Y | X = x_0} \left[ (Y - \hat{f}(x_0, \tau))^2 \right]$$

$$= \sigma_\epsilon^2 + \left[ f(x_0) - \mathop{\mathbb{E}}_{\tau}[\hat{f}(x_0, \tau)] \right]^2 + \mathop{\mathbb{E}}_{\tau} \left[ \hat{f}(x_0, \tau) - \mathop{\mathbb{E}}_{\tau'}[\hat{f}(x_0, \tau')] \right]^2$$

# The nature of the test error

*Example: MSE*

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

$$L(y, \hat{y}) = (y - \hat{y})^2$$

$$\mathrm{Err}(x_0) = \mathop{\mathbb{E}}_{\tau, Y | X = x_0} \left[ (Y - \hat{f}(x_0, \tau))^2 \right]$$

$$= \underbrace{\sigma_\epsilon^2}_{\textbf{Irreducible error}} + \underbrace{\left[ f(x_0) - \mathop{\mathbb{E}}_{\tau}[\hat{f}(x_0, \tau)] \right]^2}_{\textbf{Bias}^2} + \underbrace{\mathop{\mathbb{E}}_{\tau} \left[ \hat{f}(x_0, \tau) - \mathop{\mathbb{E}}_{\tau'}[\hat{f}(x_0, \tau')] \right]^2}_{\textbf{Variance}}$$

# Example: KNN

$$\hat{f}(x_0) = \frac{1}{k} \sum_{\substack{l=1 \\ y_l \in \mathrm{neighb.}(x_0, k)}}^{k} y_l$$

For simplicity consider $x_l$ fixed when calculating expectation over $\mathcal{T}$

# Example: KNN

$$\hat{f}(x_0) = \frac{1}{k} \sum_{\substack{l=1 \\ y_l \in \mathrm{neighb.}(x_0, k)}}^{k} y_l$$

For simplicity consider $x_l$ fixed when calculating expectation over $\mathcal{T}$

$$\mathrm{Err}(x_0) = \sigma_\epsilon^2 + \left[ f(x_0) - \frac{1}{k} \sum_{l=1}^{k} f(x_l) \right]^2 + \frac{\sigma_\epsilon^2}{k}$$

# Example: Linear Regression

$$\hat{f}(x_0) = x_0^T w$$

$$\hat{f}(x_0) = x_0^T w = x_0^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T y = h^T(x_0) y$$

$$h(x_0) \equiv \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} x_0$$

# Example: Linear Regression

$$\hat{f}(x_0) = x_0^T w$$

$$\hat{f}(x_0) = x_0^T w = x_0^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T y = h^T(x_0) y$$

$$h(x_0) \equiv \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} x_0$$

$$\mathrm{Err}(x_0) = \sigma_\epsilon^2 + \left[ f(x_0) - \mathop{\mathbb{E}}_{\tau} \hat{f}(x_0, \tau) \right]^2 + ||h(x_0)||^2 \sigma_\epsilon^2$$

# Example: Linear Regression

$$\mathrm{Err}(x_0) = \sigma_\epsilon^2 + \left[ f(x_0) - \mathop{\mathbb{E}}_{\tau} \hat{f}(x_0, \tau) \right]^2 + ||h(x_0)||^2 \sigma_\epsilon^2$$

Consider average error over training points $x_l$:

# Example: Linear Regression

$$\mathrm{Err}(x_0) = \sigma_\epsilon^2 + \left[f(x_0) - \mathop{\mathbb{E}}_{\tau} \hat{f}(x_0, \tau)\right]^2 + ||h(x_0)||^2 \sigma_\epsilon^2$$

Consider average error over training points $x_l$:

$$\frac{1}{N}\sum_i \mathrm{Err}(x_i) = \sigma_\epsilon^2 + \frac{1}{N}\sum_i \left[f(x_i) - \mathop{\mathbb{E}}_{\tau_y} \hat{f}(x_i, \tau)\right]^2 + \frac{d}{N}\sigma_\epsilon^2$$

# ‘0–1’ loss case

$$Y \in \{0, 1\} \qquad \Pr(Y = 1 | x_0) = f(x_0)$$

$$G(x) = I\left(f(x) > \frac{1}{2}\right)$$ — optimal Bayes classifier

$$\hat{G}(x) = I\left(\hat{f}(x) > \frac{1}{2}\right)$$ — estimator

# ‘0–1’ loss case

$$\mathrm{Err}(x_0) = \Pr(\hat{G}(x_0) \neq Y)$$

# ‘0–1’ loss case

$$\mathrm{Err}(x_0) = \Pr(\hat{G}(x_0) \neq Y)$$
$$= \Pr(\hat{G} = G,\, G \neq Y) + \Pr(\hat{G} \neq G,\, G = Y)$$

# ‘0–1’ loss case

$$\begin{aligned}\mathrm{Err}(x_0) &= \Pr(\hat{G}(x_0) \neq Y) \\ &= \Pr(\hat{G} = G,\, G \neq Y) + \Pr(\hat{G} \neq G,\, G = Y) \\ &= \Pr(\hat{G} = G)\Pr(G \neq Y) + \Pr(\hat{G} \neq G)\Pr(G = Y)\end{aligned}$$

# ‘0–1’ loss case

$$
\begin{aligned}
\mathrm{Err}(x_0) &= \Pr(\hat{G}(x_0) \neq Y) \\
&= \Pr(\hat{G} = G,\, G \neq Y) + \Pr(\hat{G} \neq G,\, G = Y) \\
&= \Pr(\hat{G} = G)\Pr(G \neq Y) + \Pr(\hat{G} \neq G)\Pr(G = Y) \\
&= (1 - \Pr(\hat{G} \neq G))\Pr(G \neq Y) + \Pr(\hat{G} \neq G)(1 - \Pr(G \neq Y))
\end{aligned}
$$

# ‘0–1’ loss case

$$
\begin{aligned}
\mathrm{Err}(x_0) &= \Pr(\hat{G}(x_0) \neq Y) \\
&= \Pr(\hat{G} = G,\, G \neq Y) + \Pr(\hat{G} \neq G,\, G = Y) \\
&= \Pr(\hat{G} = G)\Pr(G \neq Y) + \Pr(\hat{G} \neq G)\Pr(G = Y) \\
&= (1 - \Pr(\hat{G} \neq G))\Pr(G \neq Y) + \Pr(\hat{G} \neq G)(1 - \Pr(G \neq Y)) \\
&= \Pr(G \neq Y) + \Pr(\hat{G} \neq G)(1 - 2\Pr(G \neq Y))
\end{aligned}
$$

# ‘0–1’ loss case

$$
\begin{aligned}
\mathrm{Err}(x_0) &= \Pr(\hat{G}(x_0) \neq Y) \\
&= \Pr(\hat{G} = G,\, G \neq Y) + \Pr(\hat{G} \neq G,\, G = Y) \\
&= \Pr(\hat{G} = G)\Pr(G \neq Y) + \Pr(\hat{G} \neq G)\Pr(G = Y) \\
&= (1 - \Pr(\hat{G} \neq G))\Pr(G \neq Y) + \Pr(\hat{G} \neq G)(1 - \Pr(G \neq Y)) \\
&= \Pr(G \neq Y) + \Pr(\hat{G} \neq G)(1 - 2\Pr(G \neq Y)) \\
&= \Pr(G \neq Y) + |2f(x_0) - 1|\Pr(\hat{G} \neq G)
\end{aligned}
$$

# ‘0–1’ loss case

$$\mathrm{Err}(x_0) = \Pr(G \neq Y) + |2f(x_0) - 1| \Pr(\hat{G} \neq G)$$

Assume: $\hat{f}(x_0) \sim \mathcal{N}(\mu(x_0), \sigma^2(x_0))$

Then it can be shown:

$$\Pr(\hat{G} \neq G) = \Phi\left(\frac{(\mu(x_0) - \frac{1}{2})\mathrm{sign}(\frac{1}{2} - f(x_0))}{\sigma(x_0)}\right)$$

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$$

# ‘0–1’ loss case

$$\mathrm{Err}(x_0) = \Pr(G \neq Y) + |2f(x_0) - 1| \cdot \Phi\left(\frac{(\mu(x_0) - \frac{1}{2})\mathrm{sign}(\frac{1}{2} - f(x_0))}{\sigma(x_0)}\right)$$

# ‘0–1’ loss case

$$\mathrm{Err}(x_0) = \underbrace{\Pr(G \neq Y)}_{\textbf{Irreducible error}} + |2f(x_0) - 1| \cdot \Phi\left(\frac{\boxed{(\mu(x_0) - \frac{1}{2})\mathrm{sign}(\frac{1}{2} - f(x_0))}}{\boxed{\sigma(x_0)}}\right)$$

**‘*Boundary*’ bias**

**Variance**

# ‘0–1’ loss case

$$\mathrm{Err}(x_0) = \underbrace{\Pr(G \neq Y)}_{\textbf{Irreducible error}} + |2f(x_0) - 1| \cdot \Phi\left(\frac{\boxed{(\mu(x_0) - \frac{1}{2})\mathrm{sign}(\frac{1}{2} - f(x_0))}}{\boxed{\sigma(x_0)}}\right)$$

**‘*Boundary*’ bias**

**Variance**

- Note that depending on the sign of the bias term, increased variance may increase or decrease the overall error

# Optimism of the training error

- The usual training set error estimate:

$$\overline{\mathrm{err}} = \frac{1}{N} \sum_i L(y_i, \hat{f}(x_i))$$

- Compare it with the *in-sample* error:

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{N} \sum_i \mathop{\mathbb{E}}_{Y^0} \left[ L(Y_i^0, \hat{f}(x_i)) \middle| \tau \right]$$

**New targets sampled at the same $x_i$ points**

**Training set fixed**

# Optimism of the training error

- Typically training set error is smaller than the in-sample error, hence the *optimism* is:

$$\mathrm{op} \equiv \mathrm{Err}_{\mathrm{in}} - \overline{\mathrm{err}}$$

- This is a function of training set $\mathcal{T}$
- Finally, the average optimism is:

$$\omega = \mathop{\mathbb{E}}_{\mathcal{T}_y} [\mathrm{op}]$$

**expectation over targets of the training set**

# Optimism of the training error

- For some loss functions (e.g. MSE, 0-1) it can be shown that:

$$\omega = \frac{2}{N} \sum_i \operatorname{cov}_{\tau_y}(\hat{y}_i, y_i)$$

# Example: Linear Regression

- For linear regression this can be simplified:

$$\sum_i \operatorname*{cov}_{\tau_y}(\hat{y}_i, y_i) = d\sigma_\epsilon^2$$

- and hence the expected in-sample error is:

$$\mathop{\mathbb{E}}_{\tau_y}[\mathrm{Err}] = \mathop{\mathbb{E}}_{\tau_y}[\overline{\mathrm{err}}] + \frac{2d}{N}\sigma_\epsilon^2$$

# In-sample error estimates

- Therefore a natural estimator of the in-sample error is (the so called $C_p$ statistic):

$$C_p \equiv \overline{\mathrm{err}} + \frac{2d}{N}\sigma_\epsilon^2$$

- This can be used for model selection
- If we somehow generalize the number of free parameters $d$ (i.e. model complexity, number of degrees of freedom), this can be used for other models as well

# Model complexity

- For linear regression we had:

$$\sum_i \operatorname*{cov}_{\tau_y}(\hat{y}_i, y_i) = d\sigma_\epsilon^2$$

- Therefore its naturally to introduce *the effective number of degrees of freedom* to be:

$$\mathrm{df}(\hat{y}) \equiv \frac{\sum_i \operatorname*{cov}_{\tau_y}(\hat{y}_i, y_i)}{\sigma_\epsilon^2}$$

# Model complexity

- If our model minimizes some function $R(w)$ with a penalty on weights $\alpha||\mathrm{w}||^2$ (e.g. neural net or Ridge Regression), then:

$$\mathrm{df} = \sum_k \frac{\theta_k}{\theta_k + \alpha}$$

- where $\theta_k$ are the eigenvalues of the Hessian matrix:

$$\frac{\partial^2 R}{\partial w \partial w^T}$$

# In-sample error estimates (2)

- We have mentioned the $C_p$ statistic:

$$C_p \equiv \overline{\mathrm{err}} + \frac{2d}{N}\sigma_\epsilon^2$$

- There is a generalization of this criterion, *Akaike information criterion* (named after Hirotugu Akaike), based on the likelihood approach:

$$\mathrm{AIC} = -\frac{2}{N}\mathrm{loglik} + 2 \cdot \frac{d}{N}$$

**Likelihood, maximized on the given training sample**

- AIC estimates the expected (negative log) likelihood if we were to resample the targets, keeping the model parameters from the initial estimation

# In-sample error estimates (3)

- Another criterion, also applicable in the likelihood maximization setting, *Bayesian information criterion*:

$$\mathrm{BIC} = -2 \cdot \mathrm{loglik} + \log N \cdot d$$

- Motivated by Bayesian approach to model selection
- Tends to penalize complex models more heavily, compared to AIC (given that $N > e^2 \approx 7.4$)

# Extra-sample and expected errors

- So far we've been looking at the in-sample error behavior
- Since our model is to be used on data it has not seen before, it's logical to consider the *extra-sample error*:

$$\mathrm{Err}_\tau = \mathop{\mathbb{E}}_{X^0, Y^0}\left[L(Y^0, \hat{f}(X^0))\middle|\, \tau\right]$$

- A way of estimating this value would be:
  - split the data into train-validation-test parts
  - select the best model on the validation set
  - estimate the extra-sample error on the test set
- In the limited data case it's easier to estimate the *expected error*:

$$\mathrm{Err} = \mathop{\mathbb{E}}_{\tau}\left[\mathrm{Err}_\tau\right] = \mathop{\mathbb{E}}_{\tau, X^0, Y^0}\left[L(Y^0, \hat{f}(X^0))\middle|\, \tau\right]$$

# K-Fold Cross-Validation

To estimate the expected error:

- Split the data randomly into $K$ roughly equal-sized parts
- For each part $\mathcal{T}_i$ of these $K$ parts do:
  - Train the model on $\mathcal{T} \setminus \mathcal{T}_i$ – i.e. on everything but $\mathcal{T}_i$
  - calculate the error estimate $e_i$ on $\mathcal{T}_i$
- Estimate the expected error as:

$$\widehat{\mathrm{Err}} = \frac{1}{K} \sum e_i$$

# Question

Consider the following scenario of using K-Fold CV:

- A binary classification problem with # features >> # samples
- Select top M features with maximal correlation with the target
- Using the selected M features, build your model
- Use CV to find the best hyper-parameters and estimate the prediction error of the final model

**Question: would such approach give a reasonable result?**

# K-Fold CV - the right way

- In general, if your model pipeline consists of many steps, CV should be applied to the entire sequence of these steps
- Possible exception: unsupervised steps (i.e. steps not requiring the target values, e.g. scaling the data)

# Vapnik-Chervonenkis Dimension

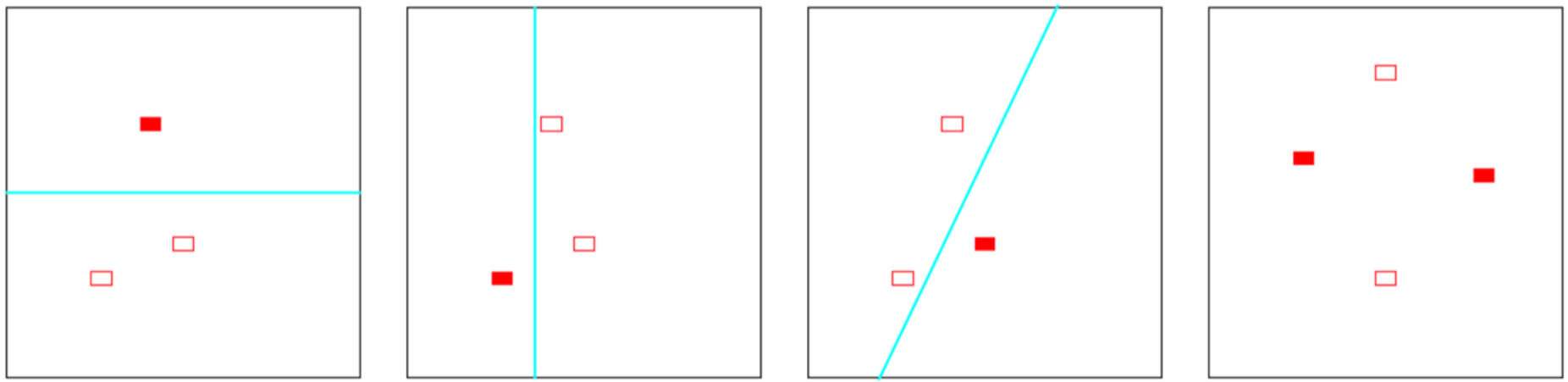
## (a general measure of model complexity)

<u>Definition</u>: a set of points is *shattered* by a class of functions if

- for every binary class label assignment to these points
- there's a function in this class that perfectly separates the classes

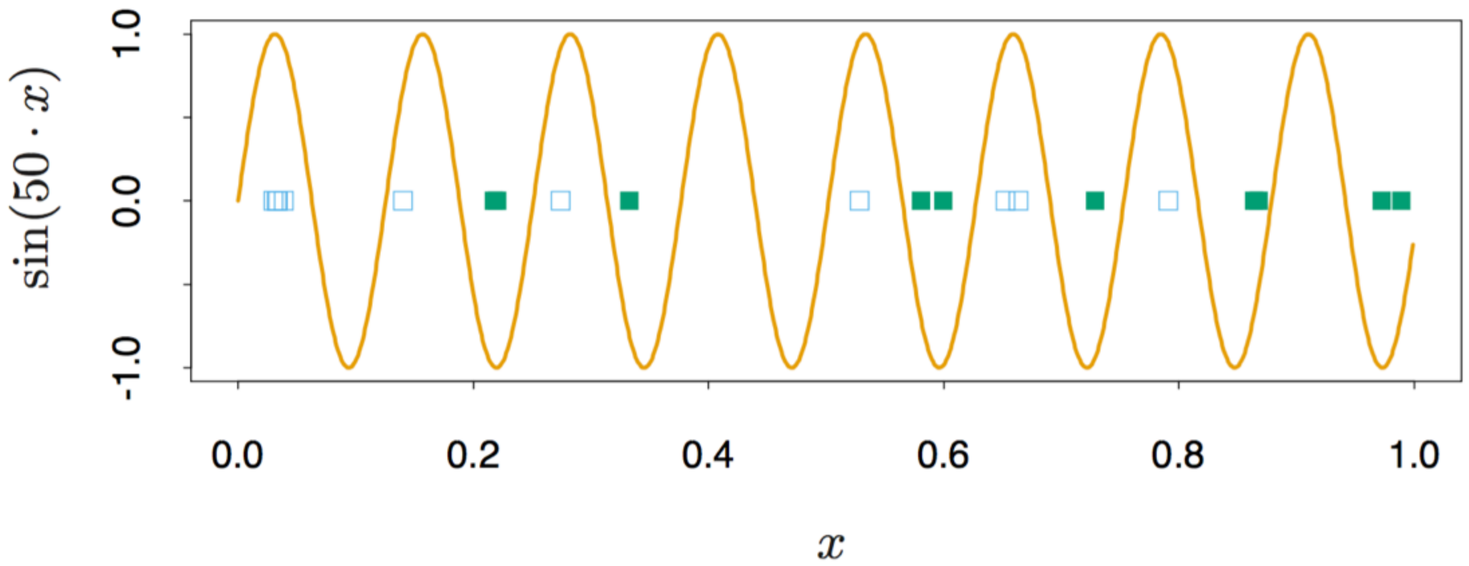
<u>Definition</u>: the *VC dimension* $h$ of the class $C$ of functions is the largest number of points that can be shattered by $C$.

# Examples



- A class of lines on 2D plane has VC dimension $h = 3$
- In general a linear indicator function in p dimensions has $h = p + 1$

# Examples




- A class of functions $I(\sin(\alpha x) > 0)$ has $h = \infty$

# VC dimension for extra-sample error estimation

- Using VC dimension, one can prove results about the optimism of the training error for a particular class of functions
- E.g. a bound for regression:

$$\mathrm{Err}_\tau \leq \overline{\mathrm{err}} \left( 1 - \sqrt{\rho - \rho \log \rho + \frac{\log N}{2N}} \right)_+^{-1}$$

$$\rho = \frac{h}{N}$$

# References


- CV vs AIC vs BIC – example from scikit-learn: https://scikit-learn.org/stable/auto_examples/linear_model/plot_lasso_model_selection.html
- T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Minig, Inference, and Prediction* (Chapter 7), Second Edition, Springer, 2009