

Chapter 2: Classification & Prediction

- ▶ 2.1 Basic Concepts of Classification and Prediction
- ▶ 2.2 Decision Tree Induction
- ▶ 2.3 Bayes Classification Methods
- ▶ 2.4 Rule Based Classification
- ▶ 2.5 Note on SVM (Support Vector Machine)
- ▶ 2.6 Lazy Learners
- ▶ 2.7 Prediction
 - 2.7.1 Definitions
 - 2.7.2 Linear Regression
 - 2.7.3 Nonlinear Regression
 - 2.7.4 Generalized Linear Models: Logistic Regression
- ▶ 2.8 How to Evaluate and Improve Classification
 - 2.7.1 Accuracy and Error Measures
 - 2.7.2 Evaluating a Classifier or Predictor
 - 2.7.3 Increasing the Accuracy

2.7.1 Definitions

- ▶ Numeric Prediction (or prediction) is the task of predicting continuous (or ordered) values for given input
- ▶ **Examples**
 - Given the profile of a customer, predict how much money he will spend
 - Predict the potential sale of a new product given its price
- ▶ The most widely used approach for prediction is **regression**
- ▶ **Regression Analysis**
 - A statistical methodology
 - Used to model the relationship between one or more **independent (predictor)** variable and a **dependent (response)** variable
 - **Predictor variables**: the attributes describing a tuple
 - **Response variable**: what we want to predict
- ▶ Many prediction problems can be solved using **linear regression**
- ▶ A **non-linear** problem can be converted to a linear one

2.7.2 Linear Regression

▶ **Straight-line regression** analysis involves

- A single predictor variable
- A response variable

$$y = b + wx$$

- The **variance** of y is **constant**
- b and w are **regression coefficients**
 - b : Y-intercept
 - w : the slope of the line
- Regression coefficients can also be considered as weights

$$y = \beta_0 + \beta_1 x$$

- Need of estimating the regression coefficients

Method of Least Squares

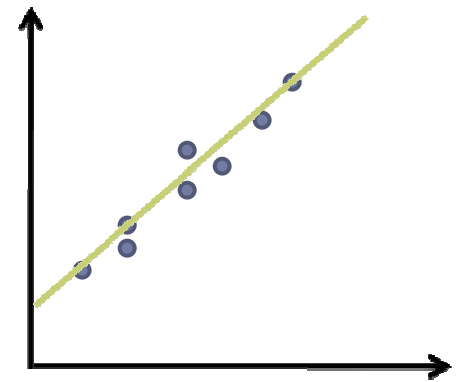
- ▶ Estimate the best-fitting straight line as the one that minimizes the error between the actual data and the estimate of the line
- ▶ Used to solve overdetermined systems (more equations than unknowns)

- ▶ **f is the model function** where

$$y_i = f(x, \beta) = \beta_0 + \beta_1 x$$

- ▶ Minimize the sum, S , of squared **residuals**

$$S = \sum_{i=1}^{|D|} r_i^2 \quad r_i = y_i - f(x_i, \hat{\beta})$$



- ▶ D : a set of training tuples with 1 predictor and 1 response each
 - (x_1, y_1)
 - (x_2, y_2)
 - ...
 - $(x_{|D|}, y_{|D|})$

Method of Least Squares

- ▶ The **minimum** of the sum of squares is found by setting the **gradient** to zero. If the model contains m parameters there are m gradient equations

$$\frac{\partial S}{\partial \beta_j} = 2 \sum_i \frac{\partial r_i}{\partial \beta_j} = 0, \quad j = 1, \dots, m$$

- ▶ When $m=2$, the regression **coefficients** are **estimated** by:

$$\beta_1 = \frac{\sum_{i=1}^{|D|} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|} (x_i - \bar{x})^2} \quad \beta_0 = \bar{y} - \beta_1 \bar{x}$$

where \bar{x} is the mean value of $x_1, x_2, \dots, x_{|D|}$
 \bar{y} is the mean value of $y_1, y_2, \dots, y_{|D|}$

Example

- ▶ Four data points: $(1,6), (2,5), (3,7)$ and $(4,10)$
- ▶ Model these data as $y = f(x, \beta) = \beta_0 + \beta_1 x$
- ▶ Find the parameters that approximately solve:

$$\begin{cases} \beta_0 + 1\beta_1 = 6 \\ \beta_0 + 2\beta_1 = 5 \\ \beta_0 + 3\beta_1 = 7 \\ \beta_0 + 4\beta_1 = 10 \end{cases}$$

$$S = [6 - (\beta_0 + 1\beta_1)]^2 + [5 - (\beta_0 + 2\beta_1)]^2 \\ + [7 - (\beta_0 + 3\beta_1)]^2 + [10 - (\beta_0 + 4\beta_1)]^2$$

- ▶ By determine the partial derivatives of S with respect to β_0 and β_1 and setting them to zero, we find: $\beta_0 = 3.5$ and $\beta_1 = 1.4$

Multiple Linear Regression

- ▶ Involve more than one predictor variables
- ▶ Model a response variable as linear function of **n** predictor variables A_1, A_2, \dots, A_n
- ▶ D: a set of training tuples with **n predictors** and **1 response** each
 - $(X_{11}, X_{12}, \dots, X_{1n}, y_1)$
 - $(X_{21}, X_{22}, \dots, X_{2n}, y_2)$
 - ...
 - $(X_{|D|1}, X_{|D|2}, \dots, X_{|D|n}, y_{|D|})$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

- ▶ The method of least square is used to estimate the coefficients. However the computation becomes long
 - Use statistical software packages (e.g., SAS, SPSS, and S-Plus)

2.7.3 Nonlinear Regression

- ▶ How to model data that does not show a linear dependence?
- ▶ Example: **polynomial regression**
 - Add polynomial terms to the basic linear model
 - Apply transformations to variables
 - Convert the nonlinear model to a linear one

- ▶ Consider a cubic polynomial relationship given by:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

- ▶ To convert this equation to linear form, we define new variables

$$x_1 = x \quad x_2 = x^2 \quad x_3 = x^3$$

The equation becomes

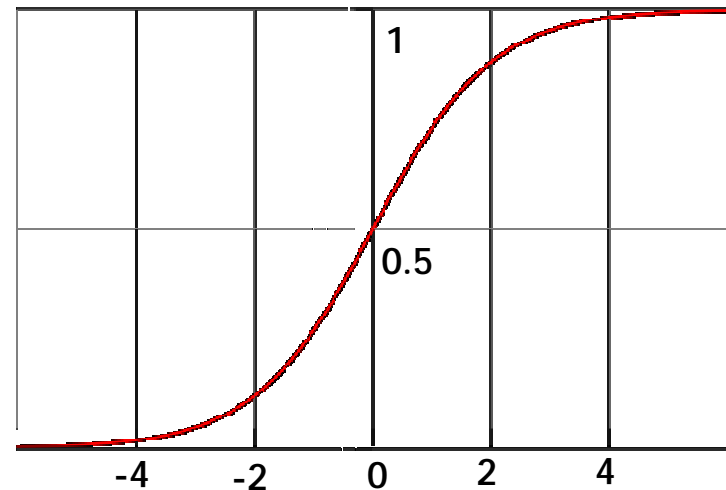
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

2.7.4 Generalized Linear Models

- ▶ Represent the theoretical foundation on which the linear regression can be applied to model classification
- ▶ The variance of the response variable, is a function of the mean value of y , unlike the linear regression where the variance of y is constant
- ▶ Common types of generalized linear models include
 - Poisson regression
 - Logistic regression
- ▶ **Logistic regression** models the probability of some event occurring as a linear function of a set of predictor variables

Logistic Regression

- ▶ The logistic regression is used for binomial regression
- ▶ It predicts the probability of occurrence of an event by fitting data to a logistic curve
- ▶ x represents the exposure to some set of risk factors
- ▶ $f(x)$ represents the probability of a particular outcome, given that set of risk factors.



$$f(x) = \frac{e^x}{1 + e^x}$$

Logistic Regression

- ▶ The variable x is a measure of the total contribution of all the risk factors (independent variables) used in the model and is known as the **logit**

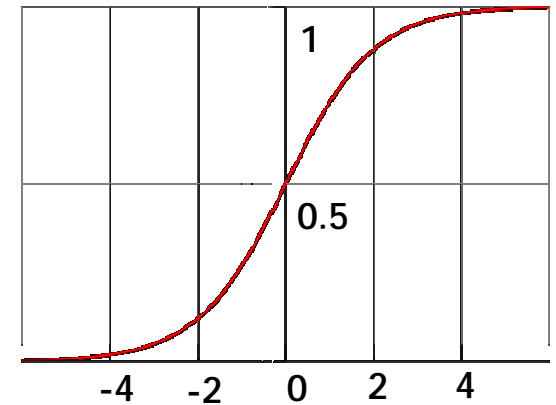
- ▶ x is usually defined as

$$x = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

- ▶ The logistic regression model is given by

$$P(Y = 1 \mid x_1, x_2, \dots, x_k) = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k}}$$

$$f(x) = \frac{e^x}{1 + e^x}$$



Logistic Regression

$$P(Y = 1 | x_1, x_2, \dots, x_k) = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k}}$$

- ▶ Estimate parameters using Maximum Likelihood Estimator
 - **Data:** $y_j, x_{1j}, x_{2j}, \dots, x_{pj}, j=1, 2, \dots, n$
 - Likelihood Function is given by:

$$L(\beta) = \prod_{j=1}^n \frac{e^{\beta_0 + \beta_1 x_{1j} + \dots + \beta_p x_{pj}}}{1 + e^{\beta_0 + \beta_1 x_{1j} + \dots + \beta_p x_{pj}}}$$

- To simplify the computation, we can maximize the log likelihood function
- ▶ To estimate the parameters
 - Compute the partial derivatives of the loglikelihood
 - Equate each partial derivative to zero, and solve the resulting nonlinear equations

Summary of Section 2.7

- ▶ Numeric Prediction is the task of predicting **continuous values**
- ▶ **Regression analysis** is mostly used for prediction
- ▶ Regression can be of different forms **Linear** and **nonlinear**
- ▶ **Logistic regression** is used to model **binomial** regression

Chapter 2: Classification & Prediction

- ▶ 2.1 Basic Concepts of Classification and Prediction
- ▶ 2.2 Decision Tree Induction
- ▶ 2.3 Bayes Classification Methods
- ▶ 2.4 Rule Based Classification
- ▶ 2.5 Note on SVM (Support Vector Machine)
- ▶ 2.6 Lazy Learners
- ▶ 2.7 Prediction
 - 2.7.1 Definitions
 - 2.7.2 Linear Regression
 - 2.7.3 Nonlinear Regression
 - 2.7.4 Generalized Linear Models: Logistic Regression
- ▶ 2.8 How to Evaluate and Improve Classification
 - 2.7.1 Accuracy and Error Measures
 - 2.7.2 Evaluating a Classifier or Predictor
 - 2.7.3 Increasing the Accuracy

2.7.1 Accuracy and Error Measures

Classifier Accuracy Measures

- ▶ Using training data to build and test a classifier can result in a misleading **overoptimistic** estimates
- ▶ Accuracy is better measures using test data that was not used to build the classifier
- ▶ **Accuracy:** $[\text{Acc}(M)]$ - accuracy of model M
 - The percentage of test set tuples that are correctly classified
 - Referred to as the **overall recognition rate** of the classifier
 - **Error rate** or **misclassification rate:** $1-\text{Acc}(M)$
 - When training data are used, the error rate is called **resubstitution error**

Classifier Accuracy Measures

- ▶ The confusion matrix as a table of at least m by m size. An entry $CM_{i,j}$ indicated the number of tuples of class i that were labeled as class j

| Real class \ Predicted class | Class ₁ | Class ₂ | ... | Class _m |
|------------------------------|--------------------|--------------------|-----|--------------------|
| Class ₁ | $CM_{1,1}$ | $CM_{1,2}$ | ... | $CM_{1,m}$ |
| Class ₂ | $CM_{2,1}$ | $CM_{2,2}$ | ... | $CM_{2,m}$ |
| ... | ... | ... | ... | ... |
| Class _m | $CM_{m,1}$ | $CM_{m,2}$ | ... | $CM_{m,m}$ |

- ▶ Ideally, most of the tuples would be represented along the diagonal of the confusion matrix

Classifier Accuracy Measures

Case of binary classification

- ▶ **Positive tuples:** tuples of the main class of interest (e.g., C_1)
- ▶ **Negative tuples:** tuples of the other class (e.g, C_2)

| Real class \ Predicted class | C_1 | C_2 |
|------------------------------|----------------|----------------|
| C_1 | True positive | False negative |
| C_2 | False positive | True negative |

- ▶ **True positives:** positive tuples correctly labeled
- ▶ **True negatives:** negative tuples correctly labeled
- ▶ **False positives:** negative tuples incorrectly labeled
- ▶ **False negatives:** positive tuples incorrectly labeled

Classifier Accuracy Measures

- ▶ Other measures can be used when the accuracy measure is not acceptable

- Sensitivity
- Specificity
- Precision

$$sens = \frac{t_pos}{pos}$$

$$spec = \frac{t_neg}{neg}$$

$$precision = \frac{t_pos}{(t_pos + f_pos)}$$

$$accuracy = sens \frac{pos}{(pos + neg)} + spec \frac{neg}{(pos + neg)}$$

- **t_pos**: the number of true positives
- **t_neg**: the number of true negatives
- **Neg**: number of positive tuples
- **Pos**: number of positive tuples
- **F_pos**: number of false positives

Predictor Error Measures

- ▶ The predictor returns continuous values
 - It is difficult to say whether the predicted value is correct or not
 - Measure how far the predicted value from the known value
- ▶ Compute **loss functions**

$$\textit{Absolute error} = |y_i - y_i'|$$

$$\textit{Squared error} = (y_i - y_i')^2$$

→ Squared error is more sensitive to outliers

- ▶ The **test error** or **generalization error** is the average loss

$$\textit{Mean absolute error} = \frac{\sum_{i=1}^{|D|} |y_i - y_i'|}{|D|}$$

$$\textit{Mean squared error} = \frac{\sum_{i=1}^{|D|} (y_i - y_i')^2}{|D|}$$

Predictor Error Measures

- ▶ The total loss can be normalized by dividing by the total loss incurred from always predicting the mean

$$\textit{Relative absolute error} = \frac{\sum_{i=1}^{|D|} |y_i - y_i'|}{\sum_{i=1}^{|D|} |y_i - \bar{y}|}$$

$$\textit{Relative squared error} = \frac{\sum_{i=1}^{|D|} (y_i - y_i')^2}{\sum_{i=1}^{|D|} (y_i - \bar{y})^2}$$

- ▶ In practice, the choice of error measure does not greatly affect prediction model selection

2.7.2 Evaluating a Classifier or Predictor

- ▶ How can we use the measures described previously to obtain a reliable estimate of classifier accuracy (or predictor accuracy in terms of error)?
- ▶ Some common techniques used for this purpose are
 - Holdout Method and Random Subsampling
 - Cross-validation
 - Bootstrap
- ▶ They assess accuracy based on randomly sampled partitions of the given data
- ▶ These techniques increase the overall computation time

Holdout and Random Subsampling

▶ Holdout

- Randomly partition the data into two independent sets: **training set** and **test set**
- Typically: **two-thirds** of the data are allocated to training set and **one-third** is allocated to test set
- The estimate is **pessimistic** because only a portion of the initial data is used to derive the model

▶ Random Subsampling

- The **holdout** method is repeated **k times**
- The overall accuracy is taken as the **average** of the accuracies obtained from each iteration

Cross-validation

- ▶ Partition the data into k mutually exclusive subsets or “folds”, D_1, D_2, \dots, D_k
- ▶ Training and testing is performed k times
 - First iteration: use D_2, \dots, D_k as training and D_1 as test
 - Second iteration: use D_1, D_3, \dots, D_k as training and D_2 as test
 - ...
- ▶ Each sample is used the same number of times for training and once for testing

Cross-validation

▶ Leave-one-out

- A **special case** of k-fold cross-validation
- K is set to the initial number of tuples
- Only **one** sample is **left out** at a time for the test set

▶ Stratified cross-validation

- The class distribution of the tuples in each fold is approximately the same as in the initial data
- ▶ In general, stratified 10-fold cross validation is recommended for estimating accuracy due to its relatively low bias and variance

Bootstrap

- ▶ **Sample** training tuples **uniformly with replacement**
 - i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set
- ▶ Several bootstrap methods, and a common one is **.632 bootstrap**
 - Suppose we are given a data set of **d tuples**
 - The data set is **sampled d times** with replacement
 - Result: a **training** set of **d samples**
 - About **63.2%** of the original data will end up in the bootstrap, and the remaining **36.8%** will form the test set (since $(1 - 1/d)^d \approx e^{-1} = 0.368$)
- ▶ Repeat the sampling procedure k times, overall accuracy of the model:

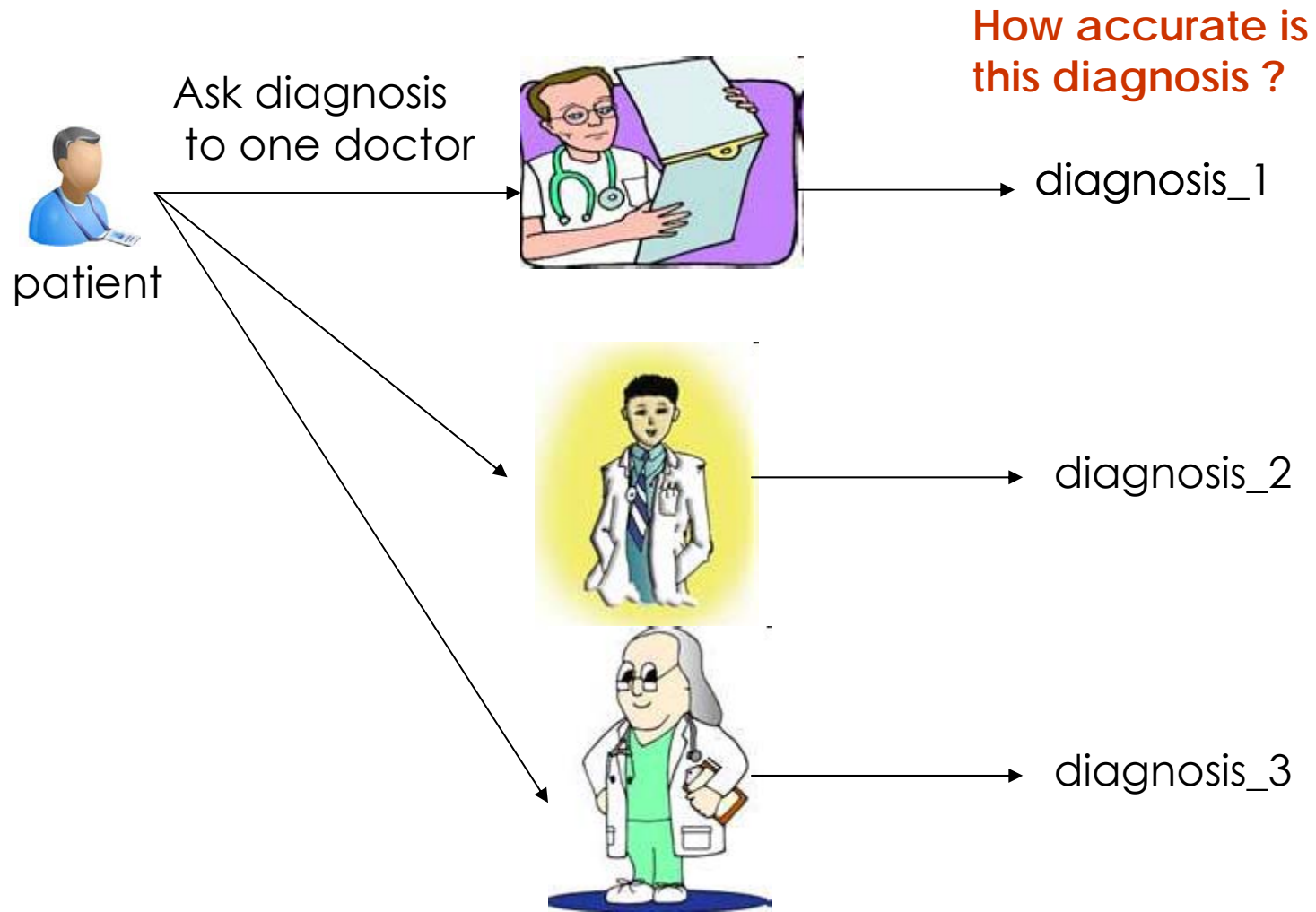
$$acc(M) = \sum_{i=1}^k (0.632 \times acc(M_i)_{test_set} + 0.368 \times acc(M_i)_{train_set})$$

2.7.3 Increasing the Accuracy

- ▶ We have seen that pruning improves the accuracy of decision trees by reducing the overfitting effect
- ▶ There are some general strategies for improving the accuracy of classifiers and predictors
- ▶ **Bagging** and **Boosting** are some of these strategies
 - **Ensemble methods**: use a combination of models
 - Combine a series of learned classifiers M_1, M_2, \dots, M_k
 - Find an improved **composite model** M^*

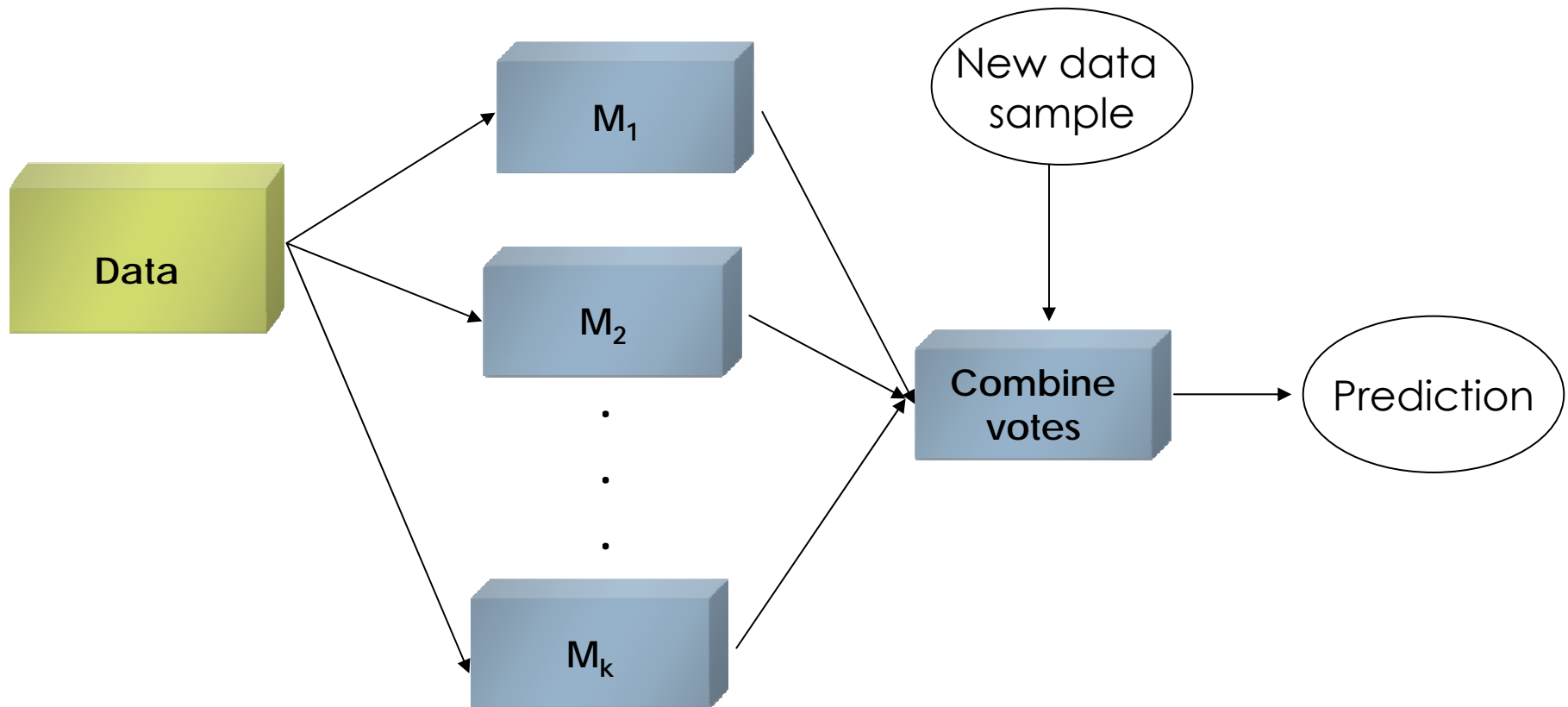
Bagging

Intuition



Choose the diagnosis that occurs more than any of the others

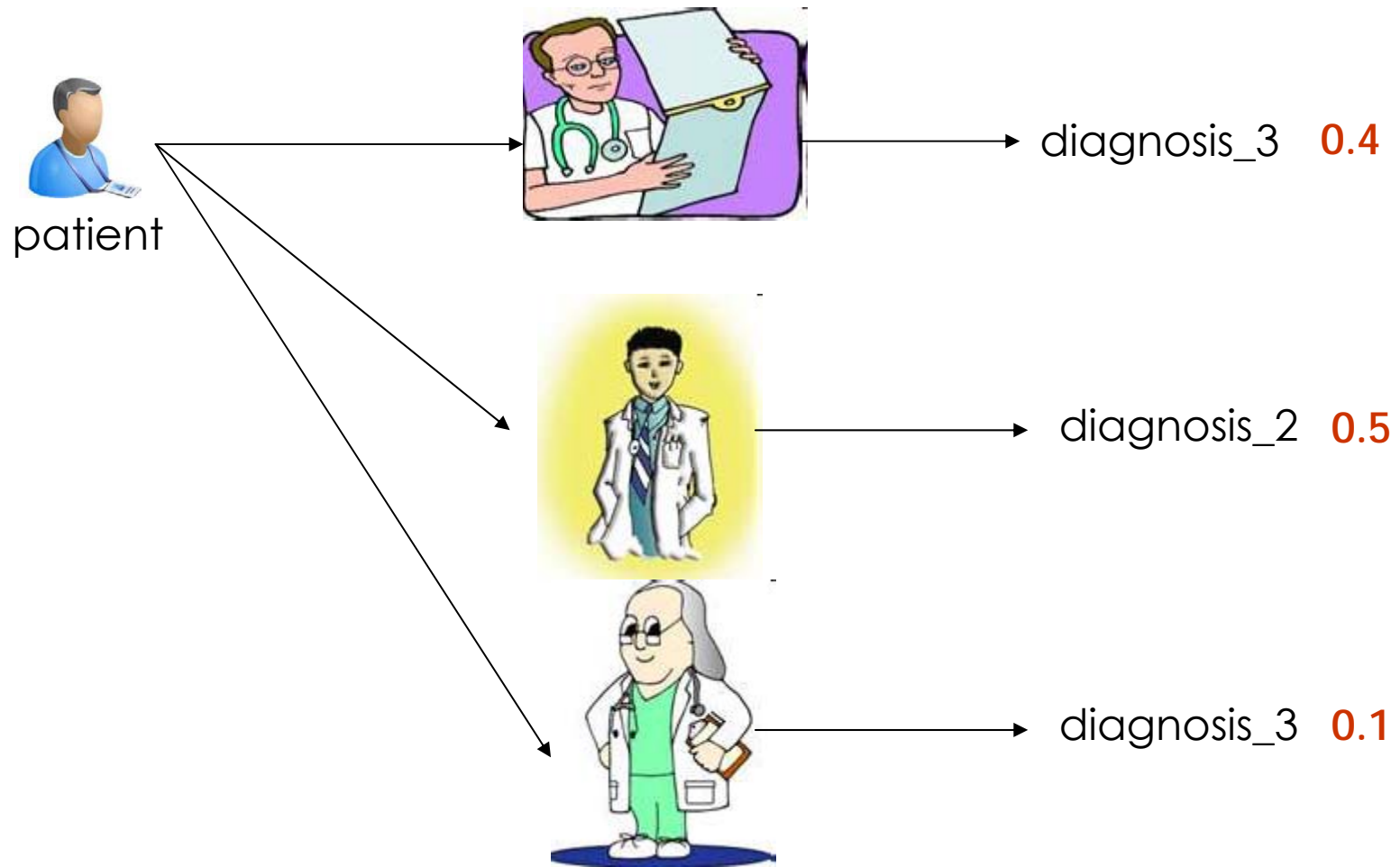
Bagging



- ▶ K iterations
- ▶ At each iteration a training set D_i is sampled with replacement
- ▶ The combined model M^* returns the most frequent class in case of classification, and the average value in case of prediction

Boosting

Intuition



Assign different weights to the doctors based on the accuracy of their previous diagnosis

Boosting

- ▶ **Weights** are assigned to **each** training **tuple**
- ▶ A series of k classifiers is iteratively learned
- ▶ After a classifier M_i is learned, the **weights are adjusted** to allow the subsequent classifier to **pay more attention** to training tuples misclassified by M_i
- ▶ The final boosted classifier M^* combines the votes of each individual classifier where the weight of each classifier is a function of its accuracy
- ▶ This strategy can be extended for the prediction of continuous values

Example: Adaboost Algorithm

- ▶ Given a set of d class-labeled tuples $(X_1, y_1), \dots, (X_d, y_d)$
- ▶ Initially, all the weights of tuples are the same: $1/d$
- ▶ Generate k classifiers in k rounds.
- ▶ At round i , tuples from D are sampled (with replacement) to form a training set D_i of the same size
- ▶ Each tuple's **chance of being selected** depends on its **weight**
- ▶ A classification model M_i is **derived** and **tested** using D_i
- ▶ If a tuple is **misclassified**, its **weight increases**, otherwise it decreases (use $\text{err}(M_i)/(1-\text{err}(M_i))$)

Example: Adaboost Algorithm

- ▶ Error rate $err(X_i)$ is the misclassification error of tuple X_i
- ▶ Classifier M_i error rate is the sum of the weights of the misclassified tuples

$$error(M_i) = \sum_j^d w_j \times err(\mathbf{X}_j)$$

- Tuple correctly classified: $err(X_i)=0$
 - Tuple incorrectly classified: $err(X_i)=1$
- ▶ The weight of classifier M_i 's vote is

$$\log \frac{1 - error(M_i)}{error(M_i)}$$

Summary of Section 2.8

- ▶ **Accuracy** is used to assess **classifiers**
- ▶ **Error measures** are used to assess **predictors**
- ▶ **Stratified 10-fold cross validation** is recommended for estimating accuracy
- ▶ **Bagging** and **boosting** are used to improve the the accuracy of classifiers and predictors