CSC-411 Artificial Intelligence

Introduction to Machine Learning

Classification

Bayesian Classification

- A statistical classifier: performs probabilistic prediction, i.e., predicts class membership probabilities

- Foundation: Based on Bayes’ Theorem.

- Performance: A simple Bayesian classifier, Naïve Bayes classifier, has comparable performance with decision tree and selected neural network classifiers.

- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data.

- Standard: Can provide a standard of optimal decision making against which other methods can be measured

Bayesian Theorem

- Given training data X, posteriori probability of a hypothesis H, P(H|X), follows the Bayes theorem

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

- Informally, this can be written as

posteriori = likelihood x prior/evidence

- Predicts X belongs to C2 iff the probability P(Ci|X) is the highest among all the P(Ck|X) for all the k classes
- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Naïve Bayesian Classifier: Training Dataset

Class:
C1:buys_computer = ‘yes’
C2:buys_computer = ‘no’

Data sample
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

age	income	student	redit_ratin	ys_compu
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayesian Classifier: An Example

- P(Ci): P(buys_computer = “yes”) = 9/14 = 0.643

  P(buys_computer = “no”) = 5/14= 0.357

- Compute P(X|Ci) for each class

  P(age = “<=30” | buys_computer = “yes”) = 2/9 = 0.222

  P(age = “<= 30” | buys_computer = “no”) = 3/5 = 0.6

  P(incom

  P(incom

  P(student = “yes” | buys_computer = “yes) = 6/9 = 0.667

  P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2

  P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = 0.667

  P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4

- X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

P(X|Ci) : P(X|buys_computer = “yes”) = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X|buys_computer = “no”) = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

P(X|Ci)*P(Ci) : P(X|buys_computer = “yes”) * P(buys_computer = “yes”) = 0.028

P(X|buys_computer = “no”) * P(buys_computer = “no”) = 0.007

Therefore, X belongs to class (“buys_computer = yes”)

Naïve Bayesian Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.
    - Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
  - Bayesian Belief Networks

Linear Regression

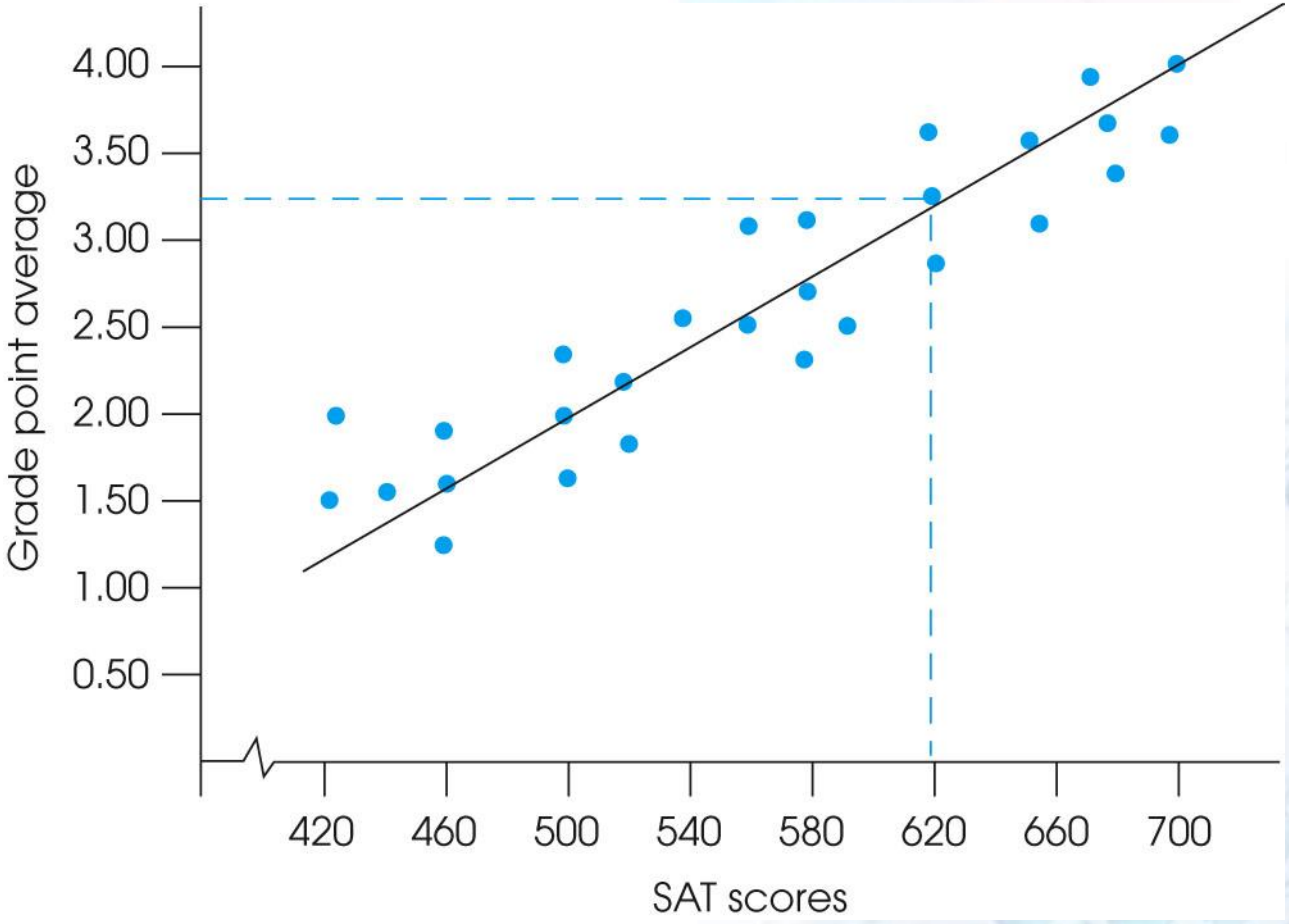
- Regression is a statistical procedure that determines the equation for the straight line that best fits a specific set of data.

- Any straight line can be represented by an equation of the form $Y = bX + a$, where b and a are constants.

- The value of b is called the slope constant and determines the direction and degree to which the line is tilted.

- The value of a is called the Y-intercept and determines the point where the line crosses the Y-axis.

Linear Regression



Linear Regression (cont.)

- How well a set of data points fits a straight line can be measured by calculating the distance between the data points and the line.

- The total error between the data points and the line is obtained by squaring each distance and then summing the squared values.

- The regression equation is designed to produce the minimum sum of squared errors.

Linear Regression (cont.)

$$x = w_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k$$

- Calculate weights from training data

- Predicted value for first training instance a(1)

$$w_0 a_0^{(1)} + w_1 a_1^{(1)} + w_2 a_2^{(1)} + \ldots + w_k a_k^{(1)} = \sum_{j=0}^{k} w_j a_j^{(1)}$$

Classification by regression

- Regression scheme be used for classification

- Two-class problem
  - Training: call the classes 0 and 1
  - Prediction: set a threshold for predicting class 0 or 1

- Multi-class problem: “multi-response linear regression”
  - Training: perform a regression for each class
    - Set output to 1 for training instances that belong to the class, 0 for instances that don’t
  - Prediction: choose the class with the largest output… or use “pairwise linear regression”, which performs a regression for every pair of classes