Lesson 12

Classification, Supervised Learning, and KNN and SGD Classifiers

Classification

- An exploratory data-mining method, which creates groups of objects of similar types or characteristics
- Refers to learning from existing categorizations and forming the groups of objects which are showing similar characteristics

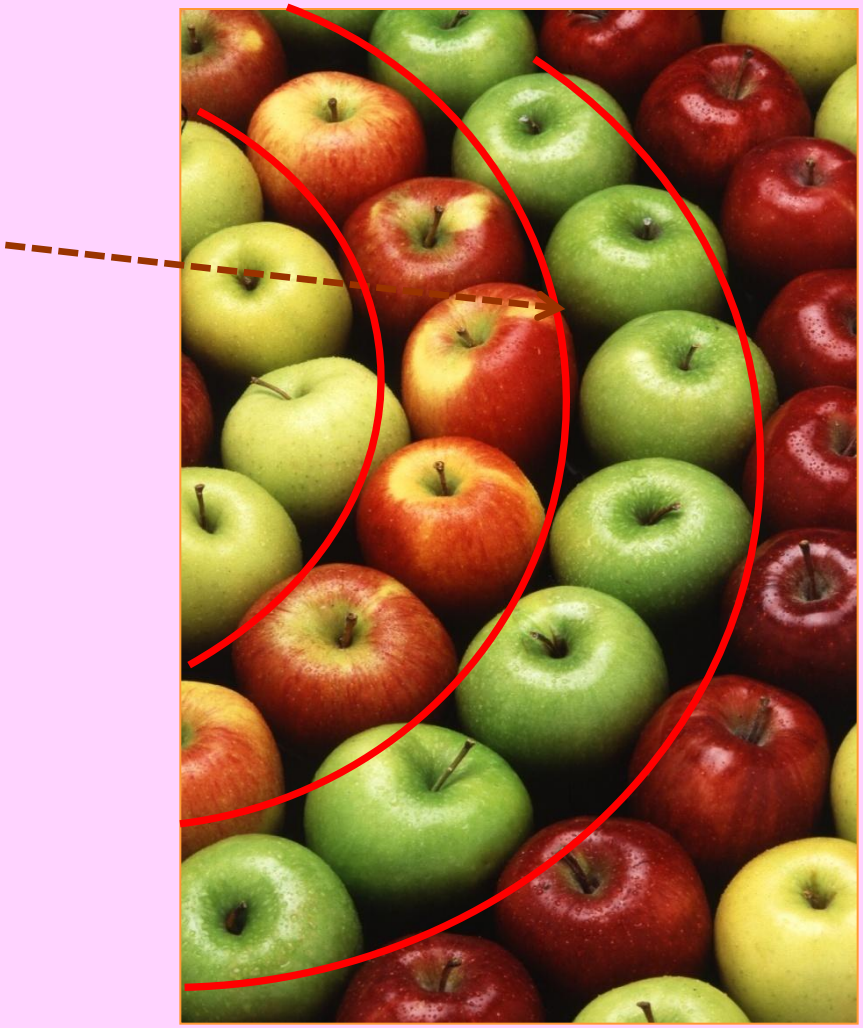
Classification

- A supervised learning method
- A machine learning algorithm which decides usage of the experience, and emulates certain human decisions
- For example, categorize students who are good in theory and practical subjects both as ‘very good’

Difference with respect to Classification

- Clustering finds only the similar objects
- Classification differs from clustering in the sense that classification assigns a class to each distinct set of characteristics in the collection
- For example, apples of different colours in the figure classified as of distinct variety

Four Classes of Apples



Applications

- Pattern recognition,
- image analysis,
- information retrieval
- bioinformatics

Supervised Learning

- Refers to a process in which an ML algorithm use known outputs and expected target variables for the selected inputs as training datasets and takes decisions or makea predictions for new inputs

Input Vectors for Classification

- Set of Input Column vectors $\mathcal{I}$ of datapoints consists of elements in the metric and non-metric space
- Set of Input predictor variables **P** along with the target output vectors **ET** used as input to a Classifier algorithm
- Training dataset examples $\mathcal{E}$ consists of both **P** and **ET**

Classifier

- Needs training, which means learning from existing categorizations and then forming groups of objects showing similar characteristics.

Training

- A learning process which uses training dataset $\mathcal{I}$ and generates a model program, M

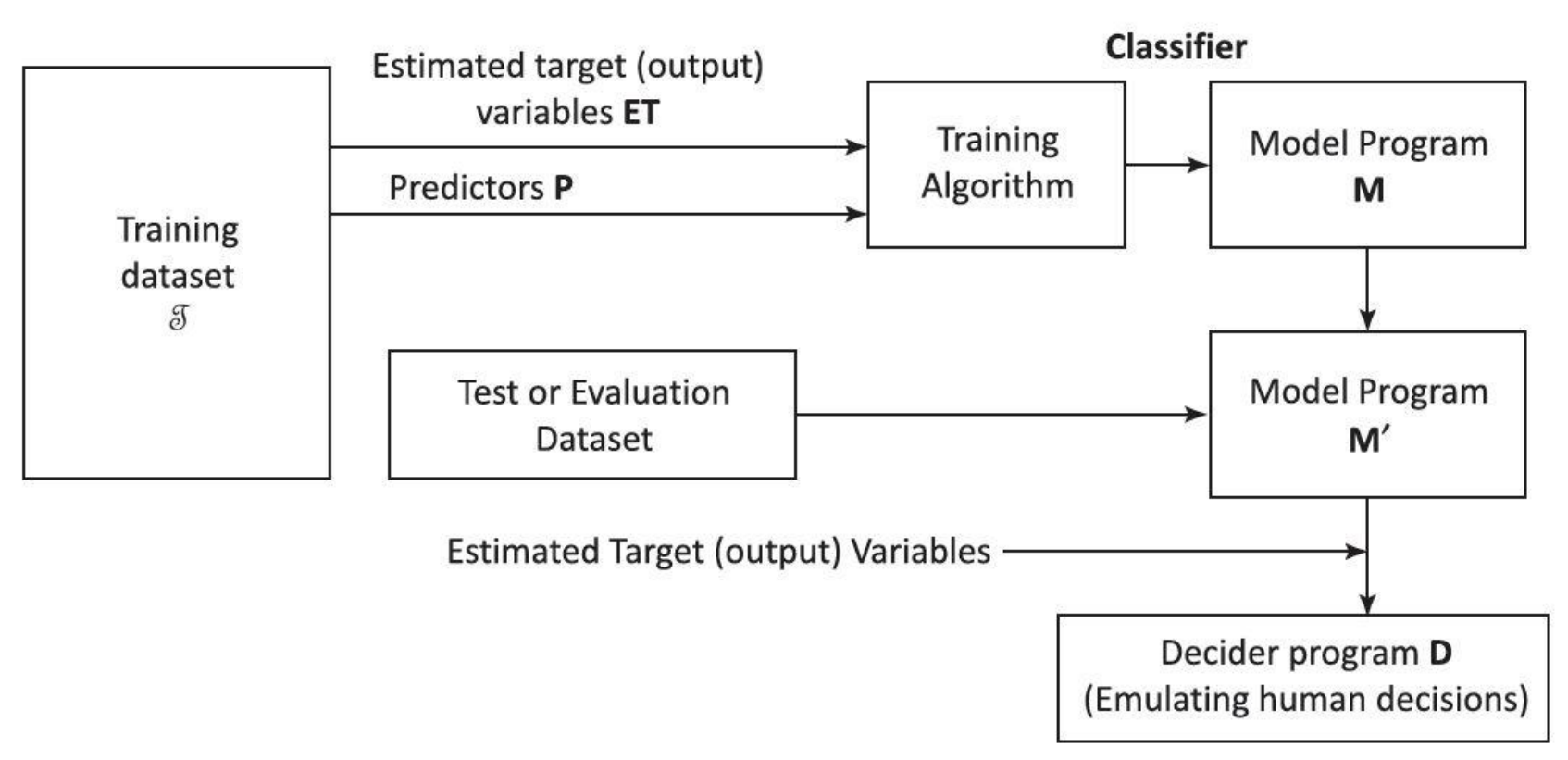
Training Dataset $\mathcal{T}$

- Means a subset $\mathcal{E}$ of an exemplary dataset which includes training variables
- $\mathcal{T}$ includes value of the target variables and predictors also

Training Algorithm

- Generates a ‘Model’, M
- M is a program which gives the output vectors for taking the decision of the class to which the input vector belongs
- Remember, a set of datapoints can be represented by a set of vectors in v-dimensional space.

Figure 6.15 Steps during the learning phase of a Classifier



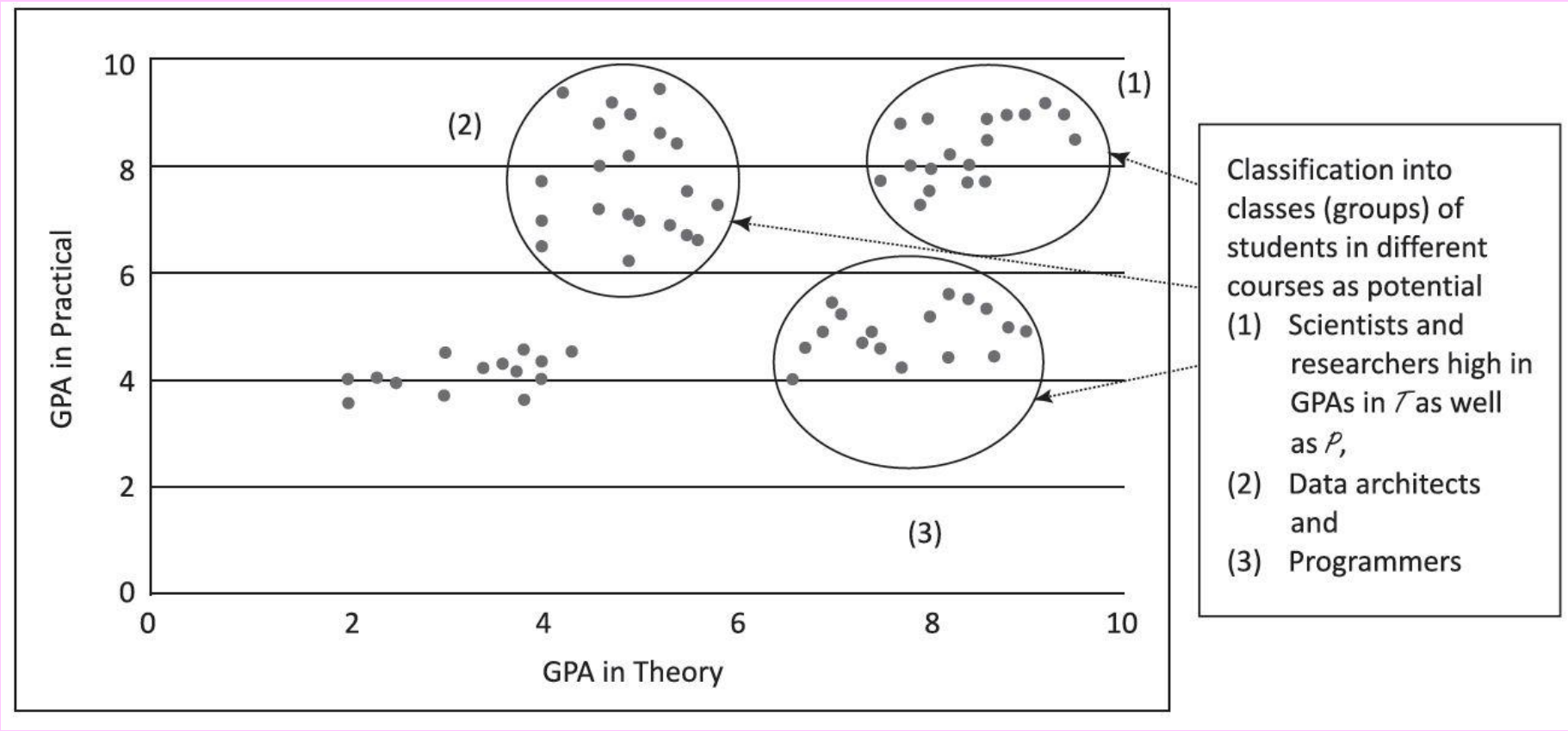
Steps during the learning phase of a classifier

- Training algorithm needs (i) training dataset T, which includes predictor variables **P**, as well as target variables [outputs ET (estimated target variables)] both as the inputs
- Generates a model M from inputs, **P** and **ET**

Steps during the learning phase of a classifier

- M internally used test or evaluation datasets
- Inputs to a copy M’ are Predictor Variable Only
- ET and M’ are Inputs to Decider Program **D**

Figure 6.16 Classification on the basis of performances of the student groups



K-NN Application Areas

- Regression
- Similar items search (k is 1 for nearest neighbour, 2 for next to nearest and 3 for next to next nearest)
- Classification

K-NN Classifierr

- Training dataset consists of k-closest examples in feature space
- Feature space means, space with categorization variables (non-metric variables)

K-NN Learning

- Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

K-NN Object Classification Criterion

- An object classification criteria is majority vote
- The lazy algorithm, learns and start giving accurate results after thorough learning

Stochastic Gradient Descent (SGD) Algorithms

- such as logistic regression are sequential, incremental efficient (fast) and used when computational needs are of small (< 0.1 M) to medium (< 10 M) dataset.
- Predictor variables can be metric and no-metric, any of the four types.

Stochastic Variable, Process or system

- Means a variable, function, process or system connected with random probability, chance or randomness
- For example, the experimental observations exhibiting stochastic deviations with respected to the expected values from regression line.

Stochastic Variable, Process or system

- Assume variables x_1, x_2, …, x_v in v-dimensional space.
- For 3-dimension metric space, x1 is x, x2 is y and x3 is z.

Objective Function

- Assume that objective is to find the coefficients, parameters or weights in a function for which the error, deviation or variance are minimum, or to find those which best classify the output data-points, responses or observations when in n observations.

Objective Function

- Assume n observations are for a dependent variable, and they are the function Q of (c_0, c1) in case of straight line, (c_0, c1, c2, .., m) in case of polynomial, or (λ, $\bar{x}$) in Kernel function [Equations (6.6a to c)]
- Let Q is sum over n values of Q_i from i = 1 to n: $Q(\mathbf{c}) = \sum Q_i(\mathbf{c})$

Objective Function

- Objective function $Q(c_1, c_2, ..., c_m) = \sum Q_i(c_0, c_1, c_2, .., c_m)$ for optimizing for best values of c_0, c1, c2, .., m from i = 1 to n observations.

- $$Q(\mathbf{c}) = \sum_{i=1}^{n} Q_i(\mathbf{c})$$

Examples of Objective Function for minimizing, converging or optimizing

- Minimize chi-square = $\chi^2 = e_i{}^2 = \sum (y_i - y'_i)^2$ [Refer Ch.06 L06 PPTs.] for best fitting lowest deviation in values in observation for the $\underline{\mathbf{c}}$ (coefficients in regression) or (λ, $\overline{x}$) Kernel functions)

Examples of Objective Function for minimizing, converging or optimizing

- Least square function is an example of objective function. For straight line, $e_i^2 = \sum\{(y_j - (c0 + c_1.x_j)\}^2$

Stochastic Gradient

- Gradient equals to change in a function $Q(c_1, c_2, \ldots c_v)$ value with respect to a very small change in a parameter (coefficient or weight) value, say $c_1, c_2, \ldots c_v$ (Partial Derivatives)
- Objective Function is a summation, to be optimized (minimized) for $c_1, c_2, .., cv)$ for values $x_1, x_2, \ldots x_n$

Stochastic Gradient Descent Method

- For example, in a function Q(x1, x2, ..., xv), gradient of Q with respect to xi equals partial differentiations, $\partial Q/\partial ci$, where i = 1, 2, ..., v.
- $Y= c0+ c1.x_1 + \ldots.cv.x_v$, where c are the coefficients (weights of x1, ,2, x3, …) for computing Y

Stochastic Gradient Descent Method

- For example, in a function Q(x1, x2, ..., xv), gradient of Q with respect to xi equals partial differentiations, $\partial Q/\partial ci$, where i = 1, 2, ..., v.
- Y= c0+ c1.x1 + ….cv.xv, where c are the coefficients (weights of x1, ,2, x3, …) for computing Y
- Gradient approach 0 at maxima or minima

Objective Function

- **Q** (**c**) = ΣQ_i (c) (Sum is for i = 0, 1, 2, ...v) in v-dimensional space
- **Q**(**c**) = $c_0 + c_1.x_1 + \ldots . c_v.x_v$
- For = 1 to n $\{c_i = c_i - \alpha \nabla Q_i$ (**c**)}, where α is learning rate (rate with which the **Q**(**c**) approaches towards minimum
- Gradient ∇Q_i (c) approach 0 at maxima or minima

Least Square Fit

- Recall Section 6.3.3. It explained how the best fit could be reached by using the ‘least squares criterion’,

SGD classifier

- Trains and learns the computation of objective function parameter values and classifies based on predictor values for each class
- .

Quadratic Kernel Function

- Square of the Linear ($\mathbf{x}^T.\mathbf{y}$)
- Dot Denotes Dot Product of Vectors
- [T denotes the transpose of a column vector]
- Transpose of matrix **A** the *i*-th row, *j*-th column element of $\mathbf{A}^T$ is the *j*-th row, *i*-th column element of matrix **A**

Quadratic Kernel Functions

- $K(x, y)=(x^T * y+c)^2$
- $K(x_i, x_j) = \text{Sqrt}\ (||x_i - x_j||^2 + c^2)$ [Multi-Quadric]
- $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T.\mathbf{y})^2$ or $(1 + \mathbf{x}^T.\mathbf{y})^2$
- Refer [https://www.cs.utah.edu/~piyush/teaching/15-9-print.pdf for further reading.]

Quadratic Kernel Function

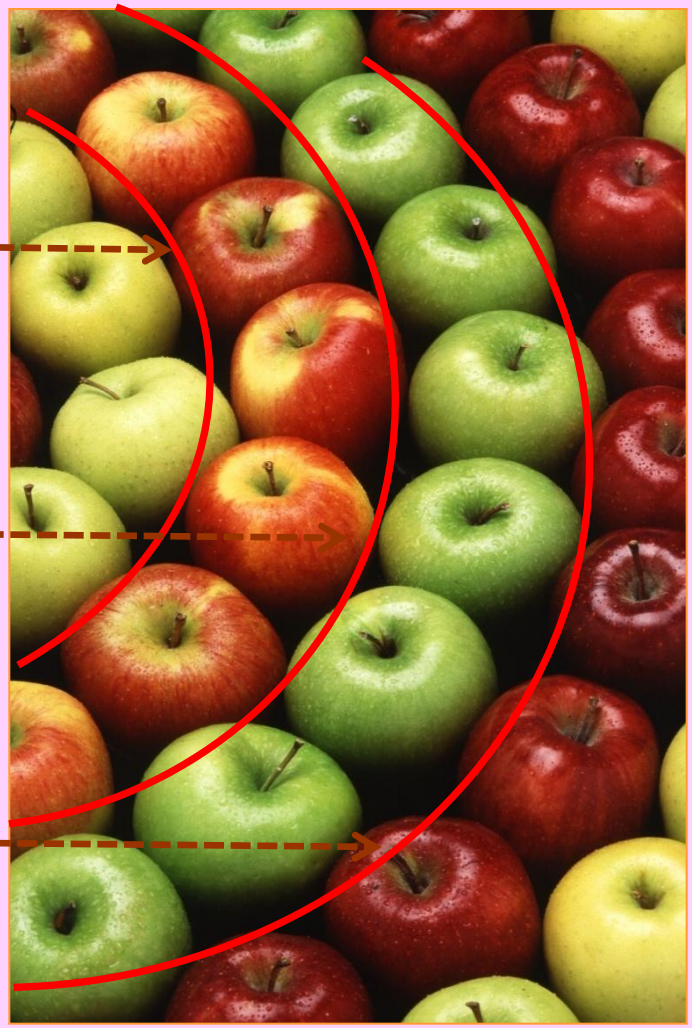
- For example, circle, parabola, ellipse, ellipsoid equations
- Objective functions and classifier vectors generated from the quadratic function Applications: Support Vectors

Three Quadratic Kernel Functions for classifying

Quadratic function K1

Quadratic function K2

Quadratic function K3



Four Classes of Apples using three objective functions based on K

Logistic Regression Trained via SGD

- Uses predictor variables and linear weights, they pass through a soft-limit function that limits output between 0 and 1;
- Uses hash values for the features, which means training algorithm assigns each feature a hash value, which is used for indexing, search and predictor variable.

Summary

We learnt:

- Classification as process of assigning a class to each distinct set of characteristics in the collection
- Training datasets, Predictor Vector, Expected Target Vectors
- Model Program M and Decider Program

Summary

We learnt:

- K-Means and K-Medoids
- Stochastic Gradient Descent Method
- Objective Functions for n observations
- Use of Regression and Quadratic Functions
- Logistic Regression

End of Lesson 12 on Classification, Supervised learning, and KNN and SGD Classifiers