Machine Learning and MLBase

Amir H. Payberah
Swedish Institute of Computer Science

amir@sics.se
May 20, 2014

What is the Problem?

Data

What is the Problem?

Data → Actionable Knowledge

What is the Problem?

Data → Actionable Knowledge

That is roughly the problem that Machine Learning addresses!

Data and Knowledge

Data (blue) ⟶ Knowledge (green)

- Is this email spam or no spam?

Data and Knowledge

Data (blue) ⟶ Knowledge (green)

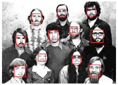
- Is this email spam or no spam?

Data and Knowledge

Data (blue) ⟶ Knowledge (green)

- Is this email spam or no spam?

- Is there a face in this picture?

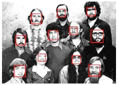


Data and Knowledge

Data (blue) → Knowledge (green)

- Is this email spam or no spam?

- Is there a face in this picture?

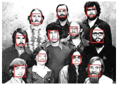


Data and Knowledge

Data (blue) ⟶ Knowledge (green)

- Is this email spam or no spam?

- Is there a face in this picture?

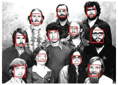


- Should I lend money to this customer given his spending behaviour?

Data (blue) → Knowledge (green)

- Is this email spam or no spam?

- Is there a face in this picture?



- Should I lend money to this customer given his spending behaviour?

Data and Knowledge

- Knowledge is not concrete

- Spam is an abstraction
- Face is an abstraction
- Who to lend to is an abstraction

You do not find spam, faces, and financial advice in datasets, you just find bits!

Abstraction

we have data

but, we want abstractions!

What is an Abstraction?

- Anything whose description does not depend exclusively on the bits you have.
- Abstraction always involves assumptions.

Machine Learning

- Machine learning is the science of automating the process of abstraction from raw data and assumptions.

- Data: painted image + dataset of normal images.

- Data: painted image + dataset of normal images.

+

- Assumption: the non-painted parts of the painted image behave as the images in the dataset.

- Data: painted image + dataset of normal images.

+

- Assumption: the non-painted parts of the painted image behave as the images in the dataset.

- Abstraction: correct image.

More Precise Definition

Arthur Samuel (1959)

Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

More Precise Definition

Arthur Samuel (1959)

Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

Tom Mitchell (1998)

Well-posed Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Experience (E), Task (T), and Performance (P)

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.

Experience (E), Task (T), and Performance (P)

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.

- Classifying emails as spam or not spam.

Experience (E), Task (T), and Performance (P)

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.

- Classifying emails as spam or not spam. T

Experience (E), Task (T), and Performance (P)

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.

- Classifying emails as spam or not spam. T

- Watching you label emails as spam or not spam.

Experience (E), Task (T), and Performance (P)

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.

- Classifying emails as spam or not spam. T

- Watching you label emails as spam or not spam. E

Experience (E), Task (T), and Performance (P)

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.

- Classifying emails as spam or not spam. T
- Watching you label emails as spam or not spam. E
- The fraction of emails correctly classified as spam/not spam.

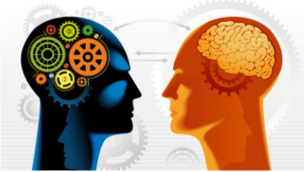
Experience (E), Task (T), and Performance (P)

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.

- Classifying emails as spam or not spam. T
- Watching you label emails as spam or not spam. E
- The fraction of emails correctly classified as spam/not spam. P

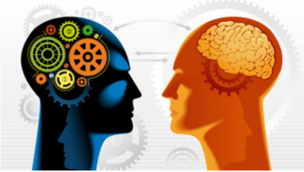
Types of Learning

- Supervised learning
- Unsupervised learning



Types of Learning

- Supervised learning
- Unsupervised learning



Supervised Learning

- Right answers are given.
  - Training data (input data) is labeled, e.g., spam/not-spam or a stock price at a time.

Supervised Learning

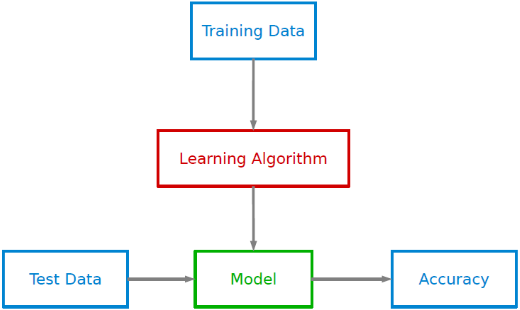
- Right answers are given.
  - Training data (input data) is labeled, e.g., spam/not-spam or a stock price at a time.
- A model is prepared through a training process.
  - The model is required to make predictions.
  - The model is corrected when those predictions are wrong.

Supervised Learning

- Right answers are given.
  - Training data (input data) is labeled, e.g., spam/not-spam or a stock price at a time.
- A model is prepared through a training process.
  - The model is required to make predictions.
  - The model is corrected when those predictions are wrong.
- The training process continues until the model achieves a desired level of accuracy on the training data.

Supervised Learning

- Training phase
- Testing phase

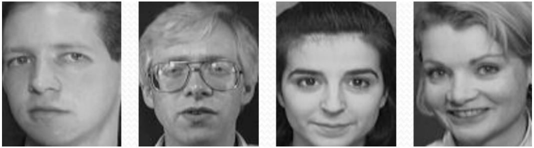


Supervised Learning: Example

- Face recognition

Training data

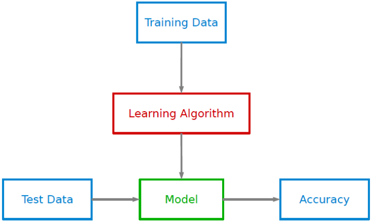
Testing data



[ORL dataset, AT&T Laboratories, Cambridge UK]

Supervised Learning - More Formal Definition (1/2)

- Set of N training examples: $(x_1, y_1), \cdots, (x_n, y_n)$.
- $x_i =< x_{i1}, x_{i2}, \cdots, x_{im} >$ is the feature vector of the *i*th example.
- y_i is the *i*th feature vector label.
- A learning algorithm seeks a function $g : X \rightarrow Y$.



Supervised Learning - More Formal Definition (2/2)

- Sometimes it is convenient to represent g using a scoring function $f : X \times Y \rightarrow \mathbb{R}$.

- Then, g is defined as returning the y value that gives the highest score: $g(x) = \arg\max_{y} f(x, y)$.

Supervised Learning Algorithms

- Classification: the output variable takes discrete values.

- Regression: the output variable takes continuous values.

Classification or Regression?

- You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.

Classification or Regression?

- You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.
  Regression

Classification or Regression?

- You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.
  Regression

- You'd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.

Classification or Regression?

- You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.
  Regression

- You'd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.
  Classification

Supervised Learning Algorithms

- Classification: the output variable takes discrete values.
- Regression: the output variable takes continuous values.

Supervised Learning Algorithms

- Classification: the output variable takes discrete values.
- Regression: the output variable takes continuous values.

Supervised Learning - Classification Algorithms

- *k*-Nearest Neighbours (kNN)
- Decision trees
- Naive Bayes
- Logistic regression
- Support Vector Machine (SVM)
- ...

Supervised Learning - Classification Algorithms

- *k*-Nearest Neighbours (kNN)
- Decision trees
- Naive Bayes
- Logistic regression
- Support Vector Machine (SVM)
- ...

Classification Algorithms - *k*-Nearest Neighbours (1/2)

- We have a set of example labeled data: $(\mathtt{x}_1, \mathtt{y}_1), \cdots, (\mathtt{x}_n, \mathtt{y}_n)$.

Classification Algorithms - *k*-Nearest Neighbours (1/2)

- We have a set of example labeled data: $(x_1, y_1), \cdots, (x_n, y_n)$.
- For data without a label, we compare it to the existing data.

Classification Algorithms - *k*-Nearest Neighbours (1/2)

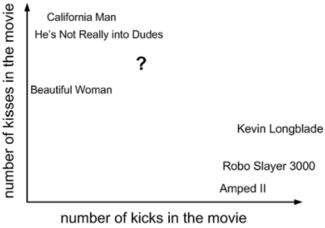
- We have a set of example labeled data: $(x_1, y_1), \cdots, (x_n, y_n)$.
- For data without a label, we compare it to the existing data.
- We then take the most similar pieces of data (the nearest neighbors) and look at their labels.

Classification Algorithms - *k*-Nearest Neighbours (1/2)

- We have a set of example labeled data: $(x_1, y_1), \cdots, (x_n, y_n)$.
- For data without a label, we compare it to the existing data.
- We then take the most similar pieces of data (the nearest neighbors) and look at their labels.
- We look at the top k most similar pieces of data from our known dataset.

Classification Algorithms - *k*-Nearest Neighbours (2/2)

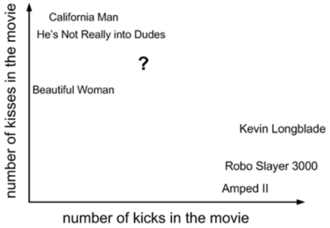
- Classifying movies into genres



Movie title	# of kicks	# of kisses	Type of movie
California Man	3	104	Romance
He's Not Really into Dudes	2	100	Romance
Beautiful Woman	1	81	Romance
Kevin Longblade	101	10	Action
Robo Slayer 3000	99	5	Action
Amped II	98	2	Action
?	18	90	Unknown

Classification Algorithms - *k*-Nearest Neighbours (2/2)

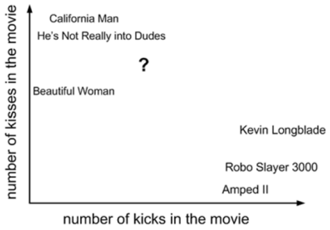
- Classifying movies into genres



Movie title	# of kicks	# of kisses	Type of movie
California Man	3 x_{11}	104 x_{12}	Romance y_1
He's Not Really into Dudes	2 x_{21}	100 x_{22}	Romance y_2
Beautiful Woman	1	81	Romance
Kevin Longblade	101	10	Action
Robo Slayer 3000	99	5	Action
Amped II	98	2	Action
?	18	90	Unknown

Classification Algorithms - *k*-Nearest Neighbours (2/2)

- Classifying movies into genres



Movie title	# of kicks	# of kisses	Type of movie
California Man	3 x_{11}	104 x_{12}	Romance y_1
He's Not Really into Dudes	2 x_{21}	100 x_{22}	Romance y_2
Beautiful Woman	1	81	Romance
Kevin Longblade	101	10	Action
Robo Slayer 3000	99	5	Action
Amped II	98	2	Action
?	18	90	Unknown

Movie title	Distance to movie "?"
California Man	20.5
He's Not Really into Dudes	18.7
Beautiful Woman	19.2
Kevin Longblade	115.3
Robo Slayer 3000	117.4
Amped II	118.9

[Peter Harrignton, "Machine Learning in Action", Manning 2012]

Supervised Learning - Classification Algorithms

- *k*-Nearest Neighbours (kNN)
- Decision trees
- Naive Bayes
- Logistic regression
- Support Vector Machine (SVM)
- ...

Classification Algorithms - Decision Tree (1/3)

- Have you ever played a game called Twenty Questions?

Classification Algorithms - Decision Tree (1/3)

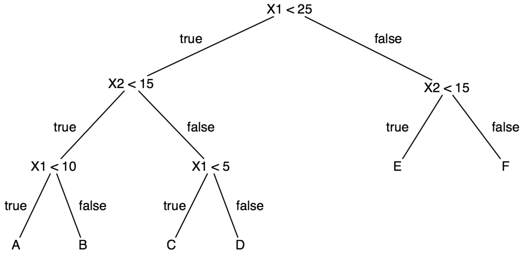
- Have you ever played a game called Twenty Questions?
- One person thinks of some object and players try to guess the object.
- Players are allowed to ask 20 questions and receive only yes or no answers.
- Each question splits the set of objects.

Classification Algorithms - Decision Tree (1/3)

- Have you ever played a game called Twenty Questions?
- One person thinks of some object and players try to guess the object.
- Players are allowed to ask 20 questions and receive only yes or no answers.
- Each question splits the set of objects.
- A decision tree works just like the game Twenty Questions.

Classification Algorithms - Decision Tree (2/3)

- The partitioning idea is used in the decision tree model: split the space recursively according to inputs in x.
- Two main types:
  - Classification tree: the predicted outcome is the class to which the data belongs, e.g., female or male.
  - Regression tree: the predicted outcome can be considered a real number, e.g., the price of a house.



Classification Algorithms - Decision Tree (3/3)

- How to construct the decision tree?

- Top-bottom algorithm:
  - Find the best split condition (quantified based on the impurity measure).
  - Stops when no improvement possible.

- Impurity measure:
  - Measures how well are the two classes separated.
  - Ideally we would like to separate all 0s and 1s.

Supervised Learning - Classification Algorithms

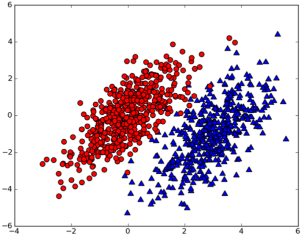
- *k*-Nearest Neighbours (kNN)
- Decision trees
- Naive Bayes
- Logistic regression
- Support Vector Machine (SVM)
- ...

Classification Algorithms - Naive Bayes (1/5)

- Using the probability theory to classify things.
- Naive Bayes is a subset of Bayesian decision theory.

Classification Algorithms - Naive Bayes (1/5)

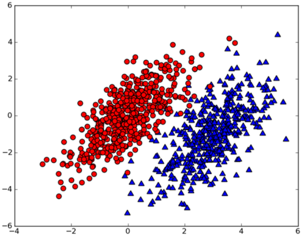
- Using the probability theory to classify things.
- Naive Bayes is a subset of Bayesian decision theory.



- y_1: circles, and y_2: triangles.
- (x_1, x_2) belongs to y_1 or y_2?

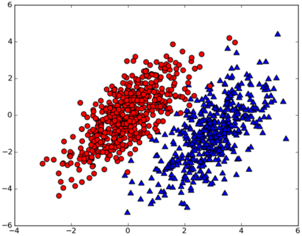
Classification Algorithms - Naive Bayes (2/5)

- (x_1, x_2) belongs to y_1 or y_2?
- If $p(y_1|x_1, x_2) > p(y_2|x_1, x_2)$, the class is y_1.
- If $p(y_1|x_1, x_2) < p(y_2|x_1, x_2)$, the class is y_2.



Classification Algorithms - Naive Bayes (2/5)

- (x_1, x_2) belongs to y_1 or y_2?
- If $p(y_1|x_1, x_2) > p(y_2|x_1, x_2)$, the class is y_1.
- If $p(y_1|x_1, x_2) < p(y_2|x_1, x_2)$, the class is y_2.



- $g(x) = \arg\max_{y} p(y|x)$
- Replace $p(y|x)$ with $\frac{p(x|y)p(y)}{p(x)}$

Classification Algorithms - Naive Bayes (3/5)

- Bayes theorem: $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$
- $p(y|x)$: probability of instance x being in class y.
- $p(x|y)$: probability of generating instance x given class y.
- $p(y)$: probability of occurrence of class y
- $p(x)$: probability of instance x occurring.

Classification Algorithms - Naive Bayes (4/5)

Is officer Drew male or female?

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

Classification Algorithms - Naive Bayes (4/5)

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

- $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$
- $p(\mathtt{male}|\mathtt{drew}) = ?$
- $p(\mathtt{female}|\mathtt{drew}) = ?$

Classification Algorithms - Naive Bayes (4/5)

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

- $\mathtt{p(y|x)} = \frac{\mathtt{p(x|y)p(y)}}{\mathtt{p(x)}}$
- $\mathtt{p(male|drew)} = \frac{\mathtt{p(drew|male)p(male)}}{\mathtt{p(drew)}} = \frac{\frac{1}{3} \times \frac{3}{8}}{\frac{3}{8}} = 0.33$
- $\mathtt{p(female|drew)} = \frac{\mathtt{p(drew|female)p(female)}}{\mathtt{p(drew)}} = \frac{\frac{2}{5} \times \frac{5}{8}}{\frac{3}{8}} = 0.66$

Classification Algorithms - Naive Bayes (4/5)

Officer Drew is female.

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

- $\mathtt{p(y|x) = \frac{p(x|y)p(y)}{p(x)}}$
- $\mathtt{p(male|drew) = \frac{p(drew|male)p(male)}{p(drew)}} = \frac{\frac{1}{3} \times \frac{3}{8}}{\frac{3}{8}} = 0.33$
- $\mathtt{p(female|drew) = \frac{p(drew|female)p(female)}{p(drew)}} = \frac{\frac{2}{5} \times \frac{5}{8}}{\frac{3}{8}} = 0.66$

Classification Algorithms - Naive Bayes (5/5)

- What if we have multiple features?

Name	Over 170cm	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

Classification Algorithms - Naive Bayes (5/5)

- What if we have multiple features?

Name	Over 170CM	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

- To simplify the task, naive Bayesian classifiers assume attributes have independent distributions:

$$p(x|y) = p(x_1|y) \times p(x_2|y) \times \cdots \times p(x_n|y)$$
$$p(\mathtt{drew}|\mathtt{male}) = p(\mathtt{over\_170cm}|\mathtt{male}) \times p(\mathtt{eye} = \mathtt{blue}|\mathtt{male}) \times \cdots$$

Supervised Learning - Classification Algorithms

- *k*-Nearest Neighbours (kNN)
- Decision trees
- Naive Bayes
- Logistic regression
- Support Vector Machine (SVM)
- ...

Classification Algorithms - Logistic Regression (1/4)

- $g(x) = \arg\max_{y} p(y|x)$.

Classification Algorithms - Logistic Regression (1/4)

- $g(x) = \arg\max_{y} p(y|x)$.
- Estimate $p(y|x)$ directly: logistic regression.

Classification Algorithms - Logistic Regression (2/4)

- Estimate $p(y|x)$ directly.
- Training dataset: $(x_1, y_1), \cdots, (x_n, y_n)$
- x_i is a vector of real-valued features $< x_{i1}, x_{i2}, \cdots, x_{im} >$
- $y_i \in \{0, 1\}$

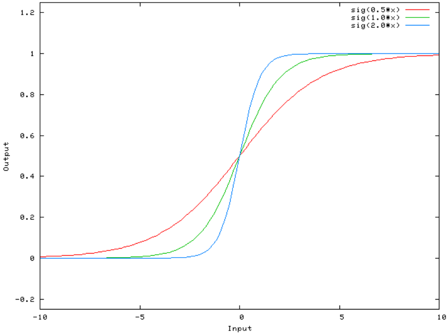
Classification Algorithms - Logistic Regression (2/4)

- Estimate $p(y|x)$ directly.
- Training dataset: $(x_1, y_1), \cdots, (x_n, y_n)$
- x_i is a vector of real-valued features $< x_{i1}, x_{i2}, \cdots, x_{im} >$
- $y_i \in \{0, 1\}$

- We take a linear combination of our input features: $z_i = w_{i0} + \sum_j w_{ij} x_{ij}$
- $h(z) = \frac{1}{1+e^{-z}}$ (sigmoid function)

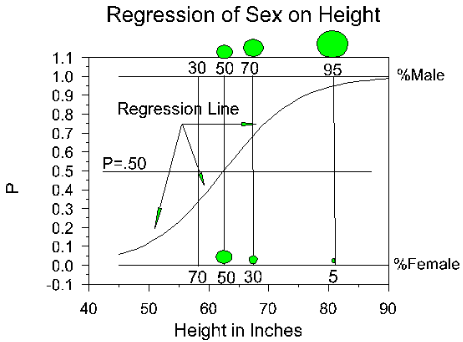
Classification Algorithms - Logistic Regression (3/4)

- $h(z) = \frac{1}{1+e^{-z}}$
- $p(y|x) = h(z)$
- if $h(z) > 0.5$, predict $y = 1$
- if $h(z) < 0.5$, predict $y = 0$



Classification Algorithms - Logistic Regression (4/4)

- Predict whether someone is male or female using height (x).
- $p(y|x) = h(z) = \frac{1}{1+e^{-z}}$
- if $h(z) > 0.5$, predict $y = 1$: male
- if $h(z) < 0.5$, predict $y = 0$: female



Supervised Learning - Classification Algorithms

- *k*-Nearest Neighbours (kNN)
- Decision trees
- Naive Bayes
- Logistic regression
- Support Vector Machine (SVM)
- ...

Classification Algorithms - SVM (1/4)

- SVM: Support Vector Machine
- Forget about probabilities

Classification Algorithms - SVM (1/4)

- SVM: Support Vector Machine
- Forget about probabilities
- Training dataset: $(x_1, y_1), \cdots, (x_n, y_n)$
- x_i is a vector of real-valued features $< x_{i1}, x_{i2}, \cdots, x_{im} >$
- $y_i \in \{1, -1\}$

Classification Algorithms - SVM (1/4)

- SVM: Support Vector Machine
- Forget about probabilities
- Training dataset: $(\mathtt{x}_1, \mathtt{y}_1), \cdots, (\mathtt{x}_n, \mathtt{y}_n)$
- $\mathtt{x}_i$ is a vector of real-valued features $< \mathtt{x}_{i1}, \mathtt{x}_{i2}, \cdots, \mathtt{x}_{im} >$
- $\mathtt{y}_i \in \{1, -1\}$
- Create function $\mathtt{f} : \mathtt{X} \rightarrow \mathtt{Y}$, and classify according to $\mathtt{f}(\mathtt{x})$.

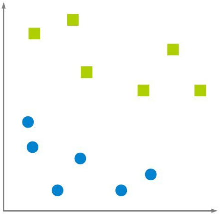
Classification Algorithms - SVM (2/4)

- x_i is a vector of real-valued features $< x_{i1}, x_{i2}, \cdots, x_{im} >$
- $y_i \in \{1, -1\}$
- Create function $f : X \rightarrow Y$, and classify according to $f(x)$.
- $f(x) = w_{i1}x_{i1} + w_{i2}x_{i2} + \cdots + w_{im}x_{im} + b = w^T x + b$

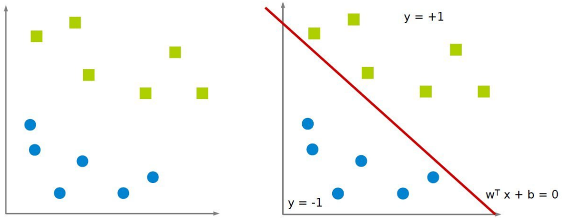
Classification Algorithms - SVM (2/4)

- x_i is a vector of real-valued features $< x_{i1}, x_{i2}, \cdots, x_{im} >$
- $y_i \in \{1, -1\}$
- Create function $f : X \to Y$, and classify according to $f(x)$.
- $f(x) = w_{i1}x_{i1} + w_{i2}x_{i2} + \cdots + w_{im}x_{im} + b = w^T x + b$
- if $f(x) > 0$, predict $y = 1$
- if $f(x) < 0$, predict $y = -1$

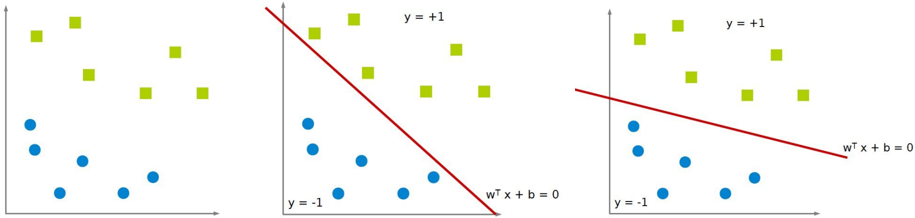
Classification Algorithms - SVM (3/4)



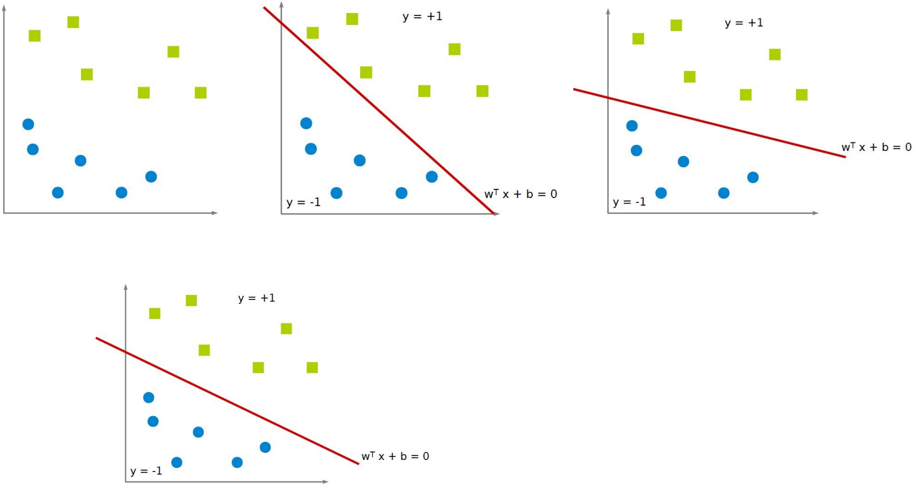
Classification Algorithms - SVM (3/4)



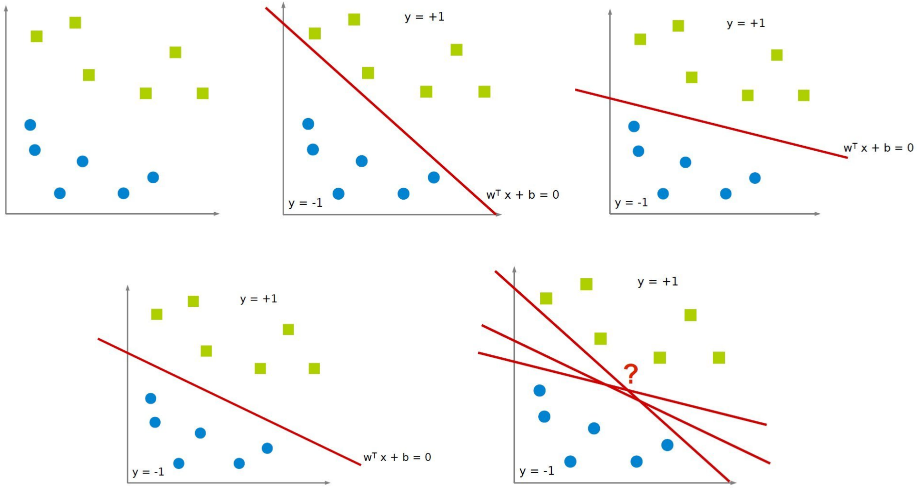
Classification Algorithms - SVM (3/4)



Classification Algorithms - SVM (3/4)

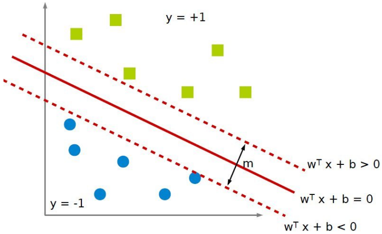


Classification Algorithms - SVM (3/4)



Classification Algorithms - SVM (4/4)

- The points closest to the separating hyperplane are known as support vectors.
- Maximize the distance from the separating line to the support vectors.
- We should maximize the margin, m.



Supervised Learning Algorithms

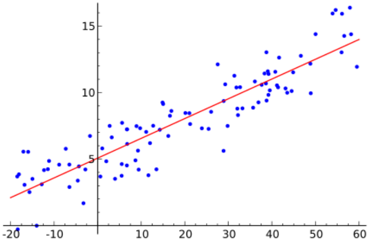
- Classification: the output variable takes discrete values.
- Regression: the output variable takes continuous values.

Classification Algorithms - Linear Regression (1/3)

- Training dataset: $(x_1, y_1), \cdots, (x_n, y_n)$.
- x_i is a vector of real-valued features $< x_{i1}, x_{i2}, \cdots, x_{im} >$
- $y_i = w_{i1}x_{i1} + w_{i2}x_{i2} + \cdots + w_{im}x_{im} + b = w^T x + b$

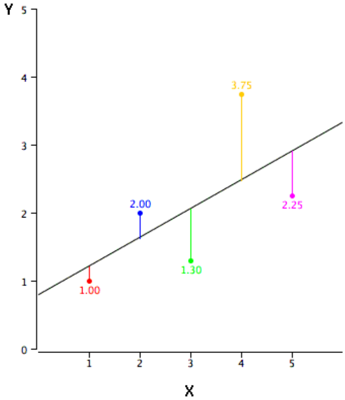
Classification Algorithms - Linear Regression (1/3)

- Training dataset: $(x_1, y_1), \cdots, (x_n, y_n)$.
- x_i is a vector of real-valued features $< x_{i1}, x_{i2}, \cdots, x_{im} >$
- $y_i = w_{i1}x_{i1} + w_{i2}x_{i2} + \cdots + w_{im}x_{im} + b = w^T x + b$
- Choose w so that $w^T x + b$ is close to y for our training dataset.



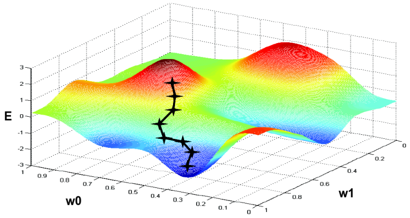
Classification Algorithms - Linear Regression (2/3)

- Choose w, such that it minimizes the cost function.
- Cost function: $E(w) = \frac{1}{2m} \sum_{i=1}^{m} (w^T x_i - y_i)^2$



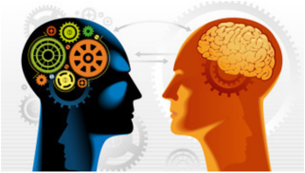
Classification Algorithms - Linear Regression (3/3)

- Cost function: $E(w) = \frac{1}{2m} \sum_{i=1}^{m} (w^T x_i - y_i)^2$
- $\min_{w} E(w)$
- Gradient descent:
  - Start with some w.
  - Keep changing w to reduce $E(w)$ until we hopefully end up at a minimum.



Types of Learning

- Supervised learning
- Unsupervised learning



Unsupervised Learning

- The data given to the learners are unlabeled.
- We want to explore the data to find some hidden structures in them.

Unsupervised Learning - Clustering

- Clustering is a technique for finding similarity groups in data, called clusters.

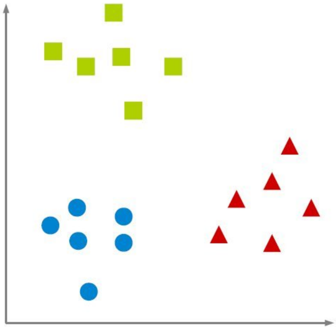
Unsupervised Learning - Clustering

- Clustering is a technique for finding similarity groups in data, called clusters.
- It groups data instances that are similar to each other in one cluster, and data instances that are very different from each other into different clusters.

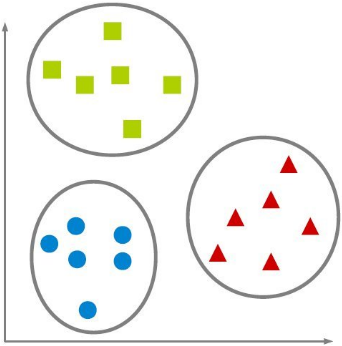
Unsupervised Learning - Clustering

- Clustering is a technique for finding similarity groups in data, called clusters.
- It groups data instances that are similar to each other in one cluster, and data instances that are very different from each other into different clusters.
- Clustering is often called an unsupervised learning task as no class values denoting an a priori grouping of the data instances are given.

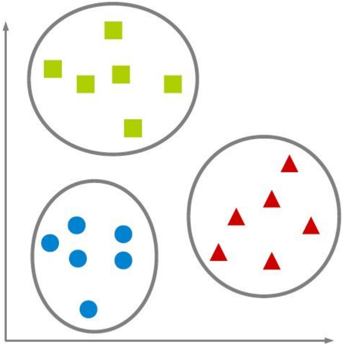
Unsupervised Learning - Clustering



Unsupervised Learning - Clustering



Unsupervised Learning - Clustering

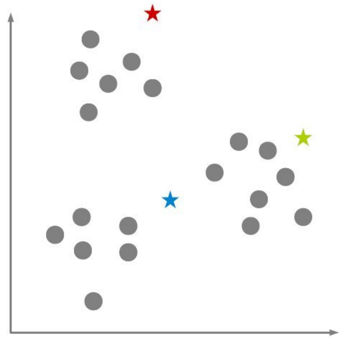


- k-means clustering is a popular method for clustering.

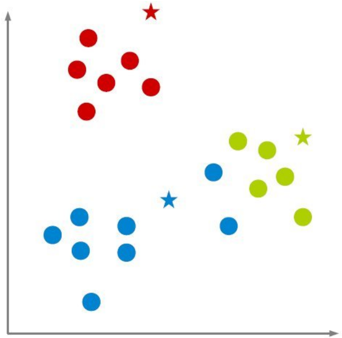
Clustering Algorithms - K-Mean Clustering (1/2)

- K: number of clusters (given)
    - One mean per cluster.
- Initialize means: by picking k samples at random.
- Iterate:
    - Assign each point to nearest mean.
    - Move mean to center of its cluster.

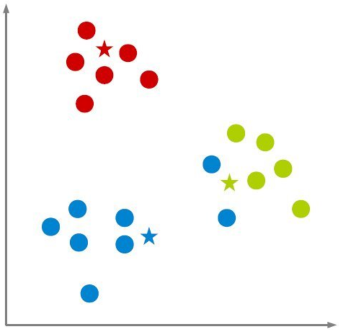
Clustering Algorithms - K-Mean Clustering (2/2)



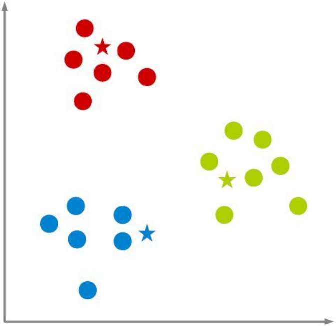
Clustering Algorithms - K-Mean Clustering (2/2)



Clustering Algorithms - K-Mean Clustering (2/2)



Clustering Algorithms - K-Mean Clustering (2/2)



Spark Machine Learning

MLBase Stack

- MLlib: ML library in Spark.
- MLI: APIs for simplified feature extraction and algorithm development.
- ML Optimizer: a declarative layer to simplify access to large scale ML.

MLlib

- Classification
  - Decision Trees, Naive Bayes, Logistic Regression, Linear SVM
- Regression
  - Linear Regression
- Clustering
  - K-Means

MLlib

- Classification
  - Decision Trees, Naive Bayes, Logistic Regression, Linear SVM
- Regression
  - Linear Regression
- Clustering
  - K-Means
- Collaborative Filtering
  - Alternating Least Squares (ALS)
- Optimization Primitives
  - SGD, Parallel Gradient

MLlib - Decision Tree (1/2)

```
// Load labeled data from a file into an RDD[LabeledPoint].
// The data format used here is <L>, <f1> <f2> ...
// LabeledPoint: label: Double, features: Array[Double]
val examples = MLUtils.loadLabeledData(sc, path).cache()
val splits = examples.randomSplit(Array(0.8, 0.2))
val training = splits(0)
val test = splits(1)

// algorithm: Classification or Regression
// impurity: criterion used for information gain calculation
// maxDepth: maximum depth of the tree
val strategy = new Strategy(algorithm, impurity, maxDepth)

// Create the model
val model = DecisionTree.train(training, strategy)
```

MLlib - Decision Tree (2/2)

```
val correctCount = test.filter(
  y => predictedValue(model, y.features) == y.label).count()

def predictedValue(model: DecisionTreeModel, features: Vector): Double = {
  if (model.predict(features) < 0.5) 0.0 else 1.0
}
```

For more details:
https://github.com/apache/spark/blob/master/examples/src/main/scala/org/apache/spark/examples/mllib
https://github.com/apache/spark/tree/master/mllib/src/main/scala/org/apache/spark/mllib/tree

MLlib - Naive Bayes

```
// Load labeled data from a LIBSVM format file into an RDD[LabeledPoint].
// The data format used here is {{label index1:value1 index2:value2 ...}}
val examples = MLUtils.loadLibSVMData(sc, path).cache()
val splits = examples.randomSplit(Array(0.8, 0.2))
val training = splits(0)
val test = splits(1)

// Create the model
val model = new NaiveBayes().run(training)

// Check the accuracy
val prediction = model.predict(test.map(_.features))
val predictionAndLabel = prediction.zip(test.map(_.label))
val accuracy = predictionAndLabel.filter(x => x._1 == x._2)
    .count().toDouble / test.count
```

For more details:
https://github.com/apache/spark/blob/master/examples/src/main/scala/org/apache/spark/examples/mllib
https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/classification/NaiveBayes.scala

MLlib - Logistic Regression

```
// Load labeled data from a LIBSVM format file into an RDD[LabeledPoint].
// The data format used here is {{label index1:value1 index2:value2 ...}}
val examples = MLUtils.loadLibSVMData(sc, path).cache()
val splits = examples.randomSplit(Array(0.8, 0.2))
val training = splits(0)
val test = splits(1)

// Create the model
val model = LogisticRegressionWithSGD.train(training, numIterations)

// Check the accuracy
val prediction = model.predict(test.map(_.features))
val predictionAndLabel = prediction.zip(test.map(_.label))
val accuracy = predictionAndLabel.filter(x => x._1 == x._2)
    .count().toDouble / test.count
```

For more details:
https://github.com/apache/spark/blob/master/examples/src/main/scala/org/apache/spark/examples/mllib
https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/classification/LogisticRegression.scala

MLlib - SVM

```
// Load labeled data from a LIBSVM format file into an RDD[LabeledPoint].
// The data format used here is {{label index1:value1 index2:value2 ...}}
val examples = MLUtils.loadLibSVMData(sc, path).cache()
val splits = examples.randomSplit(Array(0.8, 0.2))
val training = splits(0)
val test = splits(1)

// Create the model
val model = SVMWithSGD.train(training, numIterations)

// Check the accuracy
val prediction = model.predict(test.map(_.features))
val predictionAndLabel = prediction.zip(test.map(_.label))
val accuracy = predictionAndLabel.filter(x => x._1 == x._2)
    .count().toDouble / test.count
```

For more details:
https://github.com/apache/spark/blob/master/examples/src/main/scala/org/apache/spark/examples/mllib
https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/classification/SVM.scala

MLlib - Linear Regression

```scala
// Load labeled data from a LIBSVM format file into an RDD[LabeledPoint].
// The data format used here is {{label index1:value1 index2:value2 ...}}
val examples = MLUtils.loadLibSVMData(sc, path).cache()
val splits = examples.randomSplit(Array(0.8, 0.2))
val training = splits(0)
val test = splits(1)

// Create the model
val model = LinearRegressionWithSGD.train(training, numIterations)

// Check the accuracy
val prediction = model.predict(test.map(_.features))
val predictionAndLabel = prediction.zip(test.map(_.label))
val accuracy = predictionAndLabel.filter(x => x._1 == x._2)
    .count().toDouble / test.count
```

For more details:
https://github.com/apache/spark/blob/master/examples/src/main/scala/org/apache/spark/examples/mllib
https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/regression/LinearRegression.scala

MLlib - K-Mean Clustering

```
// Load and parse the data
val data = sc.textFile(...)
val parsedData = data.map(_.split(' ').map(_.toDouble))

// Cluster the data into k classes
val clusters = KMeans.train(parsedData, numClusters, numIterations)

// Evaluate clustering by computing within set sum of squared errors
val cost = clusters.computeCost(parsedData)
```

For more details:
https://github.com/apache/spark/blob/master/examples/src/main/scala/org/apache/spark/examples/mllib
https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/clustering/KMeans.scala

Summary

- Supervised learning
  - Classification: kNN, Decision Trees, Naive Bayes, Logistic Regression, SVM
  - Regression: Linear Regression

- Unsupervised learning
  - Clustering: kmeans

Questions?

Acknowledgements

Some slides were derived from Tiberio Caetano slides (NICTA), and Andrew Ng (Stanford University).