Intelligent Systems

Automatic learning systems (ALS)

T. Mihoc[1]

[1]Department of Computer Science
Babeș Bolyai University

Outline

Outline

Machine Learning

- Problem:
  - "How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes? "
- Applications:
  - Image and voice recognition
    - Handwritten recognition
    - Face detection
  - Computer vision
    - Obstacle detection
    - Footprint recognition
  - Bio-surveillance
  - Robot control
  - Predictions
  - Medical diagnostic
  - Fraud detection

Machine Learning

- Definition:
  - Arthur Samuel (1959): *field of study that gives computers the ability to learn without being explicitly programmed*
  - Herbert Simon (1970): *learning is any process by which a system improves performance from experience*
  - Ethem Alpaydin (2010): *programming computers to optimize a performance criterion using example data or past experience*
- Necessity:
  - Better computational systems
    - To difficult or to expensive to be constructed manually
    - Systems that automatically adapt (spam filters, Systems that discovers information in large databases $\rightarrow$ *data mining, financial analysis, text/image analysis*)
  - Understanding the biological systems

Outline

Design

- Improve task T
  - establish the goal (what has to be learn) objective function and its representation
  - select a learning algorithms to perform the inference of the goal based on experience
- Respect a performance metric P
  - Evaluation of the algorithms performances
- Based on experience E
  - Select an experience database

Examples

- T: Playing checkers
  P: percent of winning games
  E:playing the game
- T: handwritten recognition
  P: percent of correct recognized words
  E: database of images with different words
- T: separate the spams
  P: percent of correct classified emails
  E: databases with annotated emails

Design

- identify the function that must be learn
  Example for checkers game: a function that selects the next move/evaluates a move
- Representation of objective function
  - Different representations
    - Table
    - Symbolic rules
    - Numeric functions
    - Probabilistic functions
  - There is a trade-off between:
    - Expressiveness of representation
    - Facile learning
  - Computation of the objective function:
    - polynomial time
    - non-polynomial time

Design

Selecting an algorithm

- By using the training data
- Induce the hypothesis definition that:
  - Match the data
  - Generalize unseen data
- Main principle – error minimization (cost function – loss function)

Evaluation of a learning system

- Experimental
  - comparing different methods on different data (cross-validation)
  - collect data based on performances (accuracy, training time, testing time)
  - statistical analyze of the differences
- Theoretic – mathematical analyzes of algorithms and theorems proving
  - Computational complexity
  - Ability to match the training data
  - Complexity of the most relevant sample for learning

Design – evaluation of a learning system

- Comparing the performances of 2 algorithms for solving a given problem
  - Performance measures
  - Parameters of a statistic series
  - Proportion (percent) computed for a statistical series (ex. Accuracy)
- Comparing based on confidence intervals
  - one problem, two solving algorithms p_1, p_2
  - confidence intervals $I_1 = [p_1 - \Delta_1, p_1 + \Delta_1]$ and $I_2 = [p_2 - \Delta_2, p_2 + \Delta_2]$
  - if $I_1 \cap I_2 = \emptyset$ algorithm 1 works better than algorithm 2 (for the given problem)
  - if $I_1 \cap I_2 \neq \emptyset$ can't be decided

Design: choose the training database

Based on:

- direct experience: pairs (in, out) that are useful for the objective function

  Ex.: board annotated with correct or incorrect move
- indirect experience: useful feedback (unlike i/o pairs) for the objective function

  Ex.: sequences of moves and the final score of the game

Data sources:

- random generated examples (positive and negative examples);
- positive examples collected by a learner;
- real examples

Design: choose the training database

Content:

- training data
- test data

Characteristics

- Independent data (otherwise $\rightarrow$ collective learning)
- Training and testing data must respect the same distribution law (otherwise $\rightarrow$ transfer learning/inductive transfer)
  - vehicle recognition, truck recognition, text analyses, spam filters

Design: choose the training database

Characteristics extracted (attributes) from raw data

- Quantitative characteristics: $\rightarrow$ nominal or rational scale
  - Continuous values $\rightarrow$ weight
  - Discrete values $\rightarrow$ no. of computers
  - Range values $\rightarrow$ event times
- Qualitative characteristics
  - Nominal $\rightarrow$ colour
  - Ordinal $\rightarrow$ sound intensity (low, medium, high)
  - Structured $\rightarrow$ Trees root is a generalization of children (vehicle $\rightarrow$ car, bus, tractor, truck)

Design: choose the training database

Data transformation

- Standardization $\rightarrow$ numerical attributes
  - Remove the scale effect (different scale and units)
  - Raw values are transformed in z scores
    $z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$ where x_{ij} is the value of the j^{th} attribute of i^{th} instance, μ_j (σ_j) is the mean (respectively the standard deviation) of the j^{th} attribute of all instances
- Selection of some attributes

Machine Learning – Typology

Based on their aim (goal)

- ISs for prediction
  - Aim: predict the output for a new input based on a previously learned model
  - Eg. predicting sales of a product for a time in the future based on price, calendar month, region, average income
- ISs for regression
  - Aim: estimation of the (uni or multi - variable) function shape based on a previously learned model
  - Eg.: estimate the function that models the edge of a surface
- ISs for classification
  - Aim: classify an object into one or more known or unknown - categories based on their characteristics
  - Eg.: diagnostic systems for cancer: malign or benign or normal
- ISs for planning
  - Aim: generate a sequence of optimal actions for performing a task
  - Eg.: planning the moves of a robot from a position to a source of energy

Machine Learning – Typology

Based on the experience learned during training process

- ISs with supervised learning
- ISs with unsupervised learning
- ISs with active learning
- ISs with reinforcement learning

Outline

- Machine Learning
  - Problem Definition and Applications
  - Design
  - Automatic learning
    - Supervised learning
    - Unsupervised learning
    - Active learning; Reinforcement learning

Supervised leaning

Aim

To provide a correct output for a new entry.

Definition

Consider a training data set of N elements, organized in pairs $(entrydata_i, output_i)$ where $i = \overline{1, N}$ and a testing data set of n similar pairs.

- $entrydata_i = (atr_{i_1}, atr_{i_2}, ..., atr_{i_m})$, m the number of attributes for one data entry.
- $output_i$ can be:
  - a category from k classes (an element from a predefined set of k elements) $\rightarrow$ classification problem
  - a real number $\rightarrow$ regression problem

We denote by *Supervised learning* the process of finding a function (unknown) that maps the given entries to the outputs.

Supervised leaning

Other names

classification (regression), inductive learning

Process → 2 steps

- Training – learning, using a certain algorithm, the model from the training data
- Testing – testing the detected model using the test data (unseen data)

Characteristics

Features an experimental ad-notated Data Base for learning.

Supervised learning

Problems' types

- regression
  - Aim: prediction of an output for a new input
  - Continuous output (a real number)
  - Ex. prediction of prices
- classification
  - Aim: classifying a new input
  - Discreet output (a label from a predefined set)
  - Ex. detection of maligned tumors

Example of problems

- recognizing hand writing
- image recognition
- predicting the weather
- spam detection

Supervised learning
The quality of learning

Definition
The *quality of learning* is a measure of the algorithm's performance. This measure is determined during the training faze and the testing faze.

Example: Accuracy = the number of correct classified entries / the total number of entries.

Evaluation methods

- disjoints sets of training and testing
  - the training set is divided in two parts, one for learning one for testing
  - adequate for large data sets
- cross-validation with many (h) equal sub-sets
  - dividing the data set in h parts ($h-1$ for training 1 for testing)
  - the performance is the average on the h runs
  - adequate for small data sets
- leave-one-out cross-validation
  - similar with cross-validation but $h =$ the number of elements from the data set

Supervised learning

The quality of learning

Difficulties: over-fitting $\rightarrow$ very good performance for the learning data set and very poor on the test data.

Performance measures

- statistical measures
- efficiency (building and testing the model)
- scaling (efficiency for large data sets)
- compactness
- robustness (coping with noises and missing data)

Supervised learning

Statistical measures

- Accuracy
  - no. of correct classified / total no. of examples
  - the opposite of error
  - determined on the training set and on the testing set
- Precision (P)
  - no. of correct positive classified / total no. of examples classified as positive
  - the probability that a positive classified example is relevant
  - $TP/(TP+FP)$
- Rappel (R)
  - no. of correct positive classified / total no. of positive examples
  - the probability that a positive example is correct classified
  - $TP/(TP+FN)$
  - confusion matrix – real results versus computed results
- Score
  - combines P and R
  - $2PR/(P+R)$

Unsupervised learning

Aims:

- finding a model or a structure in a data set
- dividing a set of unlabeled examples in disjoint sets (clustering) such as:
  - the elements of a cluster are very similar
  - the elements of different clusters are very different

Definition

Consider a data set consisting of N elements. We define by *unsupervised learning* the process of determining a function (unknown) that groups the elements from the data set in several classes. The number of classes can be predefined k or unknown. The elements from one class are similar according to some criteria.

Unsupervised learning

Distance versus Similarity

- distance $\rightarrow$ min
- similarity $\rightarrow$ max

Examples of distances

- Euclidean: $d(p,q) = sqrt(\sum_{j=1}^{m}(p_j - q_j)^2)$
- Manhattan: $d(p,q) = \sum_{j=1}^{m}|p_j - q_j|$
- Mahalanobis: $d(p,q) = sqrt((p-q)^T S^{-1}(p-q))$, where S is the co-variance matrix
- Internal product: $d(p,q) = \sum_{j=1}^{m} p_j q_j$
- Cosine: $d(p,q) = \sum_{j=1}^{m} p_j q_j / (sqrt(\sum_{j=1}^{m} p_j^2) * sqrt(\sum_{j=1}^{m} q_j^2))$
- Hamming: the number of differences between p and q
- Levenshtein: the minimum no. of operations in order to transform p in q

Unsupervised leaning

Other names

clustering

Process → 2 steps

- Training – learning, using a certain algorithm, the clusters
- Testing – testing the detected model by placing a new element in a cluster

Characteristics

Features non ad-notated Data Base for learning.

Unsupervised learning

The quality of learning

- Internal criteria – high similarity inside the clusters and reduced similarity between elements from different clusters
  - distances inside the clusters
  - distances between the clusters
  - David-Bouldin index
  - Dunn index
- External criteria – using benchmark data sets that are previous grouped for testing
  - comparing with known data – almost impossible in real applications
  - precision
  - rappel
  - F-measure

Unsupervised learning

Internal criteria

- distances inside the cluster c_j that has n_j instances
  - average distance between points $D_a(c_j) = \sum_{x_{i_1}, x_{i_2} \in c_j} ||x_{i_1} x_{i_2}|| / (n_j * (n_j - 1))$
  - average of distances between the closest two points $D_{nn}(c_j) = \sum_{x_{i_1} \in c_j} min_{x_{i_2} \in c_j} ||x_{i_1} x_{i_2}|| / n_j$
  - distance between centroids $D_c(c_j) = \sum_{x_i \in c_j} ||x_i \mu_j|| / n_j$, where $\mu_j = 1/n_j \sum_{x_i \in c_j} x_i$
- distances between two clusters c_{j_1} and c_{j_2}
  - simple link: $d_s(c_{j_1}, c_{j_2}) = min_{x_{i_1} \in c_{j_1}, x_{i_2} \in c_{j_2}} \{||x_{i_1} x_{i_2}||\}$
  - complete link: $d_{co}(c_{j_1}, c_{j_2}) = max_{x_{i_1} \in c_{j_1}, x_{i_2} \in c_{j_2}} \{||x_{i_1} x_{i_2}||\}$
  - average link: $d_a(c_{j_1}, c_{j_2}) = \sum_{x_{i_1} \in c_{j_1}, x_{i_2} \in c_{j_2}} \{||x_{i_1} x_{i_2}||\} / (n_{j_1} * n_{j_2})$
  - centroid link: $d_{ce}(c_{j_1}, c_{j_2}) = ||\mu_{i_1} \mu_{i_2}||$

Unsupervised learning
Internal criteria

- David-Bouldin index $\rightarrow$ min $\rightarrow$ compact clusters
  $DB = 1/nc * \sum_{i=1}^{nc} \max_{j=\overline{1,nc}, j \neq i}((\sigma_i + \sigma_j)/d(\mu_i, \mu_j))$
  - nc - the no. of clusters,
  - μ_i the centroid of cluster i,
  - σ_i the average distance between the elements from cluster i to the centroid of the cluster,
  - d the distance between the centroids of two clusters
- Dunn index
  - identifies compact and well separated clusters
  - $D = D_{min}/D_{max}$
  - D_{min} minimum distance between two different clusters
  - D_{max} maximum distance between two elements from the same clusters

Unsupervised learning

Classification criteria: the clustering determination

- hierarchic
  - a taxonomic tree is built:
    $*$ cluster creation $\rightarrow$ recursive
    $*$ k is unknown (no. of clusters)
  - agglomerating $\rightarrow$ from small clusters towards big ones
  - divisive $\rightarrow$ from large clusters to small ones
- non-hierarchic
  - partitioning $\rightarrow$ dividing the data set $\rightarrow$ all clusters simultaneous
  - optimizes an objective function (local or global defined)
    - square error, graph based, probabilistic models, closest neighbour
  - k is predefined (no. of clusters)

Unsupervised learning

Classification criteria: the clustering determination

- based on data density
  - the clusters' creation is based on the data density in some region
  - the clusters' creation is based on the data connectivity in some region
  - one attempt to model the data distribution
  - advantage: the clusters can have any shape
- grid based
  - not exactly a new method
  - the space is segmented in regular zones
  - the objects are placed in a multidimensional grid (ex: ACO)

Unsupervised learning

Classification criteria: typology

- considering the algorithm behaviour
  - agglomerating
    1 every element is considered a cluster
    2 compute the distance between each two clusters
    3 the closest two clusters merge
    4 repeat steps 2 and 3 until a stop condition
  - divisive
    1 consider randomly from all elements centers for k clusters
    2 divide all the elements to the k clusters
    3 recompute the new centers for the clusters
    4 repeat steps 2 and 3 until the algorithm converges (no changes in the elements distribution occurred)
- considering the attributes
- considering the type of membership of elements to clusters

  – hard clustering versus fuzzy clustering

Active learning; Reinforcement learning

Active learning

The algorithm receives supplementary information during learning faze in order to improve its performances.

Reinforcement learning

The main feature is the interaction with the environment (actions lead to rewards)

Automatic learning

Classification criteria: the learned model

- Least Squares method
- Descending Gradient methods
- Evolutionary algorithms
- kNN
- Decision Trees
- Support vector machines
- Artificial neural networks
- Genetic programming
- Hidden Markov Models

Least Squares method

Consider a regression problem:

- input data $x \in R^d$
- output data $y \in R$
- determine a model f that maps x in y
- $f(x) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + ... + \beta_d * x_d$

we will define a function:

$$Loss = \sum_{i=\overline{1,d}} (y_i - f(x_i))^2$$

if we minimize it we obtain the optimal values for β

Descending Gradient Method

Modeling the β coefficients

- iteration 0: random values (or zero)
- iteration $t+1$:
  $\beta_k(t+1) = \beta_k(t) - learning_rate * error(t) * x$, where $k = \overline{1,d}$
  $\beta_0(t+1) = \beta_0(t) - learning_rate * error(t)$,

  where $error = computed - realOutput$