LabVIEW Machine Learning Toolkit
User Manual

1. Introduction

The idea of machine learning is to mimic the learning process of human beings, i.e., gaining knowledge through experience. Machine learning algorithms allow machines to generalize rules from empirical data, and, based on the learned rules, make predictions for future data. The Machine Learning Toolkit (MLT) provides various machine learning algorithms in LabVIEW. It is a powerful tool for problems such as visualization of high-dimensional data, pattern recognition, function regression and cluster identification.

2. Features

The Machine Learning Toolkit includes algorithms, data types, validation functions, and visualization tools.

2.1 Machine Learning Algorithms

2.1.1 Unsupervised learning algorithms

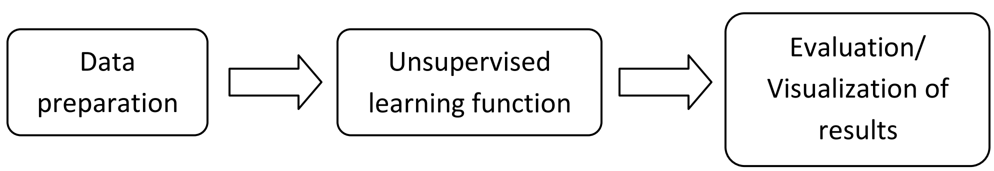
Unsupervised learning refers to the problems of revealing hidden structure in unlabeled data. Since the data are unlabeled, there is no error signal fed back to the learner in the algorithm. This distinguishes unsupervised learning from supervised learning.

Clustering is one of the main and important approaches of unsupervised learning. Clustering means the assignment of class memberships to a set of objects so that similar objects are assigned into the same class and dissimilar ones are assigned into different classes. Each class often represents a meaningful pattern in the respective problem. Clustering is thereby useful for identification of different patterns in data. For example, in image processing clustering can be used to divide a digital image into distinct regions for border detection or object recognition.

List of functions:

- *k*-means
- *k*-medians
- *k*-medoids
- Fuzzy C-means
- Gaussian Mixture Model (GMM)
- Hierarchical Clustering
- Spectral Clustering
- Vector Quantization (VQ)
- Self-Organizing Map (SOM)

Conceptual diagram of usage of the Machine Learning Toolkit for unsupervised learning



- Data preparation

  Data need to be formatted to fit the API of the unsupervised learning function the user selects.

- Unsupervised learning function application

  An unsupervised learning function is used to learn the structure of the input data.

- Evaluation/Visualization of results

  Refer to Section 2.4 for the choice of appropriate evaluation/visualization utility.

Examples:

- Example_Clustering
- Example_SOM

2.1.2 Supervised learning algorithms

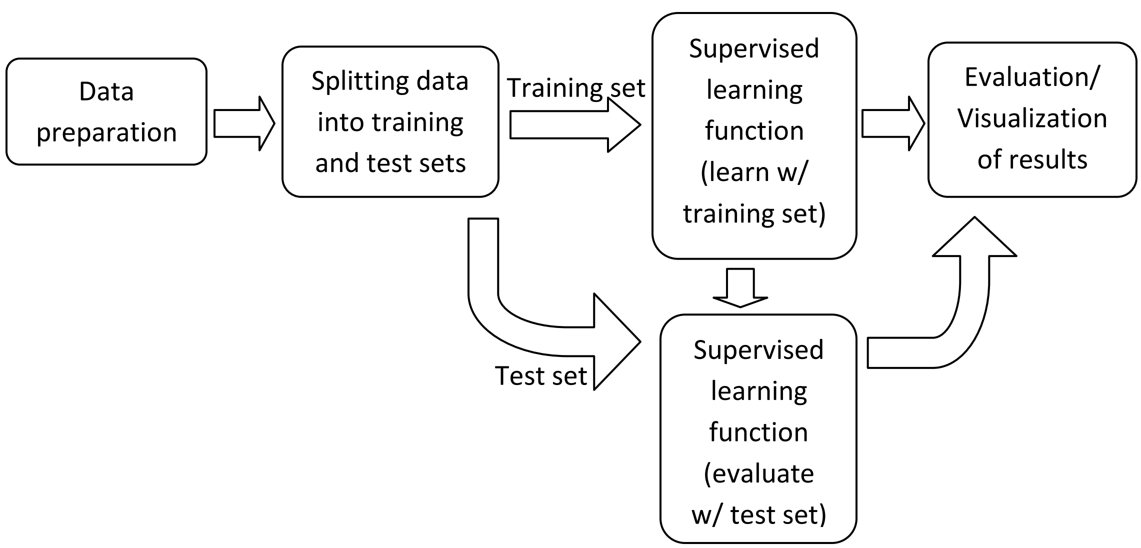
Supervised learning refers to the generalization of the relationship (function) between the input data and their corresponding outputs (labels). The relationship (function) is learned through a training set of examples, each of which is a pair of an input data and a desired output. During the training, the error between the actual and the desired outputs is frequently fed back into the system for tuning the system parameters according to certain learning rule. The system "learns" by adapting itself to minimize the error. After the training, the performance of the learned relationship (function) should be evaluated on a test set (of examples) that is separate from the training set.

Supervised learning is useful for pattern recognition, function regression, etc. One example of applications is recognition of handwritten numbers. A supervised classifier can be trained with a reservoir of handwritten numbers, each with a label (the true number each image represents). Having been validated on a separate test set, the trained classifier can be used for fast and accurate recognition of future handwritten numbers.

List of functions:

- *k*-Nearest Neighbors (*k*-NN)
- Back-propagation (BP) Neural Network
- Learning Vector Quantization (LVQ)
- Support Vector Machine (SVM)

Conceptual diagram of usage of the MLT for supervised learning



- Data preparation

  Data need to be formatted to fit the API of the unsupervised learning function the user selects.

- Splitting data into training and test sets

The MLT provides a utility (Training & Test Set.vi) to split original data into a training set and a test set with a user-specified ratio.

- Supervised learning function (learn w/ training set)

  The training set is used for the learning procedure.

- Supervised learning function (evaluate w/ test set)

  The test set is used for the evaluation of the performance.

- Evaluation/Visualization of results

  Refer to Section 2.4 for the choice of appropriate evaluation/visualization utility.

Examples:

- Example_BP Network_Classification
- Example_BP Network_Curve Fitting
- Example_LVQ
- Example_SVM

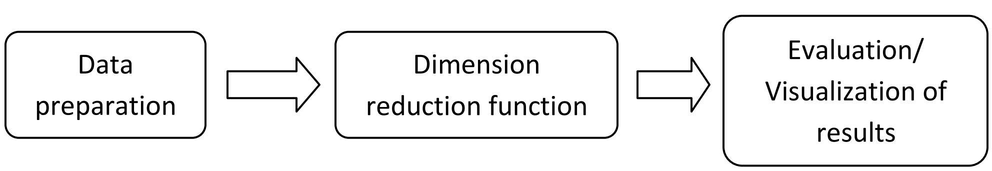
2.1.3 Dimension reduction algorithms

Dimension reduction refers to the process of reducing the number of dimension of the data. The projection of the data set in the reduced space is often desired to preserve certain important data characteristics. In some cases data analysis, such as clustering, can be done more easily and accurately in the reduced space than in the original space. One prime application of dimension reduction is face recognition, where face images represented by a large number of pixels are projected to a more manageable low-dimensional "feature" space before classification.

List of functions:

- Isometric Feature Mapping (Isomap)
- Locally Linear Embedding (LLE)
- Multidimensional Scaling (MDS)
- Principal Component Analysis (PCA)
- Kernel PCA
- Linear Discriminant Analysis (LDA)

Conceptual diagram of usage of the MLT for dimension reduction



- Data preparation

  Data need to be formatted to fit the API of the unsupervised learning function the user selects.

- Dimension reduction function

  A dimension reduction function is used to project the input data to a reduced space.

- Evaluation/Visualization of results

  Refer to Section 2.4 for the choice of appropriate evaluation/visualization utility.

Examples:

- Example_Manifold learning
- Example_LDA
- Example_Kernel PCA

2.2 Variant Data Type

For learning algorithms that require an input data type to be numeric, the data needs to be organized into a 2-D array of numeric numbers, where each row is an input sample. For learning algorithms that utilize the (dis)similarity relationships of the data samples as inputs, data samples can be any type of object. In this case, the input data needs to be organized into a 1-D array of variants. In addition, the user needs to specify the distance/kernel VI to use. Refer to Section 2.3 for the distance/kernel VI provided by the MLT.

Functions for which input data is a 1-D array of variants and a reference to a distance/kernel VI is a required input:

- *k*-medoids
- Hierarchical Clustering
- Spectral Clustering
- *k*-Nearest Neighbor (*k*-NN)
- Isometric Feature Mapping (Isomap)
- Locally Linear Embedding (LLE)
- Multidimensional Scaling (MDS)

Examples:

- Example_Clustering
- Example_Manifold learning

2.3 Distance/Kernel VI Reference

Some of the algorithms require the user to specify a distance/kernel VI. Refer to Section 2.2 for the list of applicable functions. The MLT provides some of the most frequently-used distances and kernel functions.

2.4 Validation & Visualization Utilities

The MLT provides validation and visualization utilities to facilitate the monitoring of the quality of learning. The utilities fall into three categories: cluster validity indices, evaluation of classification, visualization of learned results. The list of functions in each category is shown below.

Cluster validity indices:

- Rand Index
- Davies-Bouldin (DB) Index
- Jaccard Index
- Dunn Index

Evaluation of classification:

- Classification Accuracy
- Confusion Matrix

Visualization of learned results:

- Visualization (2D &3D)
- Plot SOM (2D &3D)

The applicability of each function to different algorithms is shown in Table I.

3. System Requirements

- Windows XP or later
- LabVIEW 2009 or later

4. Installation Notes

- Download and unzip the latest installer from NI Labs. Run **Setup.exe**.
- Launch LabVIEW so that the installed menus can rebuild.
- Open the diagram and go to **Addons** >> **Machine Learning**.

Table I. The applicability of validation and visualization utilities to different machine learning algorithms. “x” indicates that a utility is applicable to a certain algorithm.

			Validation Utility						**Visualization Utility**	
			Rand Index	DB Index	Dunn Index	Jaccard Index	Classification Accuracy	Confusion Matrix	Visualization (2D &3D)	Plot SOM (2D &3D)
Algorithm	**Unsupervised Learning**	k-means	x	x	x	x			x	
		k-medians	x	x	x	x			x	
		k-medoids	x	x	x	x			x	
		Gaussian Mixture Model	x	x	x	x			x	
		Fuzzy Cmeans	x	x	x	x			x	
		Hierarchical Clustering	x	x	x	x			x	
		Spectral Clustering	x	x	x	x			x	
		SOM							x	x
		VQ	x	x	x	x			x	
	Supervised Learning	k-NN					x	x	x	
		LVQ					x	x	x	
		SVM					x	x	x	
		BP neural network					x	x	x	
	Dimension Reduction	Isomap							x	
		LLE							x	
		LDA							x	
		MDS							x	
		PCA							x	
		Kernal PCA							x	