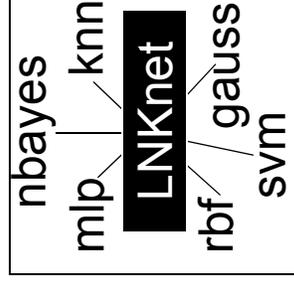


GETTING STARTED WITH LNKnet

A Quick Introduction

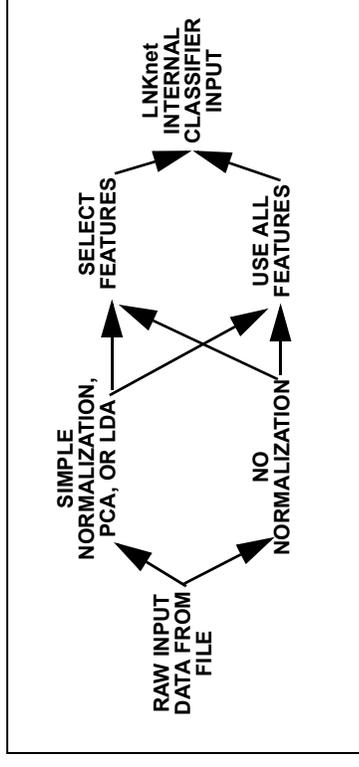


Linda Kukulich
Richard Lippmann

MIT Lincoln Laboratory
10 February 2004

5. Data Preprocessing Order

LNKnet supports three normalization methods which can be applied to input patterns before they are presented to a classifier. LNKnet also supports feature selection of normalized data. These preprocessing steps are applied as the data is being read into the classification program. Any normalization is applied first, followed by any requested feature selection, as shown in the figure below.



6. How to Obtain LNKnet

LNKnet software is available from MIT Lincoln Laboratory
([HTTP://www.ll.mit.edu/IST/](http://www.ll.mit.edu/IST/), EMAIL:kukolich@ll.mit.edu) as described by the included public-domain license.

Revised, May 1995
Revised, May 1999
Revised, February 2004

Public Domain License

LNKnet software is public domain software made available from MIT Lincoln Laboratory (<http://www.ll.mit.edu/IST/>). The following describes the notice that must accompany any redistribution of LNKnet.

(c) Copyright 2004 M.I.T. Lincoln Laboratory

Permission is hereby granted, without written agreement or royalty fee, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following three paragraphs appear in all copies of this software and/or documentation.

IN NO EVENT SHALL M.I.T. LINCOLN LABORATORY BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF M.I.T. LINCOLN LABORATORY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

M.I.T. LINCOLN LABORATORY SPECIFICALLY DISCLAIMS ANY WARRANTIES INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.

THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS AND M.I.T. LINCOLN LABORATORY HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

4. LNKnet ALGORITHMS

These classification, clustering, feature selection, and normalization algorithms are included in LNKnet:

	SUPERVISED TRAINING	COMBINED SUPERVISED-UNSUPERVISED TRAINING	UNSUPERVISED TRAINING (Clustering)
NEURAL NETWORK ALGORITHMS	Back-Propagation(BP) Adaptive Stepsize BP Cross-Entropy BP Top-2-Difference BP Hypersphere Classifier Committee Classifier	Radial Basis Function (RBF) Incremental RBF (IRBF) Top-2-Difference IRBF Learning Vector Quantizer Nearest-Cluster Classifier	Leader Clustering
MACHINE LEARNING AND STATISTICAL PATTERN CLASSIFICATION ALGORITHMS	Gaussian Linear Discriminant Quadratic Gaussian K-Nearest Neighbor (KNN) Condensed KNN Binary Tree Parzen Window Support Vector Machine (SVM) Naive Bayes	Gaussian Mixture (GMIX) Classifier Diagonal/Full Covariance GMIX Tied/Per-Class Centers GMIX	K-Means Clustering E&M Clustering
FEATURE SELECTION ALGORITHMS	Canonical Linear Discriminant Analysis Forward and Backward Search using N-fold Cross Validation		Principal Components Analysis

GETTING STARTED WITH LNKnet

UNIX SETUP

Find LNKnet Home Directory

This is the directory where LNKnet was installed. Ask your system administrator for the directory path. On my system this directory is /home/kukolich/lnknet.

Add LNKnet Binary Directory to your path

This directory contains all the executables for LNKnet. Add this directory to your \$PATH environment variable which is defined in your .cshrc or .path file in your home directory. On my system the LNKnet binary directory is /home/kukolich/lnknet/bin. On your system it will be below your LNKnet home directory.

1. Format for Data Files

Files in a Data Base

The data used by LNKnet to train and test classifiers are contained in ASCII data bases. Each data base has a description file named `<data_base>.defaults` and up to three data files named `<data_base>.train`, `<data_base>.eval`, and `<data_base>.test`. The description file provides the numbers of input features and output classes. The names of the input features and output classes can also be stored in the description file. The data files store the input feature values for each input pattern and integer class indices for every pattern. The LNKnet XOR data base is used in the tutorial below. The files in this data base are named `XOR.defaults`, `XOR.train`, and `XOR.test`.

Look at examples in LNKnet Data Directory

The example data bases included in LNKnet are stored in the LNKnet data directory. On my system this directory is `/home/kukolich/lnknet/data/class`. On your system it will be below your LNKnet home directory. Use a text scanner or editor like more, vi, or emacs to look at some sample data base files. More information about these data bases is included in the LNKnet User's Guide.

Description File Format

A description file contains one line that looks like a UNIX command issued in a shell script. The following is an annotated example of the data base description file for the XOR data base, `XOR.defaults`. Note that the input feature names "X0,X1" and the class labels "EVEN,ODD" are in comma delimited lists which can contain no spaces, dashes or colons. We highly recommend that you use the **Generate Description Files** option on the **Data Base Selection** window of LNKnet to create this file automatically for new data bases.

9. Plot Generation

To generate LNKnet plots you must select them on the plot selection window. To generate decision region and profile plots during training you must also select "Create plots for training and test files". The plot selection window is displayed by selecting the "PLOTS..." button in the middle of the right hand side of the LNKnet main window.

10. LNKnet Text Outputs and Log Files

LNKnet classifiers write outputs to the shell window from which the LNKnet graphical user interface was started. You may not see this information come out after pushing the START or CONTINUE button if the original shell window is covered by other windows. These outputs are also stored in a log file. Log files can contain a list of the flags used to call the classifier, training status information, a classification confusion matrix, and a classification error summary. How much of this information is output depends on the log file verbosity which is set on the LNKnet Report Files and Verboseities window.

11. Experiment Notebook File

An experiment notebook file, LNKnet.note, stores concise summary results for all experiments run from LNKnet. It is stored in the directory in which you started LNKnet. To store it somewhere else, change the notebook file name on the Report Files and Verboseities window to include a directory name.

12. Erase Old Error Files

When the Error File Verbosity on the Report Files and Verboseities window is set to a value other than "None", and the "No Training Error File" box is not checked, pattern by pattern information is stored in the experiment directory in an error file with ".err" in its name. This file can become large with incrementally trained classifiers trained over many epochs. Erase these files when they are no longer needed.

your data is by editing the first line labeled “Data Path” in the Data Base selection window.

5. Data base contents

A LNKnet data base consists of a description file and several ASCII data files. The format of these files is described above on pages 2 and 3 and in the LNKnet User’s Guide in Chapter 8. Examples of these files can be found in the LNKnet data base directory.

6. Experiment directory contents

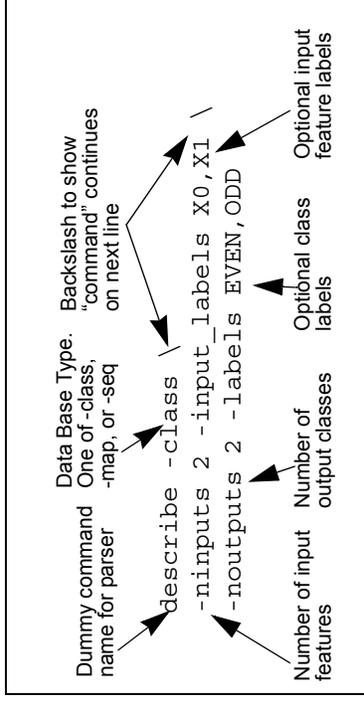
Shell scripts, screen files, log files, parameter files, error files, code files, commitee data base files, and plot files are stored in the experiment directory. If the experiment path displayed on the Report Files and Verboisities window is not a valid path, these experiment files cannot be written and the shell scripts LNKnet produces cannot be run.

7. Defaults are OK

Most LNKnet parameters have been set to reasonable default values and do not need to be changed to run experiments.

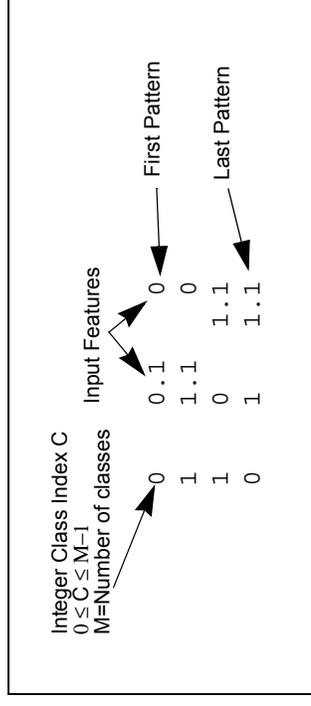
8. Graying out is OK

When a LNKnet parameter is grayed out it means that parameter is inconsistent with some previously selected parameter or is only consistent with some unselected parameter. For example, the color check box on the decision region plot window is grayed out when a clustering algorithm is selected. Clustering algorithms do not assign classes to patterns, so color coded decision region plots cannot be created. Instead a black and white plot is produced that shows the region controlled by each cluster center. As another example, the cross validation parameters on the main window are always grayed out except when N-fold cross validation is selected as the experiment action.



Data File Format

Data files contain one line for each input pattern. The first field on each line is an index for the class of the pattern. The indices start at 0 for the first class. The remaining fields are the floating point values for the input features. These fields are separated by spaces or tabs. Each line ends with a ‘\n’ newline character. The following is an annotated example of a data file with two input features, two classes, and four patterns.



Error File Format

When an experiment is run, pattern by pattern results can be stored in an error file. The amount of information included about each pattern

depends on the error file verbosity flag, `-verror`. This flag is set automatically in LNKnet when plots are selected that require error files or it can be set from the **Report Files and Verbsities** window.

Error file verbosity	Error file contents
<code>-verror 0</code>	No error file
<code>-verror 1</code>	Pattern number, true class, selected class, (true != selected), cost
<code>-verror 2</code>	Append class outputs to <code>-verror 1</code> entries
<code>-verror 3</code>	Append inputs as seen by classifier (after normalization and feature selection) to <code>-verror 2</code> entries

Pattern Number	Cost of Outputs	Classifier Outputs	Input Pattern
0	0	1	0.1 0.1
1	1	0	1.1 0.1
2	0	0	1.1 1.1
3	1	0	0.1 1.1

True Class Class Selected by Classifier Whether this was an Incorrect Classification

3. A Dozen Things you should know about LNKnet

1. Hit carriage return

Hit carriage return or tab whenever you type a name, number, or list on a LNKnet window. Hitting return while the cursor is on a line informs LNKnet of the new contents of the line. If you forget the carriage return, the line will look correct but LNKnet's internal variables will not be updated and the change will not appear in shell scripts.

2. Close windows to clean up desktop

Close popup windows and quit plots to cure desktop clutter. LNKnet has many popup windows and plots which can completely cover your desktop. Close the popups by hitting the close button on your keyboard or selecting the pushpin with your mouse. If you use a window manager other than `olwm`, delete (not destroy) the popup windows. Plot windows can be removed by typing 'q' while the mouse is over the window, selecting quit from the window's file menu, or by having the window manager quit or destroy the plot window.

3. Bin directory

LNKnet binaries are stored in `Inknet/bin` where `Inknet` is the LNKnet home directory which is the directory where the LNKnet package was installed.

4. Data directory location

The LNKnet classification data directory is `Inknet/data/class` where `Inknet` is the LNKnet home directory. Your own data files can, of course, be put in any directory you choose. You tell LNKnet where

2. A Quick Experiment

In this experiment you will train and test a Gaussian classifier which solves the XOR problem.

Step 1: Copy Data and Start LNKnet

Copy the data for this experiment from the lnknet data directory (/home/kukolich/lnknet/data/class on my system) to an experiment directory, then start LNKnet in the background. Note that you don't normally need to copy the data to your directory, but doing so will make selecting the data base during the tutorial simpler.

```
> cd ~
> mkdir QuickStart
> cd /home/kukolich/lnknet/data/class
> cp XOR.defaults ~/QuickStart
> cp XOR.train XOR.test ~/QuickStart
> cd ~/QuickStart
> lnknet&
```

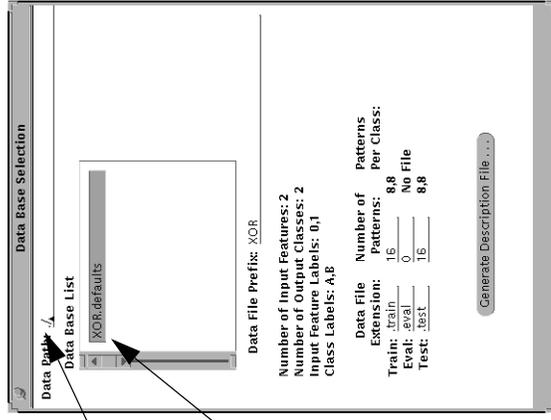
Step 2: Train and Test

Use the Left (Select) Mouse Button to select **Train and Test** as the **ACTION** and **Test** as the **Test File** on the Main window.

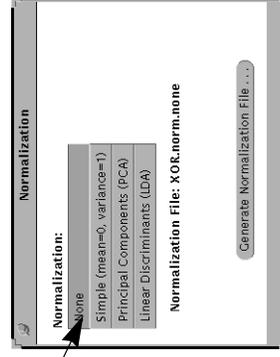
Step 3: Select Classifier

Use the Right (Menu) Mouse Button to select the Gaussian Classifier from the Algorithm menu on the Main window.

Data Base Selection Window, after Step 5

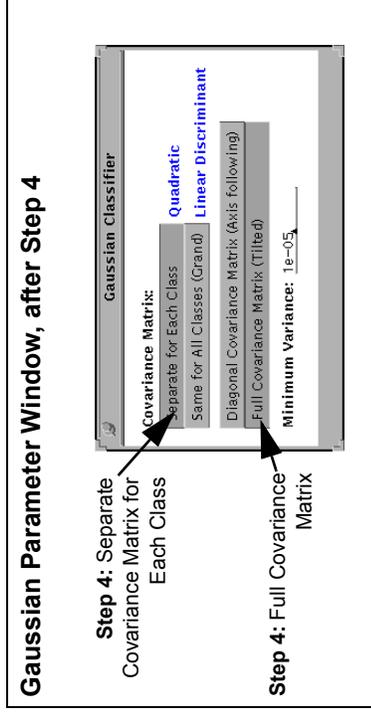


Normalization Window, after Step 6



Step 4: Set Gaussian Parameters

Select the **Algorithm Params...** button to display the Gaussian parameter window. Select **Separate for Each Class** and **Full Covariance Matrix**.



Step 5: Select Data Base

Select the **Data Base...** button on the Main LNKnet window to display the Data Base Selection window. Select **XOR.defaults** from the Data Base List.

Step 6: Select Normalization

Select the **Feature Normalization...** button on the main LNKnet window to display the Normalization window. Select **None** to turn off normalization for this experiment.

Step 7: Run Experiment

Select **START** on the Main window to train a Gaussian classifier with full per class covariance matrices using the unnormalized XOR training data and then test that classifier on the XOR test data. The results of the experiment will be printed to the shell window from which you started LNKnet. The classifier should achieve perfect results on the test data.

Main LNKnet window, after Step 7

