MATLAB for Machine Learning

Reporter: Zhaoshuai Liu

School of Data and Computer Science
Sun Yat-sen University

24747380@qq.com

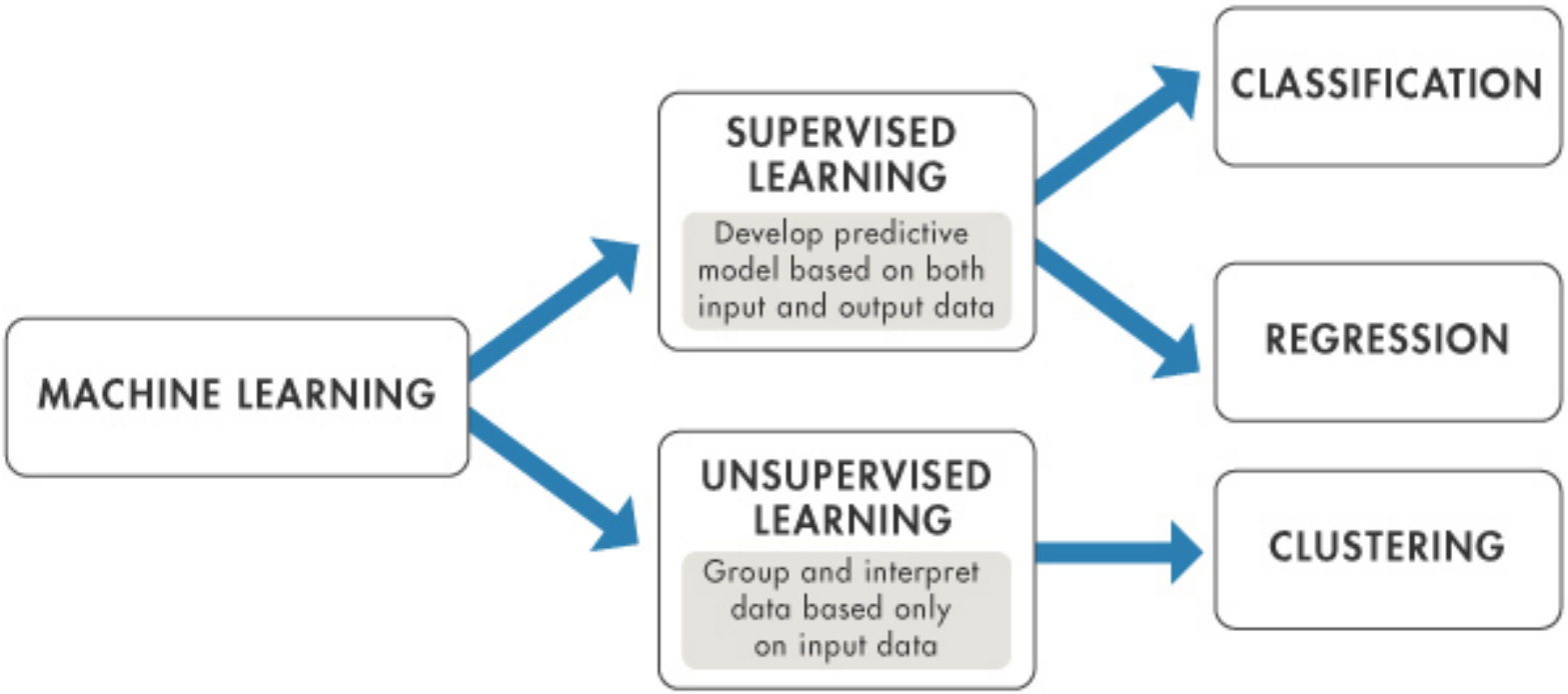
Overview

What Is Machine Learning?

- Machine learning teaches computers to do what comes naturally to humans: learn from **experience**.
- Machine learning algorithms use computational methods to "learn" information directly from **data**.
- The algorithms adaptively improve their performance as the number of **samples** available for learning **increases**.

Techniques of Machine Learning

- **Supervised learning**: train a model on known input and output data so that it can predict future outputs.
- **Unsupervised learning**: find hidden patterns or intrinsic structures in input data.

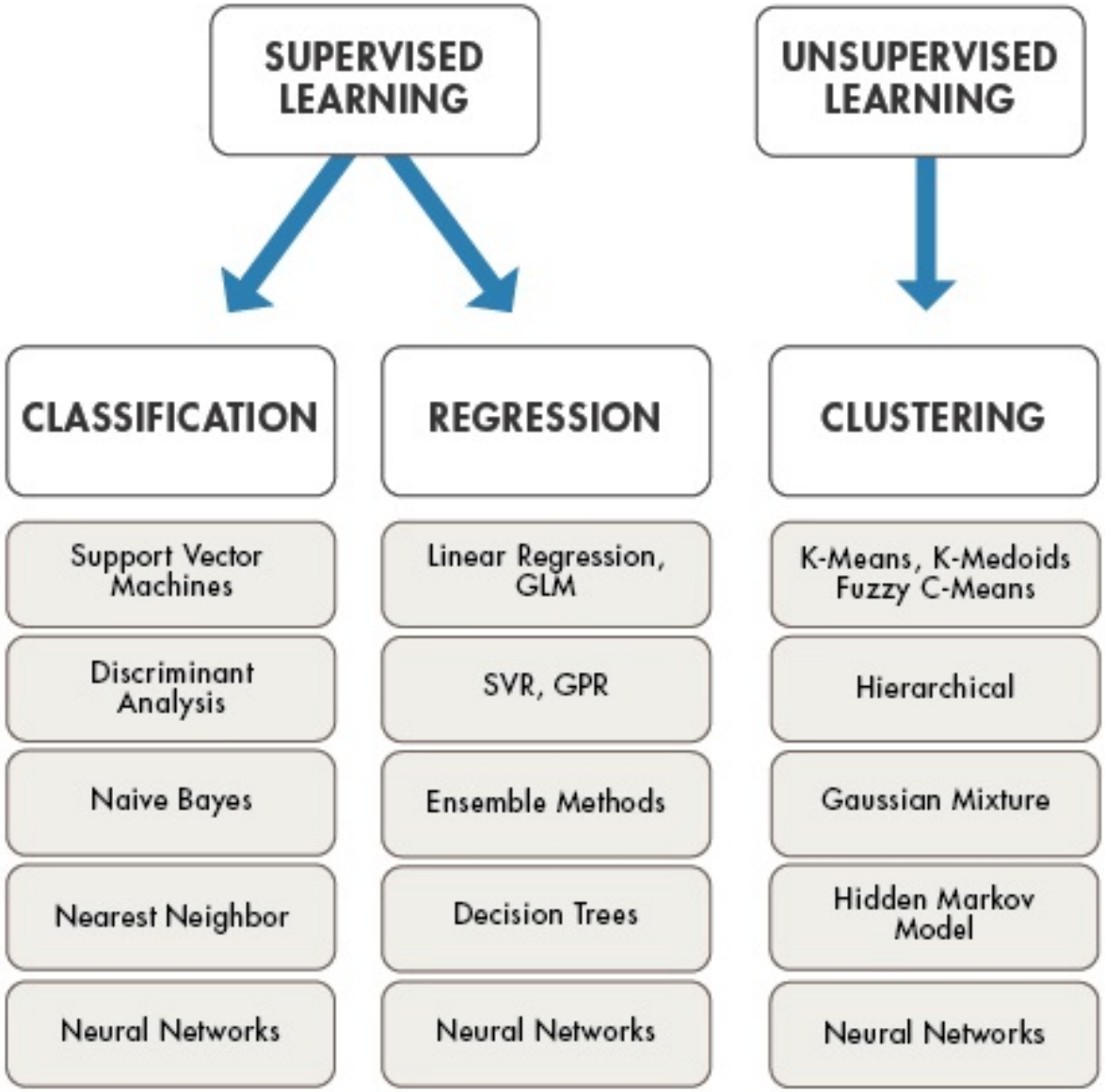


Regression vs classification

Andrew Ng:

- *Supervised learning problems are categorized into "regression" and "classification" problems.*
- *In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function.*
- *In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.*

Machine Learning Algorithms

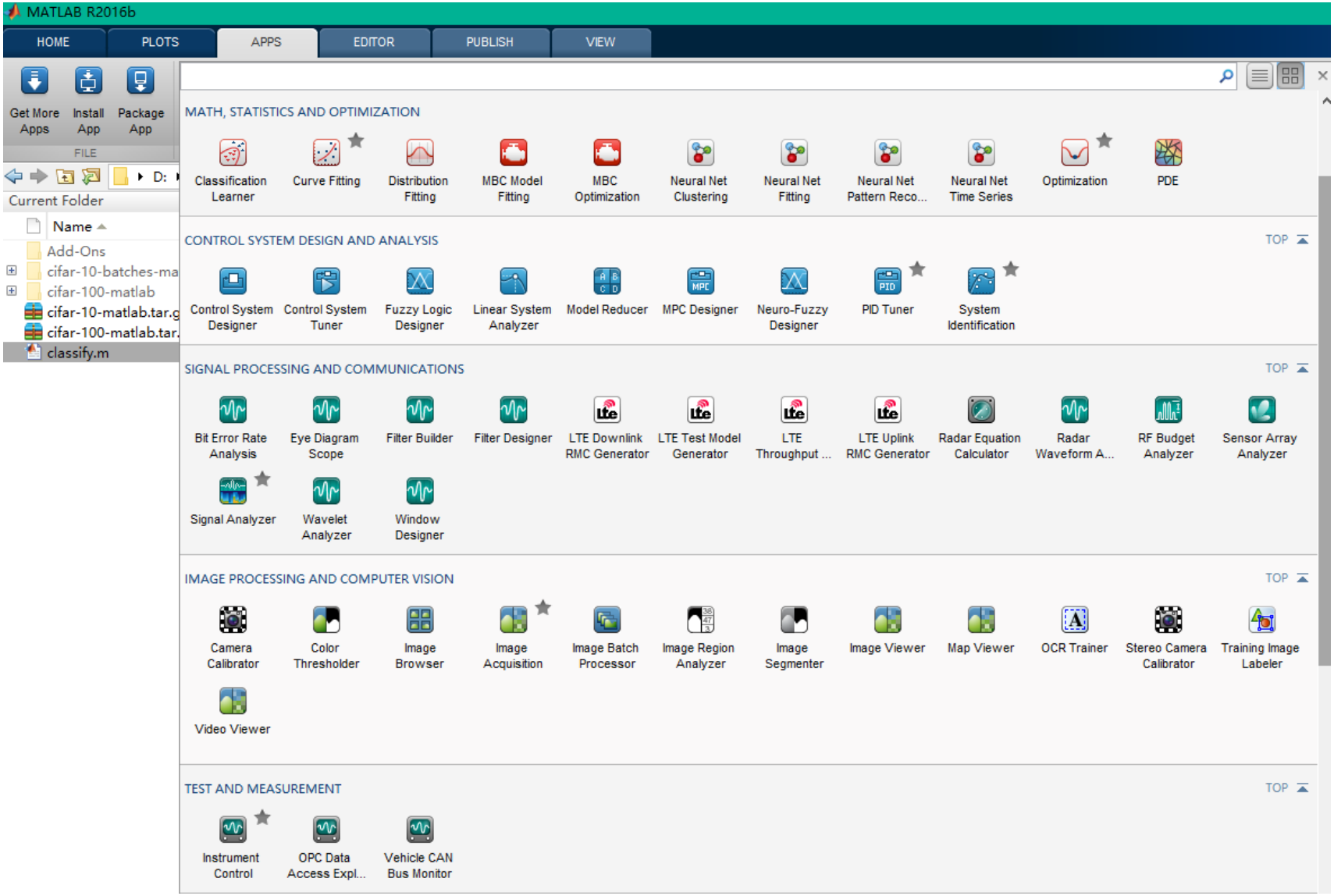


Features of MATLAB for Machine Learning

MATLAB makes the hard parts of machine learning easy with:

1. **Point-and-click apps** for training and comparing models
2. **Automatic** hyperparameter tuning and feature selection to optimize model performance
3. The ability to use the **same code** to scale processing to big data and clusters
4. Automated generation of **C/C++ code** for embedded and high-performance applications
5. Popular **classification**, **regression**, and **clustering** algorithms for supervised and unsupervised learning
6. **Faster** execution than open source on most statistical and machine learning computations

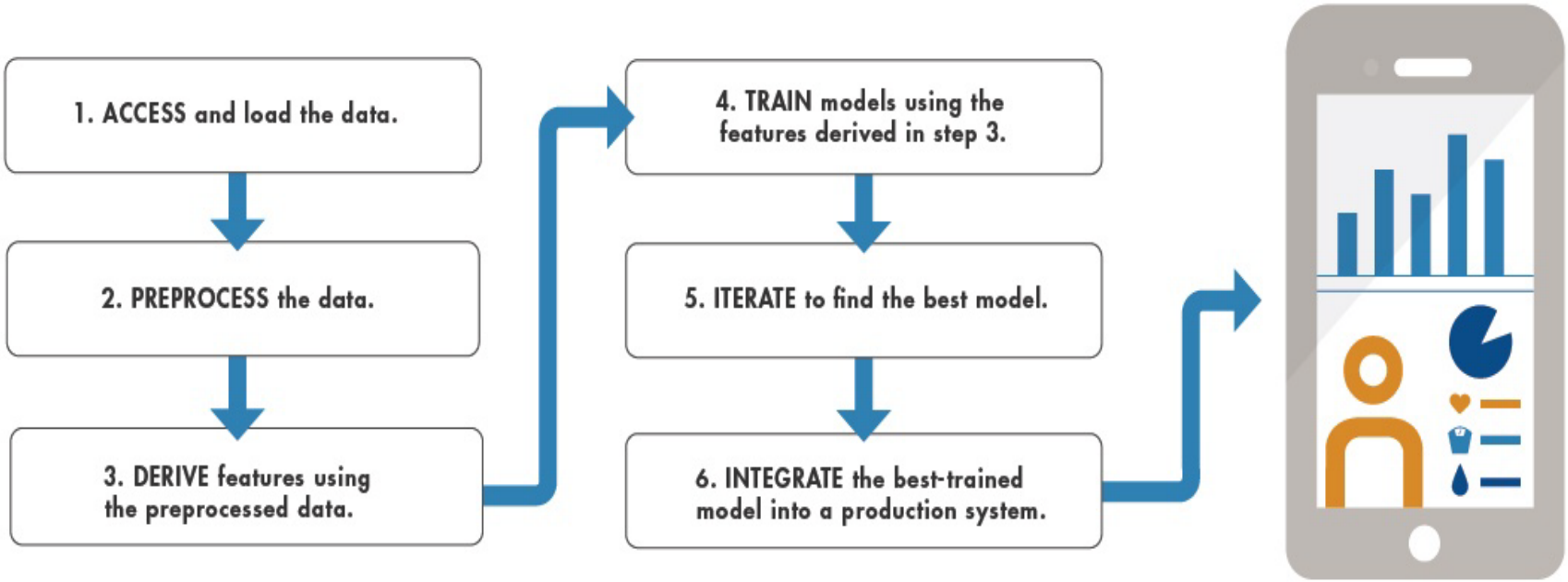
MATLAB apps and functions



MATLAB apps and functions (cont'd)

Task	MATLAB Apps and Functions	Product
Classification to predict categorical responses	Use the Classification Learner app to automatically train a selection of models and help you choose the best. You can generate MATLAB code to work with scripts. For more options, you can use the command-line interface.	Statistics and Machine Learning Toolbox™
Regression to predict continuous responses	Use the Regression Learner app to automatically train a selection of models and help you choose the best. You can generate MATLAB code to work with scripts and other function options. For more options, you can use the command-line interface.	Statistics and Machine Learning Toolbox
Clustering	Use cluster analysis functions.	Statistics and Machine Learning Toolbox
Computational finance tasks such as credit scoring	Use tools for modeling credit risk analysis.	Financial Toolbox™ Risk Management Toolbox™
Deep learning with neural networks for classification and regression	Use pretrained networks and functions to train convolutional neural networks.	Deep Learning Toolbox™
Facial recognition, motion detection, and object detection	Use deep learning tools for image processing and computer vision.	Deep Learning Toolbox Computer Vision Toolbox™

The Entire Workflow in MATLAB



MATLAB Toolboxes for Machine Learning

- MATLAB toolboxes are collections of MATLAB **functions** solving particular classes of problems.
- MATLAB has specific toolboxes for processing machine learning problems:
  - **Statistics and Machine Learning Toolbox**
  - Deep Learning Toolbox
  - Reinforcement Learning Toolbox

1. Introductions to Machine Learning and MATLAB

2. MATLAB Toolboxes for Machine Learning
   - Statistics and Machine Learning Toolbox
   - Deep Learning Toolbox
   - Reinforcement Learning Toolbox

3. Reference Materials

Statistics and Machine Learning Toolbox

Statistics and Machine Learning Toolbox provides **functions** and **apps** to describe, analyze, and model data.

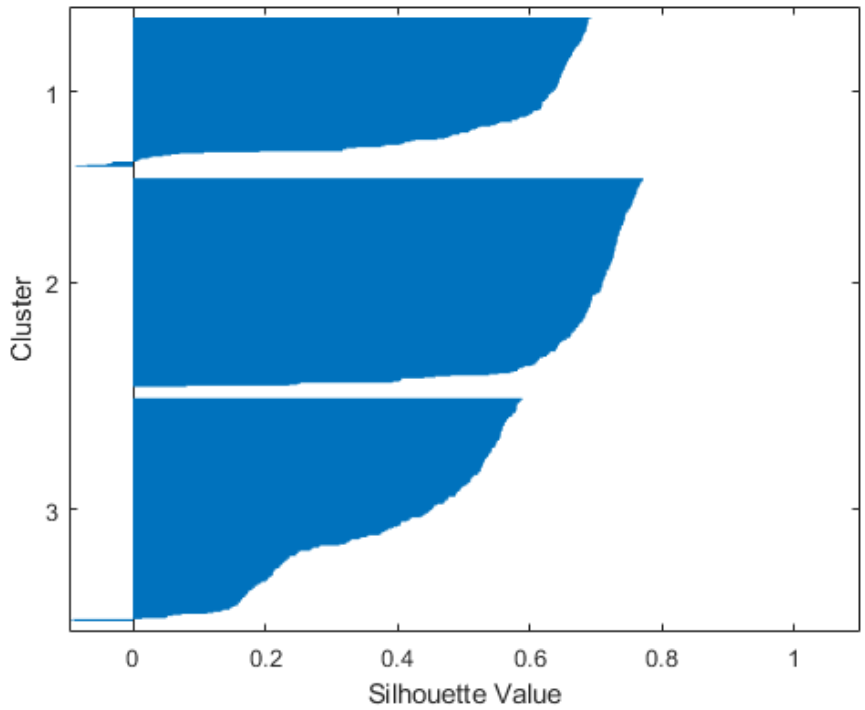
- **Multidimensional data analysis**: feature selection, stepwise regression, principal component analysis (PCA), regularization, and other dimensionality reduction methods.
- **Supervised and unsupervised machine learning algorithms**: support vector machines (SVMs), boosted and bagged decision trees, k-nearest neighbor, k-means, k-medoids, hierarchical clustering, Gaussian mixture models, and hidden Markov models.

Statistics and Machine Learning Toolbox
$\approx$ clustering + classification + regression

Method	Basis of Algorithm	Input to Algorithm	Requires Specified Number of Clusters	Cluster Shapes Identified
Hierarchical Clustering	Distance between objects	Pairwise distances between observations	No	Arbitrarily shaped clusters, depending on the specified `'Linkage'` algorithm
k-Means Clustering and k-Medoids Clustering	Distance between objects and centroids	Actual observations	Yes	Spheroidal clusters with equal diagonal covariance
Density-Based Spatial Clustering of Algorithms with Noise (DBSCAN)	Density of regions in the data	Actual observations or pairwise distances between observations	No	Arbitrarily shaped clusters
Gaussian Mixture Models	Mixture of Gaussian distributions	Actual observations	Yes	Spheroidal clusters with different covariance structures
Nearest Neighbors	Distance between objects	Actual observations	No	Arbitrarily shaped clusters
Spectral Clustering (Partition Data Using Spectral Clustering)	Graph representing connections between data points	Actual observations or similarity matrix	Yes, but the algorithm also provides a way to estimate the number of clusters	Arbitrarily shaped clusters

```
[idx3,C,sumdist3] = kmeans(X,3,'Distance','cityblock','Display','final');
```

```
[silh3,h] = silhouette(X,idx3,'cityblock');
xlabel('Silhouette Value')
ylabel('Cluster')
```



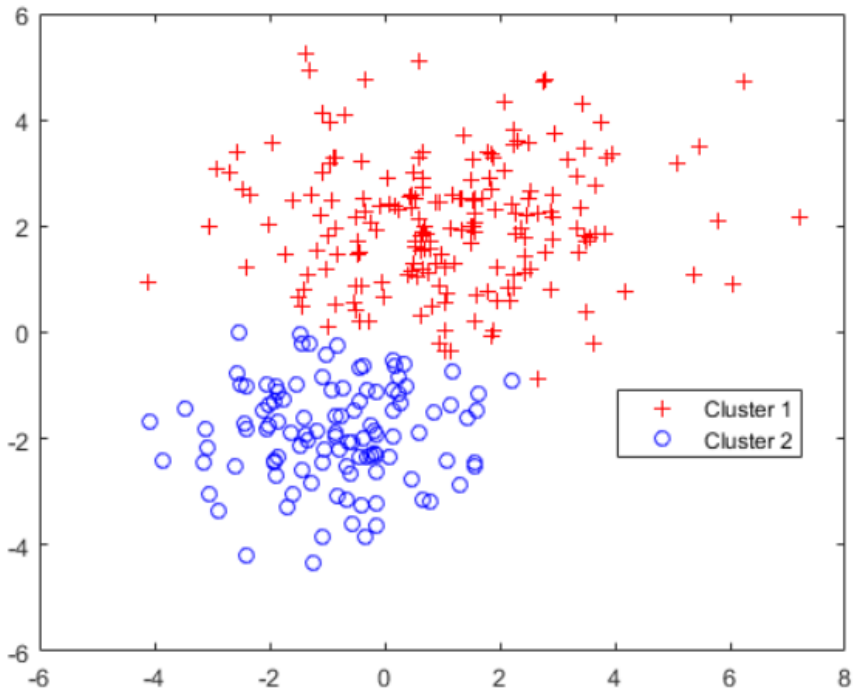
A silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.

https://ww2.mathworks.cn/help/stats/k-means-clustering.html

Statistics and Machine Learning Toolbox

Clustering Using Gaussian Mixture Model

```
idx = cluster(gm,X);
cluster1 = (idx == 1); % |1| for cluster 1 membership
cluster2 = (idx == 2); % |2| for cluster 2 membership

figure
gscatter(X(:,1),X(:,2),idx,'rb','+o')
legend('Cluster 1','Cluster 2','Location','best')
```

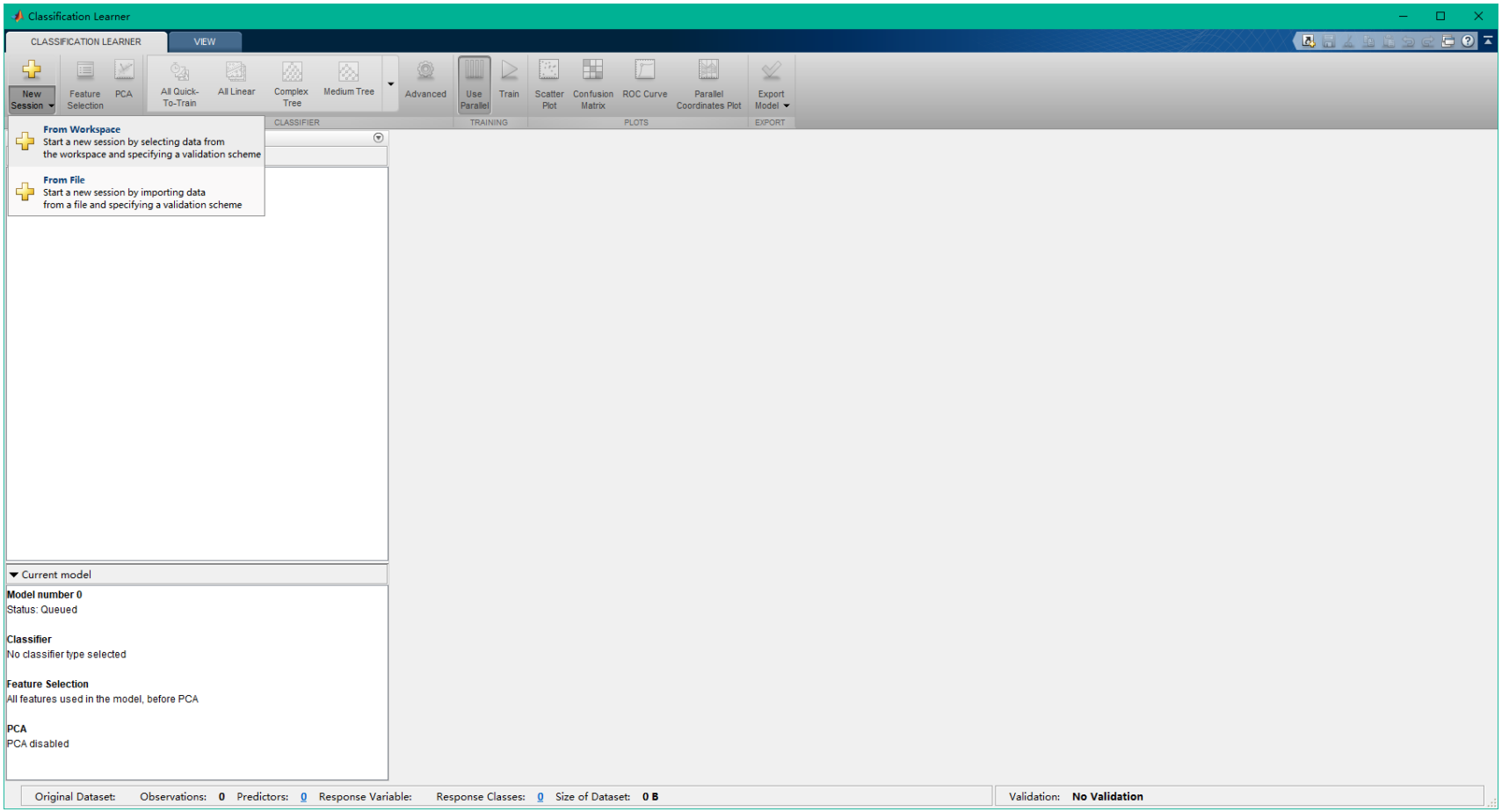


https://ww2.mathworks.cn/help/stats/clustering-using-gaussian-mixture-models.html

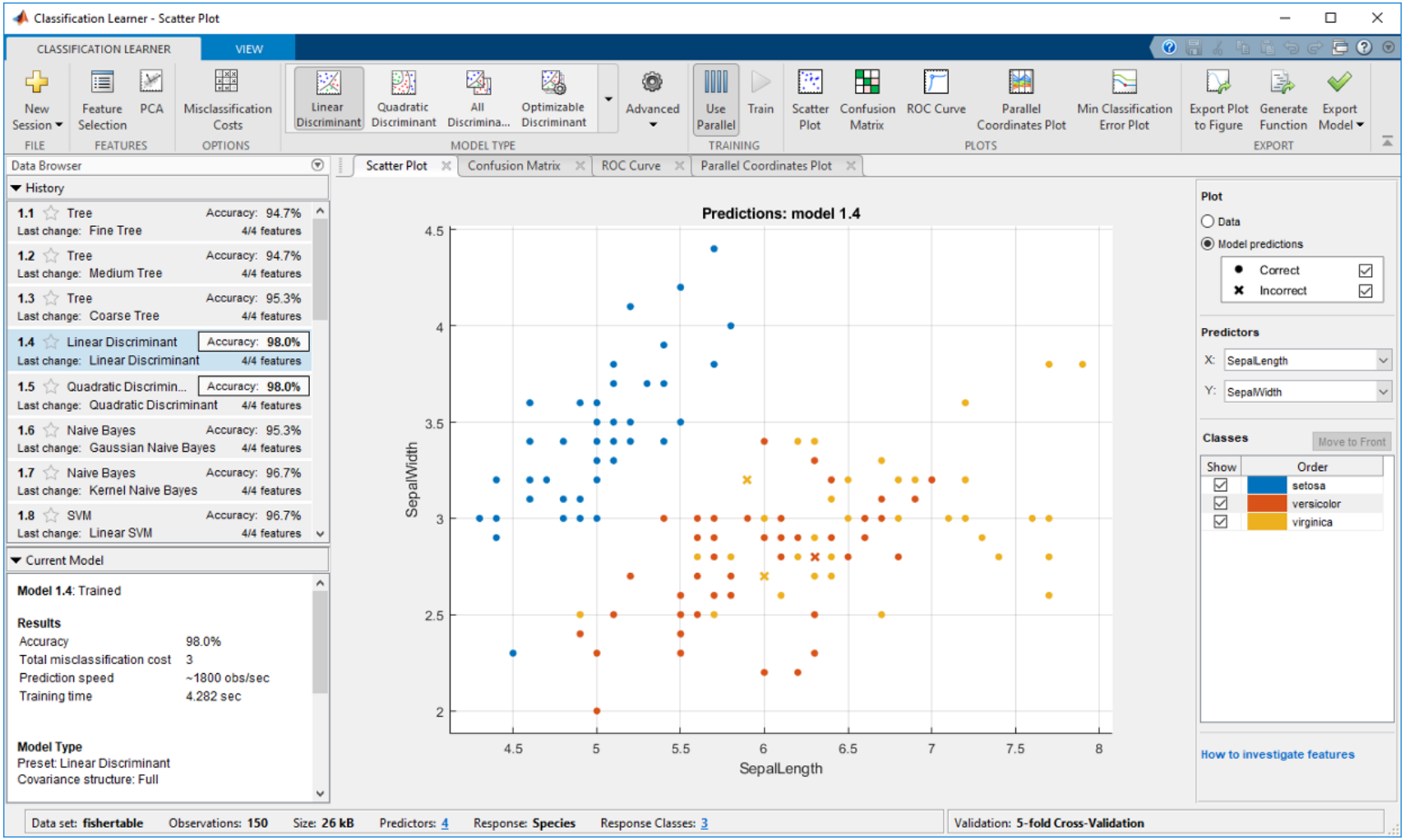
Classification Learner app can train models to classify data using supervised machine learning.

- Classifiers: **decision trees**, discriminant analysis, support vector machines, logistic regression, nearest neighbors, **naive Bayes**, and ensemble classification.
- Explore data, specify validation schemes, select features, and visualize results.
- Export models to make predictions with new data.
- Generate MATLAB code for further analysis.

Load Data


Classification Learner
CLASSIFICATION LEARNER
VIEW
New Session
Feature Selection
PCA
All Quick-To-Train
All Linear
Complex Tree
Medium Tree
Advanced
Use Parallel
Train
Scatter Plot
Confusion Matrix
ROC Curve
Parallel Coordinates Plot
Export Model
CLASSIFIER
TRAINING
PLOTS
EXPORT
From Workspace
Start a new session by selecting data from the workspace and specifying a validation scheme
From File
Start a new session by importing data from a file and specifying a validation scheme
Current model
Model number 0
Status: Queued
Classifier
No classifier type selected
Feature Selection
All features used in the model, before PCA
PCA
PCA disabled
Original Dataset:
Observations: 0
Predictors: 0
Response Variable:
Response Classes: 0
Size of Dataset: 0 B
Validation: No Validation

Results



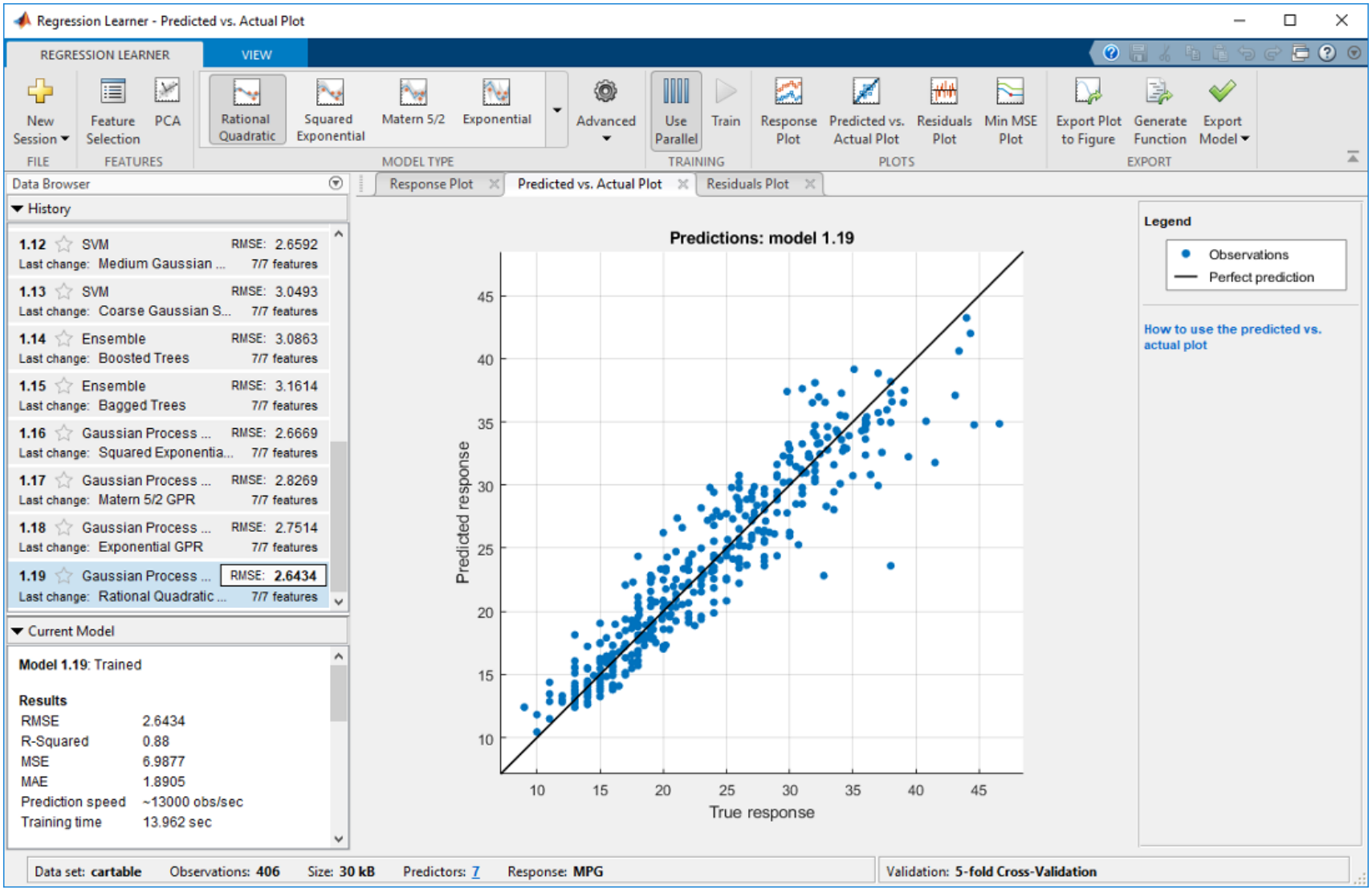
https://ww2.mathworks.cn/help/stats/train-decision-trees-in-classification-learner-app.html

Regression Learner app can train models to predict continuous data using supervised machine learning.

- Models: linear regression models, regression trees, Gaussian process regression models, support vector machines, and ensembles of regression trees.
- Explore data, specify validation schemes, select features, and visualize results.
- Export models to make predictions with new data.
- Generate MATLAB code for further analysis.

Statistics and Machine Learning Toolbox

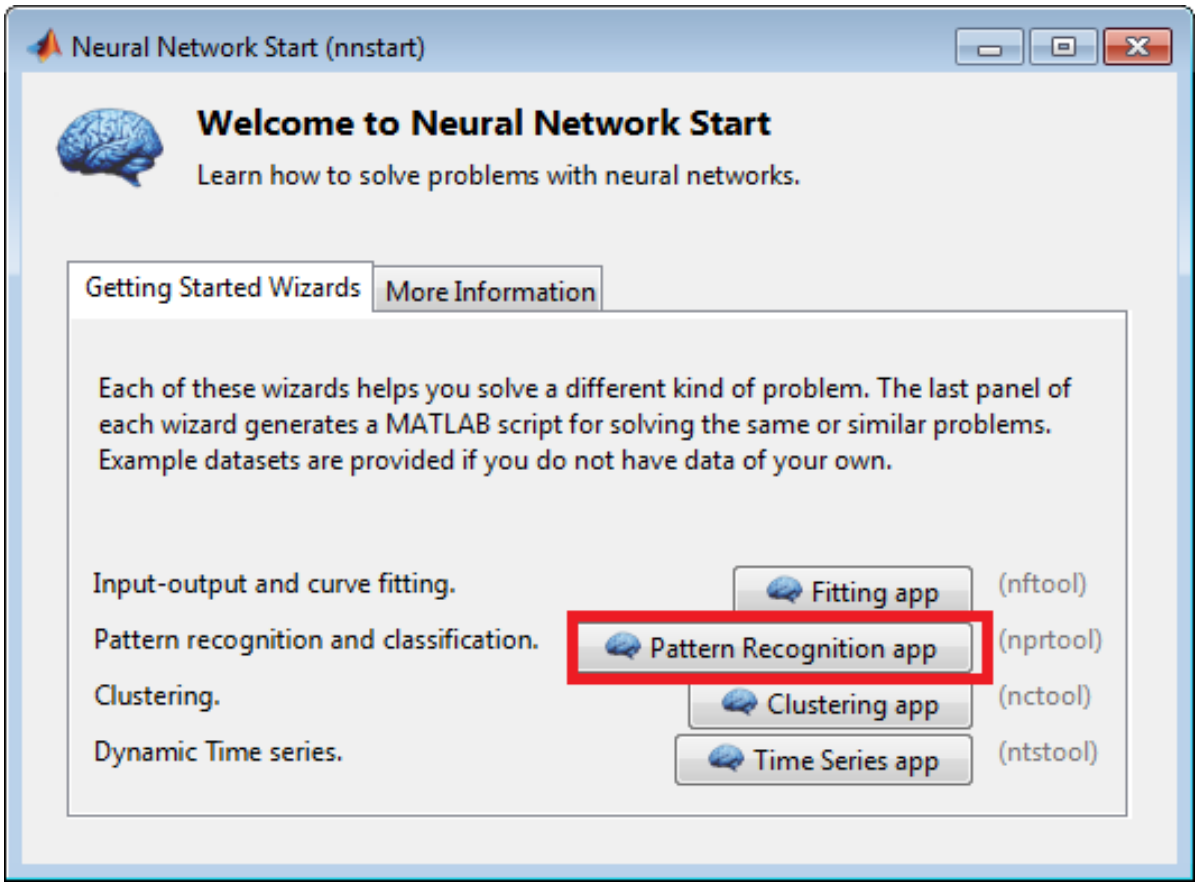
2. Train Regression Models in Regression Learner App (cont'd)



https://ww2.mathworks.cn/help/stats/train-regression-models-in-regression-learner-app.html

1. Introductions to Machine Learning and MATLAB
2. MATLAB Toolboxes for Machine Learning
   - Statistics and Machine Learning Toolbox
   - Deep Learning Toolbox
   - Reinforcement Learning Toolbox
3. Reference Materials

1. Open the **Neural Network Start GUI** with command: **nnstart**



2. Click **Pattern Recognition app** to open the **Neural Network Pattern Recognition app**. (or the command **nprtool**.)
3. Open the **Select Data window**.
4. Continue to the **Validation and Test Data window**.
5. Set the **number of hidden neurons** in the hidden layer.
6. **Train** the network.
7. **Evaluate** the network.

https://ww2.mathworks.cn/help/deeplearning/gs/classify-patterns-with-a-neural-network.html

Deep Learning

- Use **convolutional neural networks** (CNNs) and **long short-term memory** (LSTM) networks to perform classification and regression on image, time-series, and text data.
- Build advanced network architectures such as **generative adversarial networks** (GANs).
- Apps and plots can visualize **activations**, edit and analyze **network architectures**, and monitor **training progress**.
- Import models from **TensorFlow-Keras** and **Caffe**.
- Supports **transfer learning**.

Deep Learning

Create Simple Image Classification Network

1. Load Data

```
digitDatasetPath = fullfile(matlabroot,'toolbox','nnet','nndemos', ...
    'nndatasets','DigitDataset');

imds = imageDatastore(digitDatasetPath, ...
    'IncludeSubfolders',true, ...
    'LabelSource','foldernames');
```

```
numTrainFiles = 750;
[imdsTrain,imdsValidation] = splitEachLabel(imds,numTrainFiles,'randomize');
```

750 images

training set and validation set

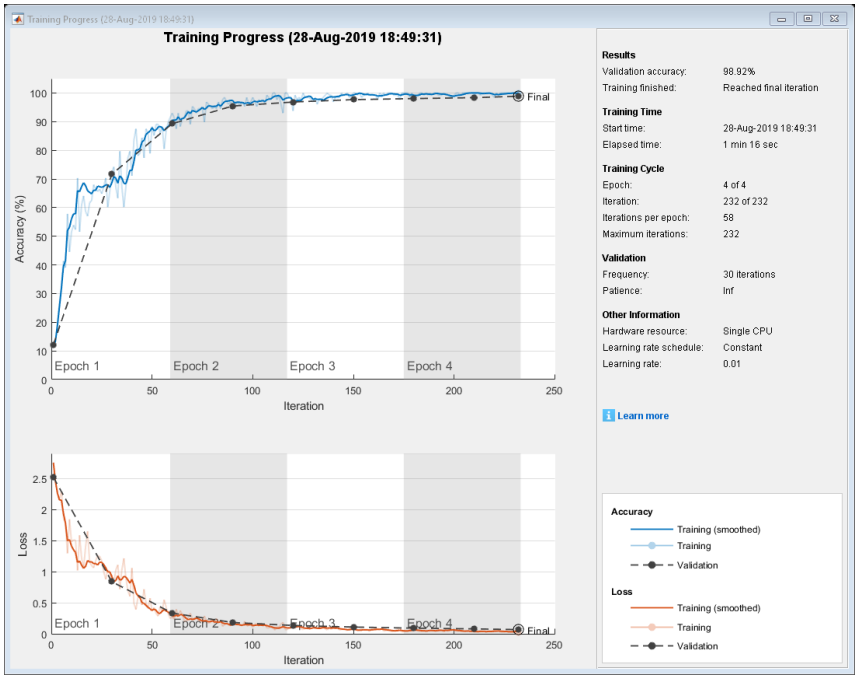
2. Define Network Architecture

```
inputSize = [28 28 1];
numClasses = 10;

layers = [
    imageInputLayer(inputSize)
    convolution2dLayer(5,20)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer];
```

Each image is 28-by-28 pixels

3. Train Network

```
options = trainingOptions('sgdm', ...
    'MaxEpochs',4, ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(imdsTrain,layers,options);
```



4. Test Network

```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = mean(YPred == YValidation)
```

https://ww2.mathworks.cn/help/deeplearning/gs/create-simple-deep-learning-classification-network.html

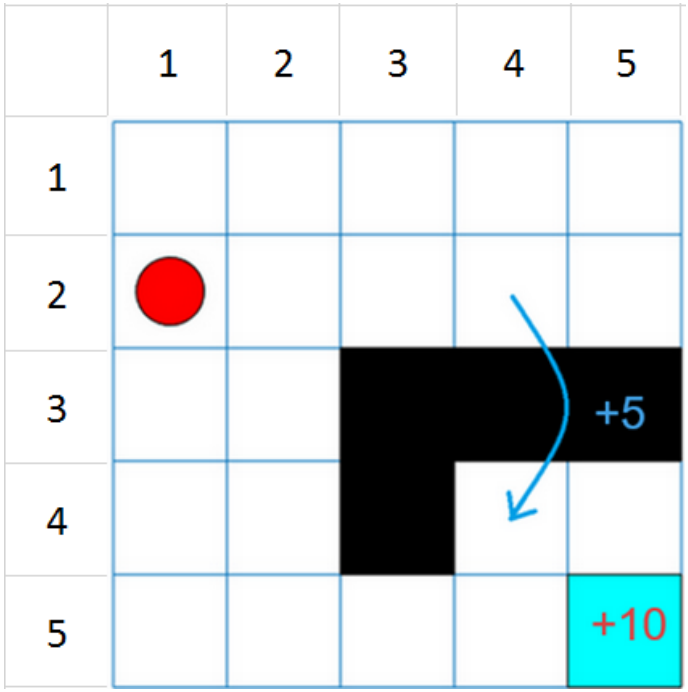
1. Introductions to Machine Learning and MATLAB
2. MATLAB Toolboxes for Machine Learning
   - Statistics and Machine Learning Toolbox
   - Deep Learning Toolbox
   - Reinforcement Learning Toolbox
3. Reference Materials

Reinforcement Learning Toolbox

- Functions for training policies using reinforcement learning algorithms including DQN, A2C, and DDPG.
- Train policies by enabling them to interact with environments represented by MATLAB or Simulink models.
- Evaluate algorithms, experiment with hyperparameter settings, and monitor training progress.
- Existing policies can be imported from deep learning frameworks such as TensorFlow, Keras and PyTorch.

1. Create Grid World Environment

```
env = rlPredefinedEnv("BasicGridWorld");
```



- starts at [2,1], ends at [5,5].
- 4 possible actions (North, South, East and West).
- blocked by black cells.
- a special jump from cell [2,4] to cell [4,4] with +5 reward.
- all other actions result in -1 reward.

https://ww2.mathworks.cn/help/reinforcement-learning/ug/train-q-learning-agent-to-solve-basic-grid-world.html

2. Create Q-Learning Agent

```
qTable = rlTable(getObservationInfo(env),getActionInfo(env));   creata Q table
tableRep = rlRepresentation(qTable);
tableRep.Options.LearnRate = 1;   Set the learn rate of the representation to 1
```

3. Train Q-Learning Agent

```
agentOpts = rlQAgentOptions;
agentOpts.EpsilonGreedyExploration.Epsilon = .04; configure the epsilon-greedy exploration
qAgent = rlQAgent(tableRep,agentOpts);                create a Q-learning  agent
```

4. Train Q-Learning Agent

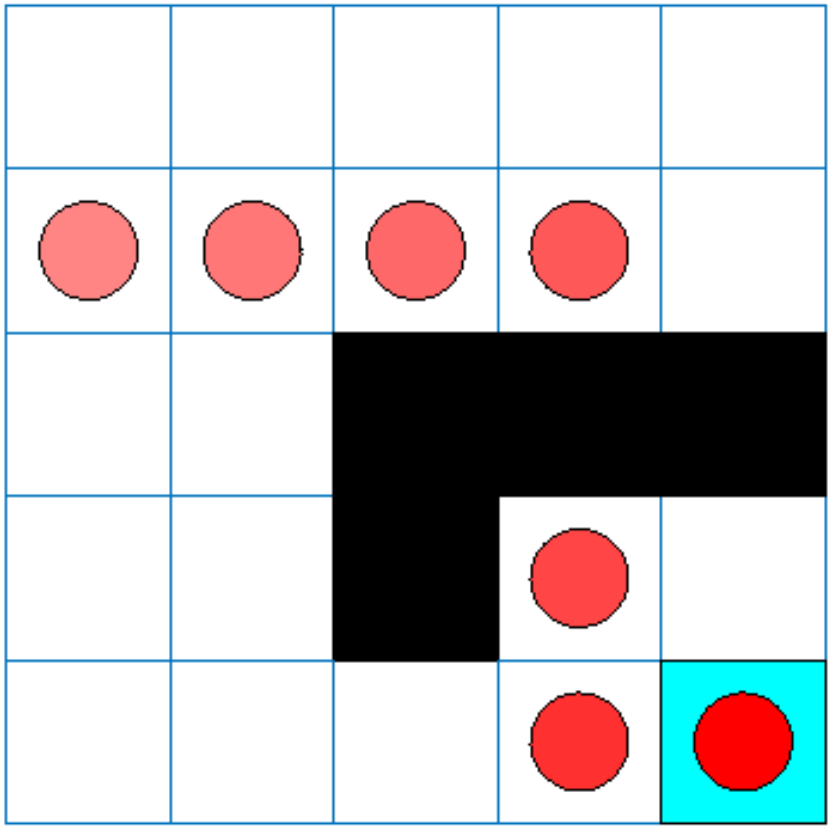
```
trainOpts = rlTrainingOptions;
trainOpts.MaxStepsPerEpisode = 50;
trainOpts.MaxEpisodes= 200;
trainOpts.StopTrainingCriteria = "AverageReward";
trainOpts.StopTrainingValue = 11;
trainOpts.ScoreAveragingWindowLength = 30;
```

5. Validate Q-Learning Results

```
plot(env)
env.Model.Viewer.ShowTrace = true;
env.Model.Viewer.clearTrace;
```

```
sim(qAgent,env)
```

Simulate



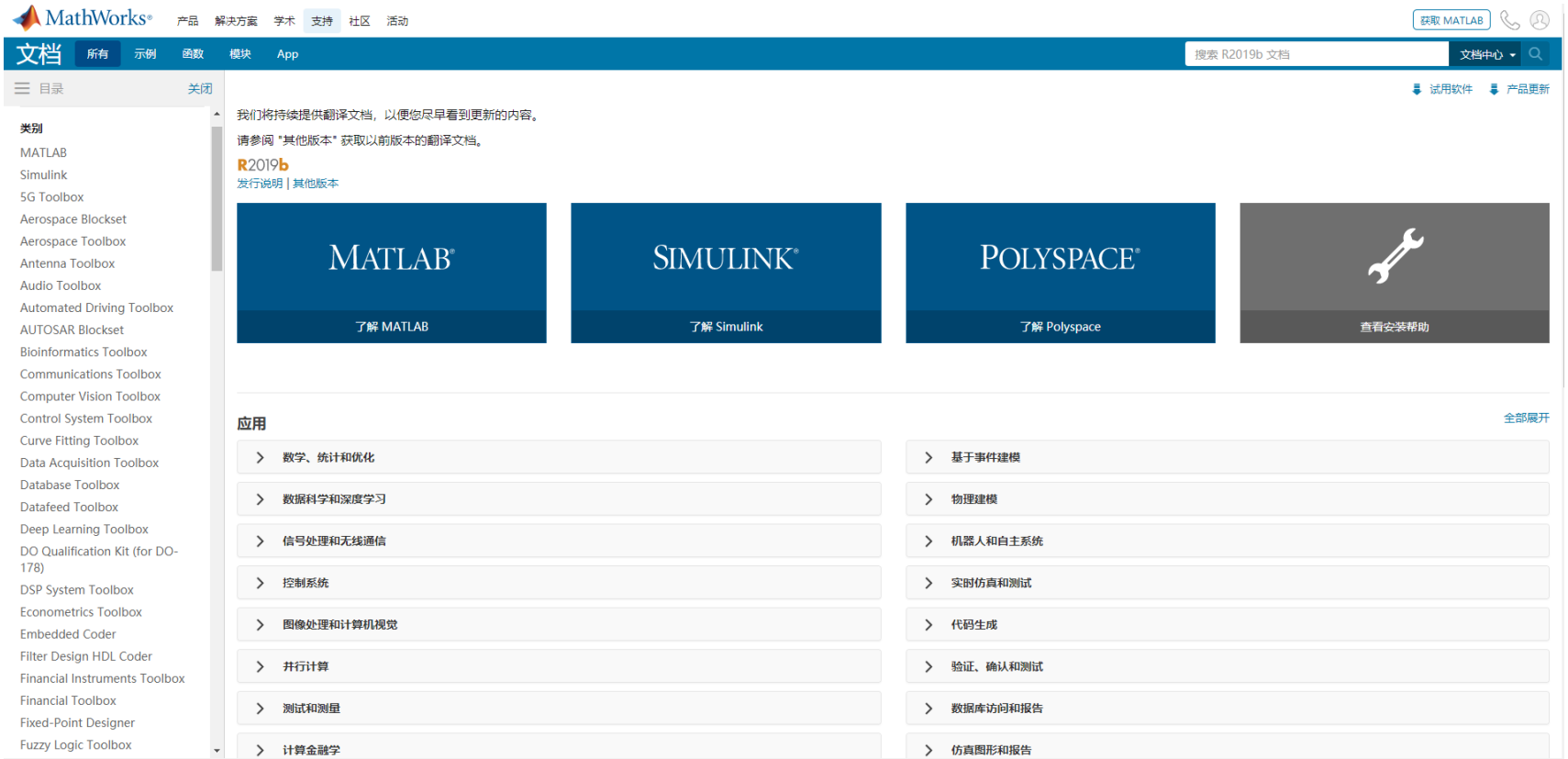
1. Introductions to Machine Learning and MATLAB

2. MATLAB Toolboxes for Machine Learning
   - Statistics and Machine Learning Toolbox
   - Deep Learning Toolbox
   - Reinforcement Learning Toolbox

3. Reference Materials

Reference Materials

★ ★ ★ https://ww2.mathworks.cn/help/index.html?s_tid=CRUX_lftnav



Reference Materials (cont'd)

⋆ https://blog.csdn.net/devil_bye/article/details/80536006

原创 **Coursera吴恩达机器学习笔记及代码练习(Matlab版)**

置顶 2018-06-01 12:20:31 不晓得X 阅读数 13144 ☆ 收藏 更多

coursera课程主页
https://www.coursera.org/learn/machine-learning

之前寒假其实已经在B站上看过Andrew的这门机器学习了，先在这里给出链接，基本上都是有中文字幕的，喜欢弹幕的小伙伴可以看这个。
https://www.bilibili.com/video/av9912938

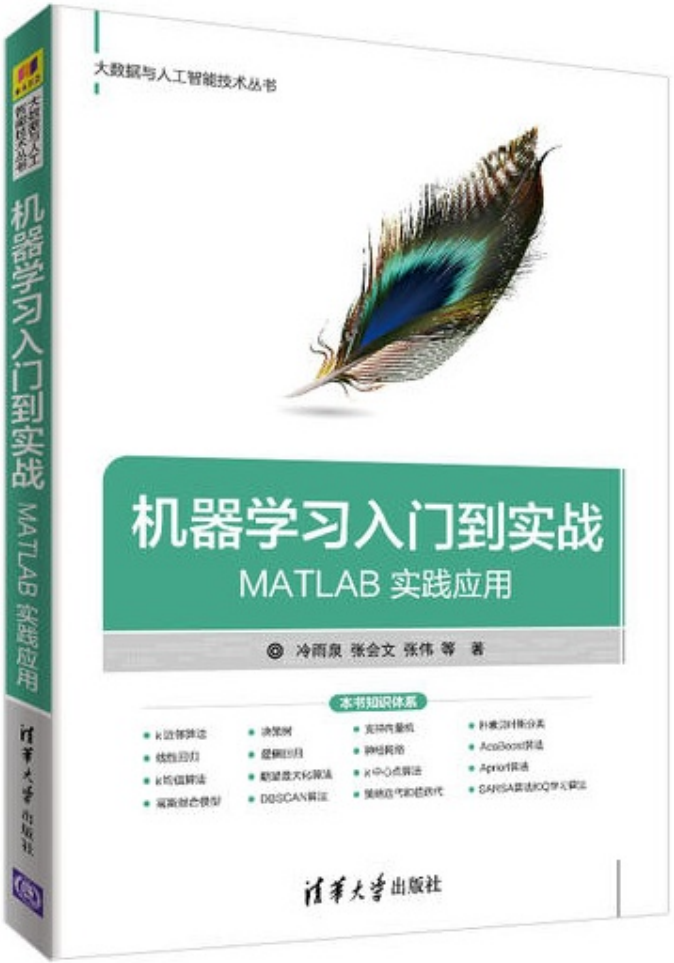
网易云课堂也上线了这门课，因为是官方翻译肯定比B站的好，缺点也是没有配套练习。
https://study.163.com/course/introduction/1004570029.htm

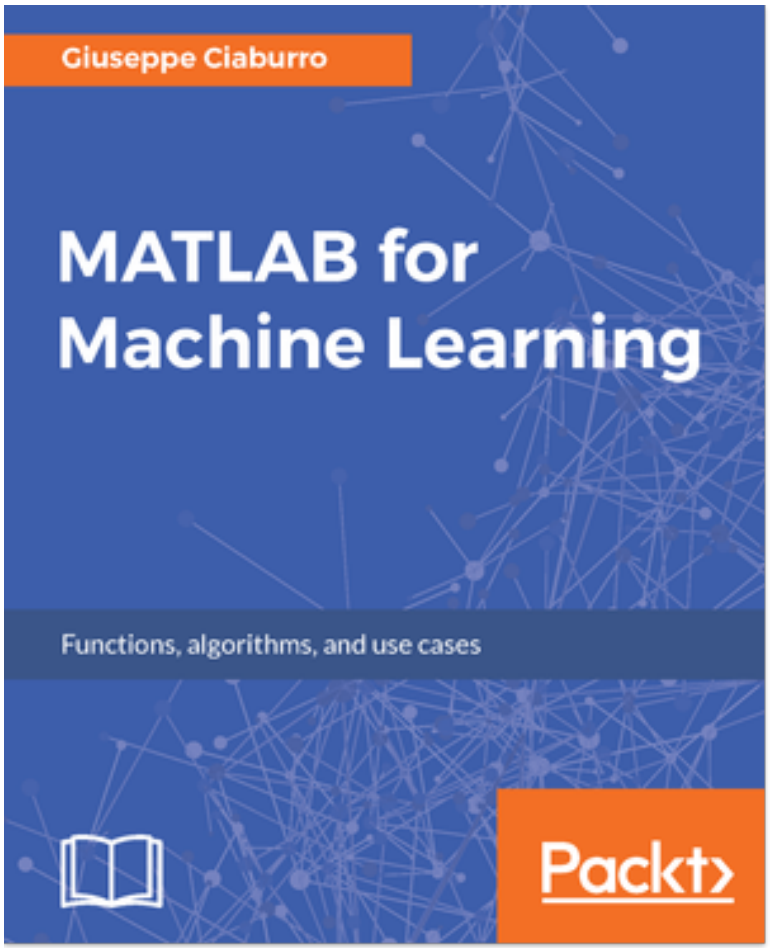
但是因为时间长了，有所遗忘，而且还是觉得配合Coursera上原本的练习和学习资源比较好，所以在最近又开始重看一遍，顺便做下练习。看网上很少能搜到代码练习的答案，所以在这里分享出来。老师推荐使用MATLAB或者octave，因为MATLAB可以在线使用，我就选择了使用MATLAB。之后可能还会用python再实现一遍，加深印象。

matlab版的代码实现已全部完成

github地址：https://github.com/xjwhhh/AndrewNgMachineLearning
欢迎follow和star

Reference Materials (cont'd)





The End