

クイックスタートガイド MATLAB による機械学習

学習器アプリを用いたモデル学習

ワークスペースやファイルからデータのインポート

主成分分析を有効化し次元の削減

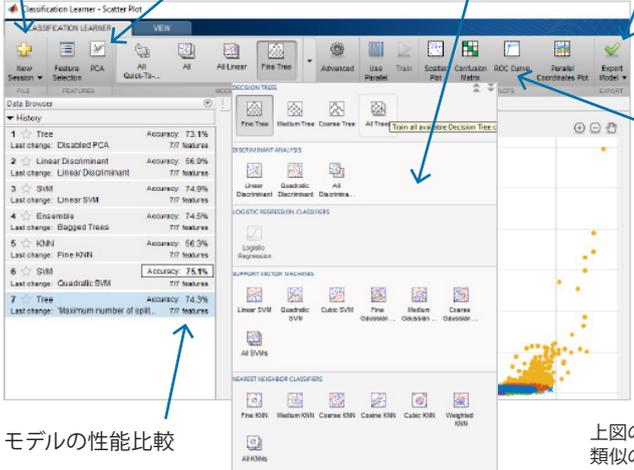
アルゴリズムの選択

更なる最適化や他コードとの統合のため、モデルのエクスポート

視覚的にモデルの性能を評価

モデルの性能比較

上図のアプリ: 分類学習器
類似のアプリ: 回帰学習器



機械学習関数

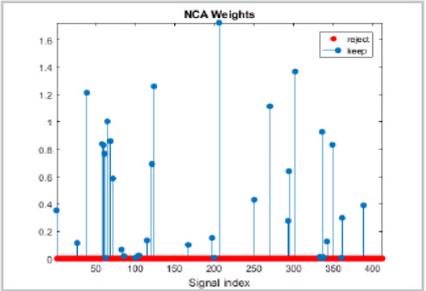
関数の命名規則
fit + c(classification) / r(egression) + model
例) SVM分類器 $m = \text{fitcsvm}(X,Y)$

アルゴリズム		モデル
決定木	c,r	tree
線形モデル	c,r	linear
サポートベクターマシン	c,r	svm
ガウスクーネル	c,r	kernel
アンサンブル (ランダムフォレストを含む)	c,r	ensemble
(一般)線形回帰		(g)lm
非線形回帰		nlm

特徴選択

近傍成分解析
有効な特徴量識別の自動化

```
fscnca(X labels, 'Lambda',...);  
find mdl.FeatureWeights > 0.01
```



こちらも利用可能です:

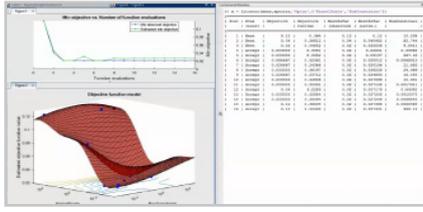
- 主成分分析
- ステップワイス回帰
- スパースフィルタ
- 再構成 ICA
- 行列の因数分解
- t-SNE

ハイパーパラメータ調整

パラメータ探索と変更をアプリで実行

ベイズ最適化による自動調整
ベイズモデルを活用し、ハイパーパラメータ空間のどの点を次に試すか選択。グリッド探索よりはるかに速い。

```
mdl = fit...(X,labels,  
'OptimizeHyperparameters','auto');
```

配布

スタンドアロン、Web アプリ、Spark
スタンドアロンアプリや MapReduce アプリ、Apache Spark™ アプリ、Web アプリ、Microsoft® Excel® のアドインの形で共有

エンタープライズ IT/OT との統合
MATLAB Compiler SDK™ を用いて C/C++ や Java®、.NET、Python® ライブラリに変換

C-コード生成
MATLAB Coder™ を用いて、組込み配布のために自動で C/C++コードへ変換

- モデルの学習** $Mdl = \text{fitcsvm}(X,Y);$
- saveCompactModel** ($Mdl, 'mySVM'$);
- エントリーポイント関数の定義**

```
function label = predictSVM(x)  
    m = loadCompactModel('mySVM');  
    label = predict(m,x);  
end
```
- Cコード生成**

```
codegen predictSVM -args {X}
```

詳細を見る: mathworks.com/machine-learning