scikit-learn ゼミ

1.2. Support Vector Machines
1.2.1. Classification
1.2.1.1. Multi-class classification
1.2.1.2. Scores and probabilities
1.2.1.3. Unbalanced problems

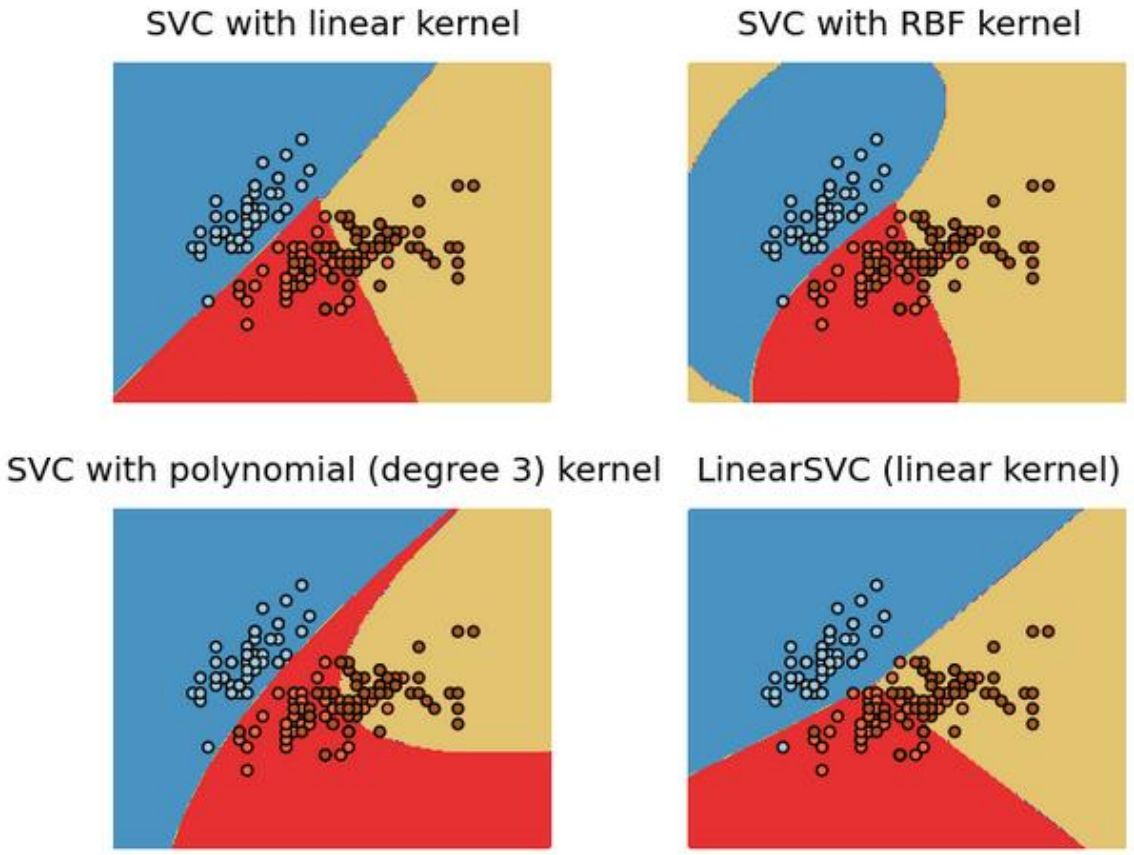
河野和平

Support Vector Machine

- 教師付き学習
- 分類，回帰，外れ値検出に使用される
- 特徴が高次元のとき効果的
  (次元数>サンプル数なら更に効果的)
- メモリを効率的に利用可能
- 様々なカーネル関数を決定関数に指定できる

Classification

- 多値分類を行うクラス(SVC,NuSVC,LinearSVC)



入力

- array X
  - 訓練サンプルを格納
  - $size: [n_{sample}, n_{features}]$

- array Y
  - 訓練サンプルのクラスラベルを格納(整数値)
  - $size: [n_{sample}]$

実行例

- 訓練データの入力

```
>>> from sklearn import svm
>>> X = [[0, 0], [1, 1]]
>>> y = [0, 1]
>>> clf = svm.SVC()
>>> clf.fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, degree=3,
gamma=0.0, kernel='rbf', max_iter=-1, probability=False, random_state=None,
shrinking=True, tol=0.001, verbose=False)
```

- 新しいデータの入力

```
>>> clf.predict([[2., 2.]])
array([1])
```

- サポートベクトル

```
>>> # get support vectors
>>> clf.support_vectors_
array([[ 0.,  0.],
       [ 1.,  1.]])
>>> # get indices of support vectors
>>> clf.support_
array([0, 1]...)
>>> # get number of support vectors for each class
>>> clf.n_support_
array([1, 1]...)
```

Multi-class classification

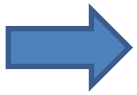
- one-against-one法
  - SVC,NuSVC
  - クラス数: n_{class}
  - 分離平面: $n_{class}\frac{n_{class}-1}{2}$

```
>>> X = [[0], [1], [2], [3]]
>>> Y = [0, 1, 2, 3]
>>> clf = svm.SVC()
>>> clf.fit(X, Y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, degree=3,
gamma=0.0, kernel='rbf', max_iter=-1, probability=False, random_state=None,
shrinking=True, tol=0.001, verbose=False)
>>> dec = clf.decision_function([[1]])
>>> dec.shape[1] # 4 classes: 4*3/2 = 6
6
```

Multi-class classification

- one-vs-the-rest法
  - LinearSVC
  - クラス数: n_{class}
  - 分離平面: n_{class}

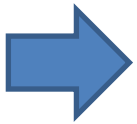
```
>>> lin_clf = svm.LinearSVC()
>>> lin_clf.fit(X, Y)
LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
intercept_scaling=1, loss='l2', multi_class='ovr', penalty='l2',
random_state=None, tol=0.0001, verbose=0)
>>> dec = lin_clf.decision_function([[1]])
>>> dec.shape[1]
4
```



Crammer and Singer によってほぼ同精度で
処理時間の短い手法が提案されている。

Scores and probabilities

- SVC
  - 決定関数はそれぞれのサンプルにスコアを与える
  - オプション
    - $probability = True$
    - そのクラスに属する確率を与える。
    - 2クラス分類 : Platt Scaling
    - 多クラス分類 : Platt Scaling のWuらによる拡張

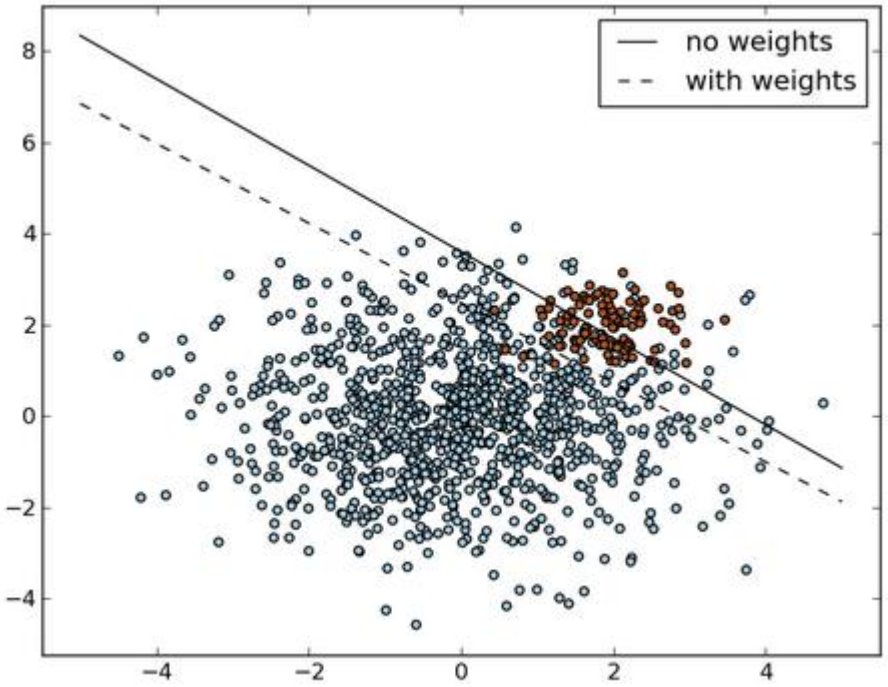


信頼性の高いスコアであれば必要ない

Unbalanced problems

- クラスラベル
  $\{class\_label: value\}$

$C \times value$



Unbalanced problems

- i 番目のサンプル

$$C \times sample_{weight}[i]$$