

# scikit-learn ゼミ

## 1.2.6. Kernel functions

### 1.2.6.1. Custom Kernels

1.2.6.1.1. Using Python functions as kernels

1.2.6.1.1. Using the Gram matrix

1.2.6.1.1. Parameters of the RBF Kernel

小野寺喜行

# Kernel functions

## カーネルの関数の種類

- ・線形カーネル:  $\langle x, x' \rangle$

線形のサポートベクタマシンと同じ  
大規模で疎なデータでよく用いられる

- ・多項式カーネル:  $(\gamma \langle x, x' \rangle + r)^d$

$d \rightarrow$  degree ,  $r \rightarrow$  coef0

画像を分類するときによく用いられる

# Kernel functions

## カーネルの関数の種類

- ・rbfカーネル:  $(-\gamma|x - x'|^2)$

$\gamma \rightarrow \text{gamma}, \gamma > 0$

データに関する事前知識がない場合に用いられる汎用的なカーネル

- ・シグモイドカーネル:  $\tanh(\gamma\langle x, x' \rangle + r)$

$r \rightarrow \text{coef0}$

# Kernel functions

## カーネルの設定

```
>>> linear_svc = svm.SVC(kernel='linear')
>>> linear_svc.kernel
'linear'
>>> rbf_svc = svm.SVC(kernel='rbf')
>>> rbf_svc.kernel
'rbf'
```

# Custom Kernels

Pythonの関数として与えられるカーネル  
やグラム行列を事前計算



独自のカスタムカーネルを定義できる

他の分類器と違う点

- `support_vectors_`: 値なし
- サポートベクタのインデックス → `support_`
- `fit()` の第1引数のリファレンスが保存される

# Using Python functions as kernels

- ・コンストラクタで kernel に関数を渡すことでカスタムカーネルの使用が可能
- ・カスタムカーネルは引数に2つの行列、第3の行列を返す

```
>>> import numpy as np
>>> from sklearn import svm
>>> def my_kernel(x, y):
...     return np.dot(x, y.T)
...
>>> clf = svm.SVC(kernel=my_kernel)
```

線形カーネルを定めて、  
そのカーネルを使用する分類器を作るコード

# Using the Gram matrix

```
>>> import numpy as np
>>> from sklearn import svm
>>> X = np.array([[0, 0], [1, 1]])
>>> y = [0, 1]
>>> clf = svm.SVC(kernel='precomputed')
>>> # linear kernel computation
>>> gram = np.dot(X, X.T)
>>> clf.fit(gram, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, degree=3,
gamma=0.0, kernel='precomputed', max_iter=-1, probability=False,
random_state=None, shrinking=True, tol=0.001, verbose=False)
>>> # predict on training examples
>>> clf.predict(gram)
array([0, 1])
```

- kernel='precomputed'とセット
- fitメソッドでXの代わりにグラム行列を渡す

# Parameters of the RBF Kernel

## パラメータ $c$

- ・大  $\rightarrow$  分類の正確さを優先
- ・小  $\rightarrow$  決定面の単純さを優先

## パラメータ $\gamma$

- ・単体のトレーニング例がどれ位の影響力を持つか

GridSearchCVを使い最適な $c$ と $\gamma$ の値を求める