

LIBSVM

サ ミンソン

# LIBSVM

- LibSVM—台湾国立大学のLinらによって作られたSVMのライブラリ
- サポートベクタ分類器 (C-SVC、nu-SVC)、回帰分析 (epsilon-SVR、nu-SVR)、分布評価 (1クラスSVM) のための統合ソフト
- svm-scale 、svm-train、svm-predict  
svm-scaleで訓練セットをスケーリングした後、svm-trainでmodelを構築し、svm-predictによって予測を行う

# データファイルの書式

- Libsvmは、データを訓練用と評価用の2つのデータファイルに分ける必要がある。
- クラスラベル 番号:数値 番号:数値 ...  
クラスラベル 番号:数値 番号:数値 ...  
...

# svm-scale

作成した訓練セットを引数に従ってスケールリングする

- -l (スケールリングの最小値の指定、デフォルト -1)
  - -u (スケールリングの最大値の指定、デフォルト 1)
  - -s savefile
  - -r restorefile
- 
- `svm-scale -s scaling_parameters traing_data > scaled_traing_data`
  - `svm-scale -r scaling_parameters test_data > scaled_test_data`

# svm-train

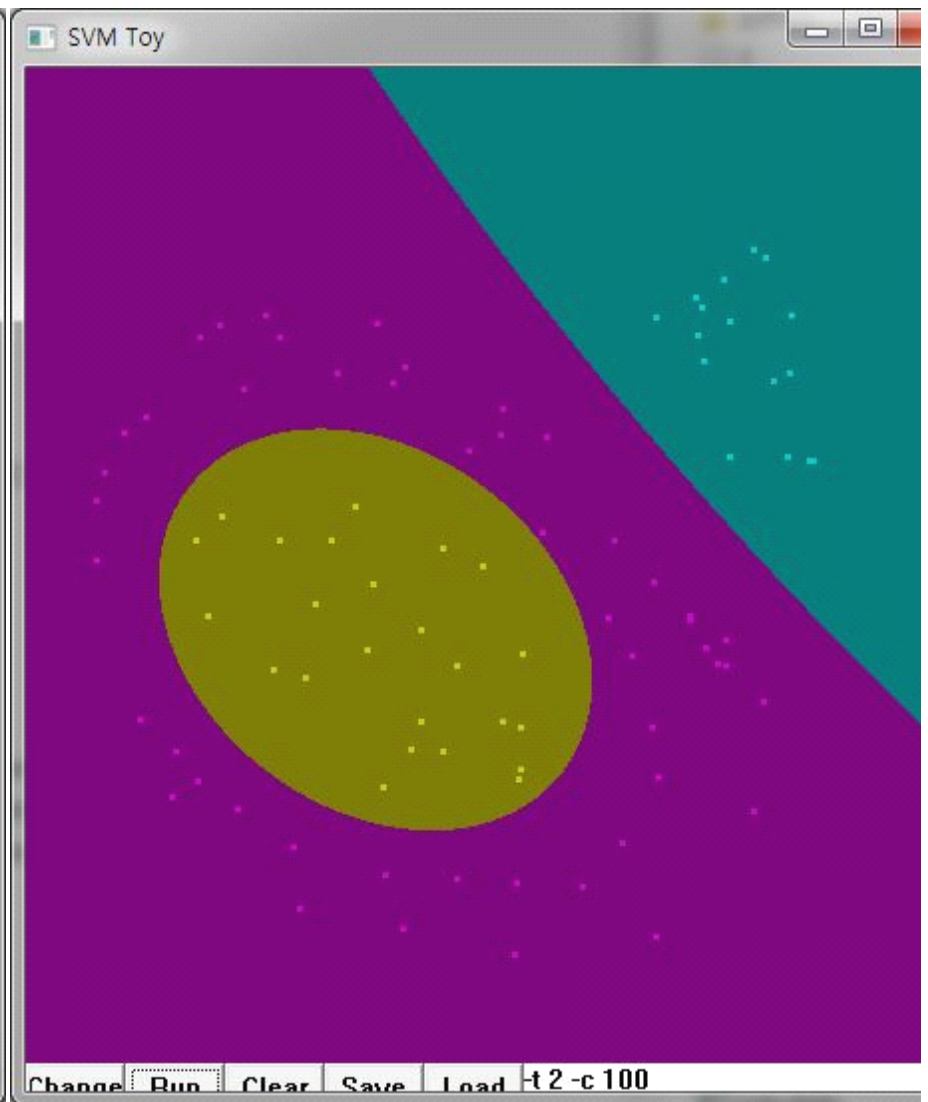
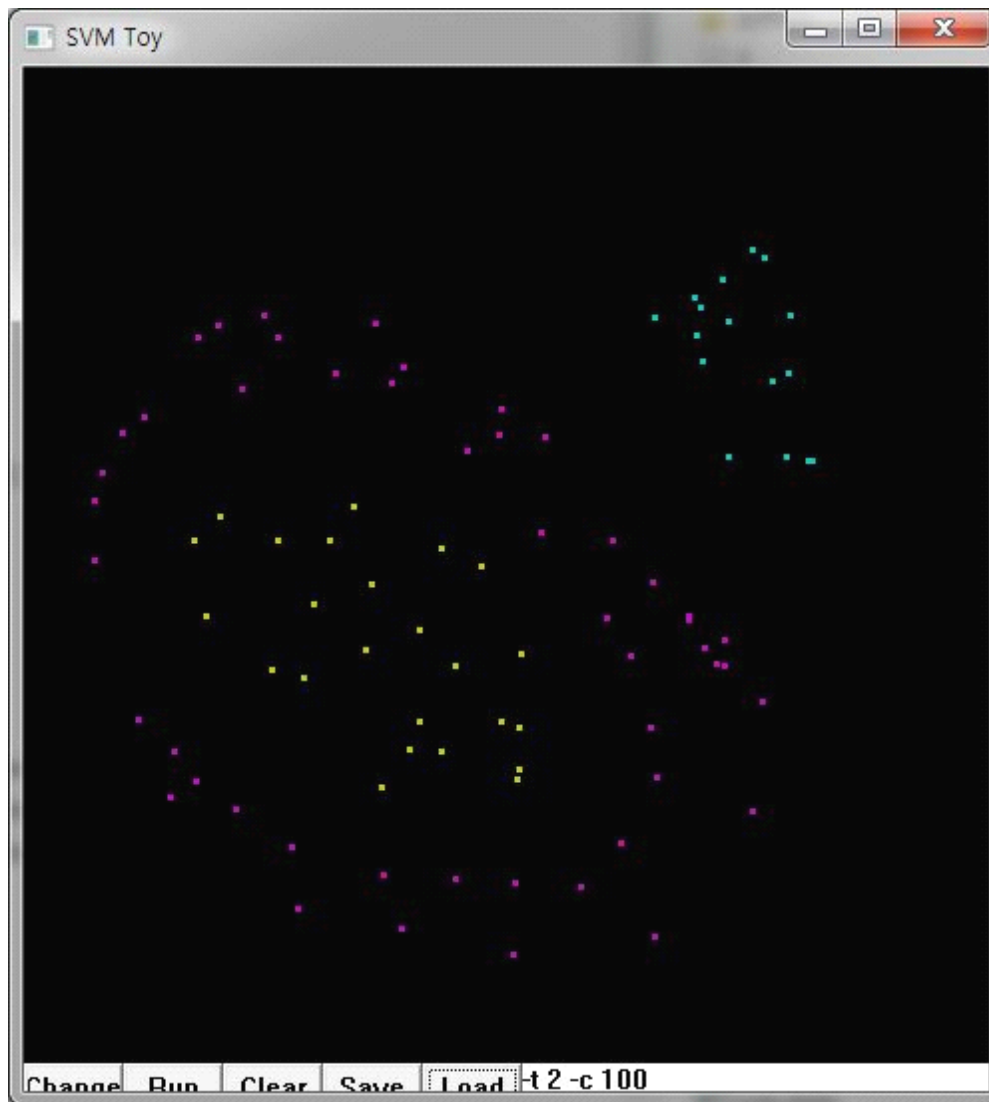
- 訓練セットから予測のためのモデルを生成
- `-s svm_type` : SVMタイプの指定 (デフォルト 0)
  - 0 -- C-SVC
  - 1 -- nu-SVC
  - 2 -- one-class SVM
  - 3 -- epsilon-SVR
  - 4 -- nu-SVR
- `-t kernel_type` : カーネル関数の指定 (デフォルト 2)
  - 0 -- 線形(linear):  $u \cdot v$
  - 1 -- 多項式(polynomial):  $(\gamma \cdot u \cdot v + \text{coef0})^{\text{degree}}$
  - 2 -- RBF(radial basis function):  $\exp(-\gamma \cdot |u-v|^2)$
  - 3 -- シグモイド(sigmoid):  $\tanh(\gamma \cdot u \cdot v + \text{coef0})$

# 続き

- `-d degree` : カーネル関数のdegreeの指定 (デフォルト 3)
- `-g gamma` : カーネル関数のgammaの指定 (デフォルト  $1/k$ ,  $k$ は入力ベクトルの次元)
- `-r coef0` : カーネル関数のcoef0の指定 (デフォルト 0)
- `-c cost` : コストパラメータの指定 (C-SVC, epsilon-SVR, and nu-SVRで使用、デフォルト 1)
- `-n nu` : nuパラメータの指定 (nu-SVC, one-class SVM, and nu-SVRで使用、デフォルト 0.5)
- `-m cachesize` : 使用キャッシュメモリサイズの指定(単位MB, デフォルト 40)
- `-e epsilon` : 終了の閾値の設定 (デフォルト 0.001)
- `-v n`: n-foldのクロスバリデーションを実行

# svm-predict

- 予測を実行
- `svm-predict test_file model_file output_file`



관리자: C:\Windows\system32\cmd.exe

```
C:\Users\juka\libsvm-2.9\windows>svm-train.exe -s 0 -c 10 -t 1 -g 1 -r 1 -d 3 dat003.txt dat003.model
```

```
...*.*
```

```
optimization finished, #iter = 504
```

```
nu = 0.569781
```

```
obj = -444.812920, rho = 6.003698
```

```
nSV = 72, nBSV = 65
```

```
*
```

```
optimization finished, #iter = 14
```

```
nu = 0.011811
```

```
obj = -3.836696, rho = -1.058451
```

```
nSV = 3, nBSV = 0
```

```
*
```

```
optimization finished, #iter = 18
```

```
nu = 0.025318
```

```
obj = -14.400749, rho = -2.310527
```

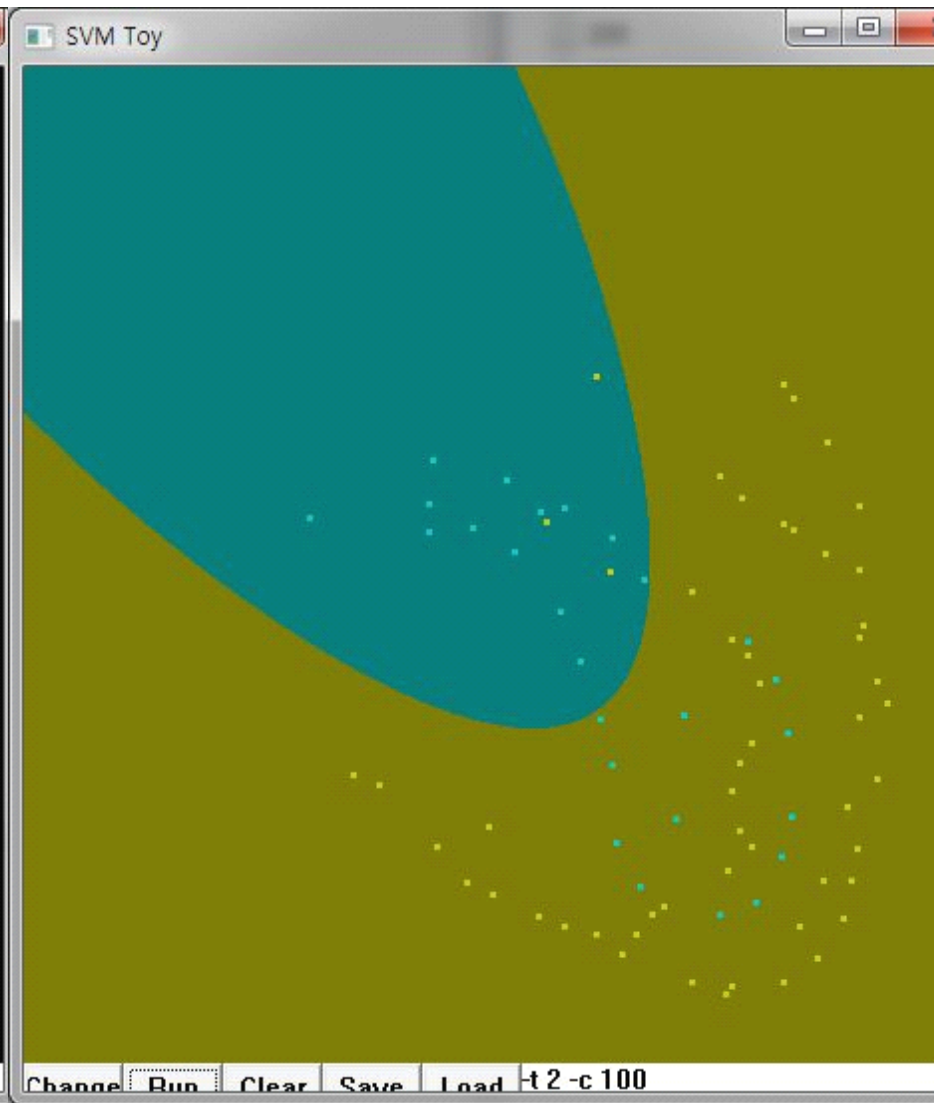
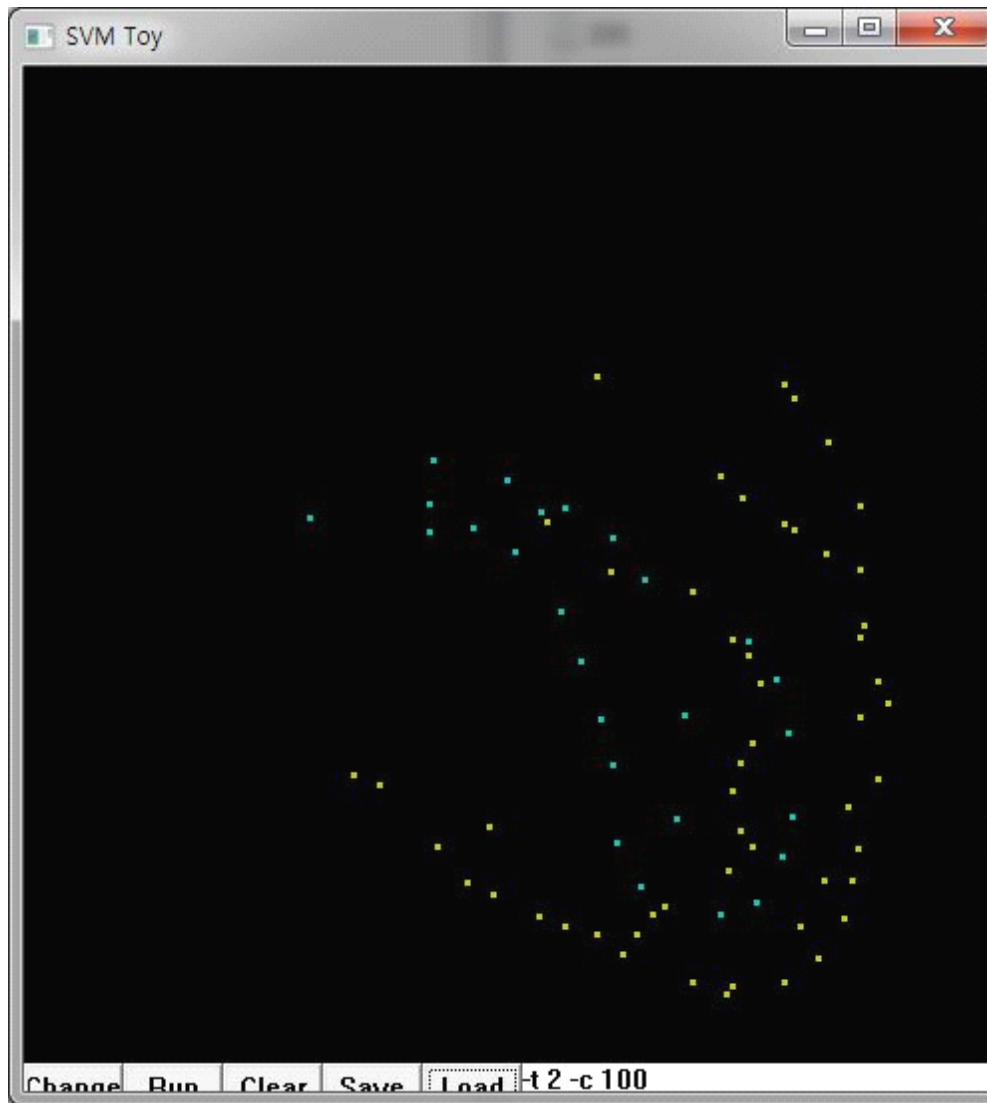
```
nSV = 4, nBSV = 1
```

```
Total nSV = 76
```

```
C:\Users\juka\libsvm-2.9\windows>svm-predict.exe dat003.txt dat003.model dat003.result
```

```
Accuracy = 100% (144/144) (classification)
```

```
C:\Users\juka\libsvm-2.9\windows>
```



관리자: C:\Windows\system32\cmd.exe

```
C:\Users\juka\libsvm-2.9\windows>svm-train.exe -s 0 -c 10 -t 1 -g 1 -r 1 -d 3 dat004.txt dat004.model
```

```
.*  
optimization finished, #iter = 236  
nu = 0.502020  
obj = -562.982138, rho = -4.062305  
nSV = 65, nBSV = 58  
Total nSV = 65
```

```
C:\Users\juka\libsvm-2.9\windows>svm-predict.exe dat004.txt dat004.model dat004.result
```

```
Accuracy = 79.5082% (97/122) (classification)
```

```
C:\Users\juka\libsvm-2.9\windows>
```

관리자: C:\Windows\system32\cmd.exe

```
2009-11-01 오전 06:26 137,728 svm-train.exe
2009-11-25 오전 01:28 8,160,653 train1.txt
                27개 파일 20,615,316 바이트
                3개 디렉터리 17,094,545,408 바이트 남음
```

```
C:\Users\juka\libsvm-2.9\windows>svm-train.exe -s 2 train1.txt train1.model
```

```
*
```

```
optimization finished, #iter = 95
obj = 10471.946242, rho = 140.329130
nSV = 157, nBSV = 143
```

```
C:\Users\juka\libsvm-2.9\windows>svm-predict.exe train1.txt train1.model train.
result
Accuracy = 52% (156/300) (classification)
```

```
C:\Users\juka\libsvm-2.9\windows>_
```