Support Vector Machine

S. Gadat

TSE -Université Toulouse I Capitole

Concerning the two first exercices, the useful files can be found here http://carlit.toulouse.inra.fr/wikiz/images/e/e5/Exos1+2B2.rar

1 Linear SVM

First toy example

We aim to better understand the behaviour of the Support Vector Machine algorithm. Let us recall that a linear SVM on $\mathbb{R}^p$ solves an optimisation problem of the form

$$\min_{\omega \ in \mathbb{R}^p} C \sum_{i=1}^{n} L(y_i, \omega.x_i + b) + \| \omega \|^2 . \tag{1}$$

Moreover, the vector ω has the following form $\omega = \sum_{i=1}^{n} \alpha_i x_i$, où $\alpha_i \neq 0$ if and only if x_i is a support vector.

1. Load the data with `datalin` (generate them with the R code) and the library `kernlab`.
2. Learn and visualize the linear SVM for several values of the constant C. With R:

```
svp<-ksvm(xtrain,ytrain,type="C-svc",kernel='vanilladot',C=??,scaled=c())
plotsvm(svp,xtrain)
```

3. Could you explain both terms involved in Equation (1)?
4. Explain and check experimentally the influence of the size of C on the separative hyperplane and the number of support vectors. What are the dashed lines?
5. What could happen with more mixed data (for example sampled according to a mixture of 2 Gaussian distributions with closed centers, or with a Gaussian that possesses a large variance)?

6. Visualise the test set and the classifier. Compute the predictions on the test set. (Help R: `predict(svp,xtest)`). How the prediction is computed on any unknown point x when the minimizer (1) ω^* is found?

Application aux données SPAM

1. Load the dataSPAM `dataC` with their associated labels.

2. Learn several linear SVM with different values of C.

3. Plot the misclassification error rate on the training set for several values of C. Do the same things for the misclassification error rate on the test set.

4. Explain the behaviour of the two errors.

5. How to estimate an optimal choice of C only on a training set? How could we estimate the ability of the classifier on a virtual test set?

2 SVM with kernel

We recall that a radial basis kernel is a Gaussian kernel defined as $(x, y) \in (\mathbb{R}^p)^2$ par : $K(x, y) = \exp\left(-\frac{\|x-y\|}{2\sigma^2}\right)$.

In a general settings, the SVM classifier with kernel has the following form:

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) + b.$$

1. Briefly recall the definition of a positive definite kernel.

2. In what situation these kernels may be useful?

3. Load the data `datamix` or generate them with the R code (use the library `kernlab`).

4. Learn and visualize a linear SVM on the data? Any remark? Help with R:

   ```
   svp <- ksvm(xtrain,ytrain,type="C-svc",kernel='rbf',kpar=list(sigma=1),C=1)
   ```

5. Use a radial basis gaussian kernel, any remark?

6. How the separating hyperplane is built?

7. Describe the influence of C and σ on the separating region and on the support vectors.

3 Real application on the Ozone dataset

We will study the ozone dataset. We aim to predict an ozone rush when see atmospheric or meteorological conidtions with SVM. Load the library `e1071`.

The main principle is to look for a sparse research of supporting points. It is then important to understand what parameters influence the SVM:

- Choice of the regularization parameter or fit to the data.
- Kernel parameters (bandwidth kernel, degree, ...)

3.1 Scholar example

The dataset `iris` is commonly used for scholar purpose. 3 categories of plants are to be discriminated (*sétosa, versicolor, virginica*). We observe 4 variables: (length and width of sepal and petal). There exists 50 samples for each categorie.

```
library(e1071)
# declaration des donnees
data(iris)
# Compute the model with default parameters
# (Gaussian kernel, penalization 1, gamma=0.25)
model = svm(Species ~ ., data = iris)
print(model)
summary(model)
# prediction on the learning set
pred = predict(model, iris[,1:4])
# Confusion matrix on the learning set
table(pred, iris$Species)
# Visualization of the categories (coloers) and support vectors
("+")
plot(cmdscale(dist(iris[,-5])),col = as.integer(iris[,5]),pch =
c("o","+")[1:150 %in% model$index + 1])
```

Remark the density of supporting points near the frontier (more difficult to discriminate). These observations are very important and hardly influence the prediction and the margin.
We can automatically set up several parameters with the R function `tune()`.

```
obj = tune.svm(Species~., data = iris,gamma = 2^(-7:0), cost = 2^(-2:3))
summary(obj)
plot(obj)
```

Regression on the ozone concentration

We will restrict our choice to the Gaussian kernel. The function `tune.svm()` allows to test several different situations while estimating the accuracy of prediction (using a cross validation procedure for example). Such a procedure could be quite long from a computational point of view. The computational cost of the SVM increases more than linearly with the number of observations but not so much with the number of variables. It is a kind of *Big Data* procedure!

The dataset comes from MétéoFrance and contains the following variables:

- `JOUR`: le type de jour ; férié (1) ou pas (0),
- `O3obs`: la concentration d'ozone effectivement observée le lendemain à 17h locales correspondant souvent au maximum de pollution observée,
- `MOCAGE`: prévision de cette pollution obtenue par un modèle déterministe de mécanique des fluides (équation de Navier-Stockes),
- `TEMPE`: température prévue par MétéoFrance pour le lendemain 17h,
- `RMH2O`: rapport d'humidité,
- `NO2`: concentration en dioxyde d'azote,
- `NO`: concentration en monoxyde d'azote,
- `STATION`: lieu de l'observation : Aix-en-Provence, Rambouillet, Munchhausen, Cadarache ou Plan de Cuques,
- `VentMOD`: force du vent et
- `VentANG`: orientation du vent

1. Load the data `ozone.dat` (available at `http://carlit.toulouse.inra.fr/wikiz/images/0/02/Ozone.dat.rar`. Convert the variable `JOUR` as a factor. Transform with a square the variable `RMH2O` et and transform with a log the variables `NO2` and `NO` (why?). Omit the initial variables

2. Split the dataset with `datappr` (used to learn) and `datestr` (used to test). Approximate effective sizes: 80% and 20%.

3. Initialy developed to deal with classification problem, SVM has been extended to regression ones. We can estimate the optimal penalization with:

```
svm.reg=svm(O3obs∼.,data=datappr)
plot(tune.svm(O3obs∼.,data=datappr, cost=c(1, 1.5,2,2.5,3,3.5)))
```

4. A default value is set to 1. Compute the optimal penalization with the Gaussian kernel and learn this optimal model. Plot the residuals.

Discrimination

We now study a classification problem (presence of an ozone rush). Create from the variable `O3obs` a binary variable that thesholds the level 150 $\mu g.m^{-3}$ (0 or 1) and call this new variable `DepSeuil`. Split again the dataset in two parts `datappq` and `datestq` (omit `O3obs` for the last one).

```
# optimisation
plot(tune.svm(DepSeuil~.,data=datappq, cost=c(1,1.25,1.5,1.75,2)))
# apprentissage
svm.dis=svm(DepSeuil~.,data=datappq,cost=1.25)
```

Compute the misclassification error.

```
# matrice de confusion
table(svm.dis$fitted,datappq$DepSeuil)
```

Prediction on the test set

Using the "best" parameters found above, compute the error rate on the test set (quadratic for the regression, misclassification)