Machine Learning for biology

V. Monbet

UFR de Mathématiques
Université de Rennes 1

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Support Vector Machines (SVM)

- Support Vector Machines are classifiers i.e. they predict a qualitative variable, typically $Y \in \{-1, 1\}$.
- SVM combine 2 tricks.
  1. It is a kernel method.
  2. It is a large margin linear classifier (in the representation space $\mathcal{F}$).

- Remind that when $Y \in \{-1, 1\}$ and $g(x)$ is a classifier, $yg(x) > 0$ if the sample x is correctly classified by g.
- Remark, that if $f(x) = \beta_0 + x\beta$ is a linear frontier betwenn the classes, $yf(x) > 0$ also means a correct classification.

Linear classifier with large margin

- $\mathbf{x} \in \mathcal{X}$, $y \in \{-1, 1\}$, data: $\mathcal{S}_n = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$
- If the **classes are separable**, the problem is to find an hyperplan

$$f(\mathbf{x}) = \beta_0 + \mathbf{x}\boldsymbol{\beta}$$

such that the margin M is the largest

$$\max_{\beta_0, \boldsymbol{\beta}, \|\boldsymbol{\beta}\|=1} M \text{ under constraints } y_i(\beta_0 + \mathbf{x}_i\boldsymbol{\beta}) \geq M, i = 1, \cdots, n$$

The constraint $\|\beta\| = 1$ can be taken into account by writing $y_i(\beta_0 + \mathbf{x}_i\beta) \geq M\|\beta\|$.

- And with $M = 1/\|\beta\|$,

$$\min_{\beta_0, \beta} \frac{1}{2}\|\beta\|^2 \text{ under constraints } y_i(\beta_0 + \mathbf{x}_i\beta) \geq 1, i = 1, \cdots, n$$

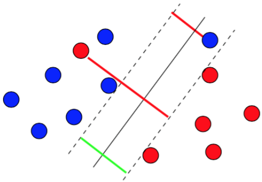
- The constraint implies that all the points are well classified.
Note that, for a regression problem the constraint is substituted by $y_i - (\beta_0 + \mathbf{x}_i\boldsymbol{\beta}) \leq M$ *and* $-y_i + (\beta_0 + \mathbf{x}_i\boldsymbol{\beta}) \leq M$.

Linear classifier with large margin

- The large margin classification problem find a trade off between large margin and a few errors

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{\mathit{margin}(\beta_0, \boldsymbol{\beta})} + C \times \mathit{errors}(\beta_0, \boldsymbol{\beta})$$

- C is a regularization parameter. When C tends to infinity, no error is allowed.



Soft margin SVM formulation

- The margin of a labelled point $(\mathbf{x}, y)$ is defined by

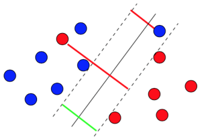
$$\text{margin}(\mathbf{x}, y) = y(\beta_0 + \mathbf{x}\boldsymbol{\beta})$$

- The error is

$$\begin{array}{ll} 0 & \text{if } \text{margin}(\mathbf{x}, y) > 0, \\ 1 - \text{margin}(\mathbf{x}, y) & \text{otherwise.} \end{array}$$

- The soft margin SVM solves

$$\min_{\beta_0, \boldsymbol{\beta}} \left\{ ||\boldsymbol{\beta}||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i(\beta_0 + \mathbf{x}_i\boldsymbol{\beta})) \right\}$$



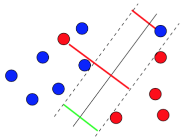
Soft margin SVM formulation

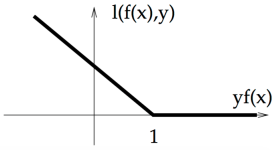
- With the hinge loss function

$$\ell_{\text{hinge}}(u, y) = \max(1 - yu, 0) = \begin{cases} 0 & \text{if } yu \geq 1 \\ 1 - yu & \text{otherwise} \end{cases}$$

and $\lambda = 1/C$, problem is rewritten

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} \ell_{\text{hinge}}(\beta_0 + \mathbf{x}_i \boldsymbol{\beta}, y_i) + \lambda ||\boldsymbol{\beta}||^2$$





Lagrangian formulation

- Find $(\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}$ which solves

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} ||\boldsymbol{\beta}||^2$$

  s.t. $(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) y_i \geq 1, \; i = 1, \cdots, n$

- It is equivalent to looking for the lagrangian saddle point

$$\max_{\alpha} \min_{\beta_0, \boldsymbol{\beta}} \mathcal{L}(\beta_0, \boldsymbol{\beta}, \alpha)$$

  where $\alpha_i \geq 0$ are the Lagrange multipliers and

$$\mathcal{L}(\beta_0, \boldsymbol{\beta}, \alpha) = \frac{1}{2} ||\boldsymbol{\beta}||^2 - \sum_{i=1}^{n} \alpha_i \left((\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) y_i - 1 \right)$$

α_i represents the influence of the constraint linked to point $\mathbf{x}_i$ thus the influence of point $\mathbf{x}_i$.

Gradients

$$\mathcal{L}(\beta_0, \boldsymbol{\beta}, \alpha) = \frac{1}{2}||\boldsymbol{\beta}||^2 - \sum_{i=1}^{n} \alpha_i \left((\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) y_i - 1\right)$$

Computing the gradients

$$\nabla_{\boldsymbol{\beta}} \mathcal{L}(\beta_0, \boldsymbol{\beta}, \alpha) = \boldsymbol{\beta} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial \mathcal{L}(\beta_0, \boldsymbol{\beta}, \alpha)}{\partial \beta_0} = - \sum_{i=1}^{n} \alpha_i y_i$$

When the gradients are 0, we have

$$\boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Conditions for SVM

$$\boldsymbol{\beta} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) \geq 1, \quad i = 1, \cdots, n$$

$$\alpha_i \geq 0, \quad i = 1, \cdots, n$$

$$\alpha_i \left(y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) - 1 \right) = 0, \quad i = 1, \cdots, n$$

The last condition (called the complementary condition) split the data into two sets

- The set of active constraints (usefull points)

$$\left\{ i \in \{1, \cdots, n\} | y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) = 1 \right\}$$

  That are the points which are effectively used in the calculus.

- The set of useless points

$$\{ i \in \{1, \cdots, n\} | \alpha_i = 0 \}$$

  They correspond to well classified points and are not involved in the calculus.

- SVM formulation with $\boldsymbol{\beta}$

$$\max_{\boldsymbol{\beta},\beta_0,\alpha} \frac{1}{2}||\boldsymbol{\beta}||^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) - 1 \right)$$

with $\alpha_i \geq 0$ and $\boldsymbol{\beta} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0$ and $\sum_{i=1}^n \alpha_i y_i = 0$

- Now, using the fact that $\boldsymbol{\beta} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$, we obtain a formulation without $\boldsymbol{\beta}$

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i y_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

with $\alpha_i \geq 0$ and $\sum_{i=1}^n \alpha_i y_i = 0$.
It is a quadratic problem too.

- Predict with the decision function

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + \beta_0$$

SVM in the features space

In the features space, a kernel replaces the inner products.

- Train the SVM by maximizes

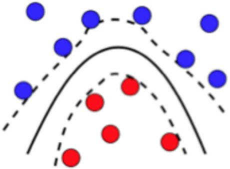
$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i y_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$$

under the constraints

$$0 \le \alpha_i y_i \le C, \quad \text{for } i = 1, \cdots, n$$

predict with the decision function

$$f(x) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \beta_0$$



- The training points with $\alpha_i \neq 0$ are called support vectors.
  Only support vectors are important for the classification of new points:

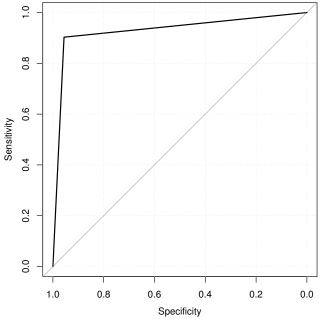
$$f(x) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \beta_0 = f(x) = \sum_{i \in SV} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \beta_0$$

- SVM leads to very flexible classifiers.
- Parameter C drives the regularization. It has to be chosen by the user.
- The strength of SVM in high dimension ($p > n$) is that it solves a convex problem only for the support vectors.

- In Support Vector Regression (SVR) similar ideas are used.
  Algorithm hyper parameters : kernel and its parameters, C.

Example for SVM: Leukemia

- Gaussian kernel
- Algorithm parameter to choose: C, γ



Outline

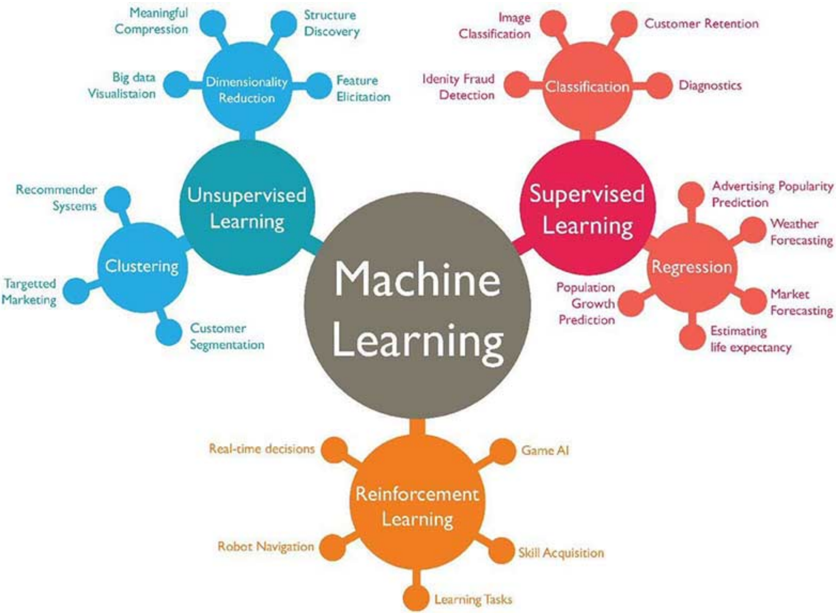
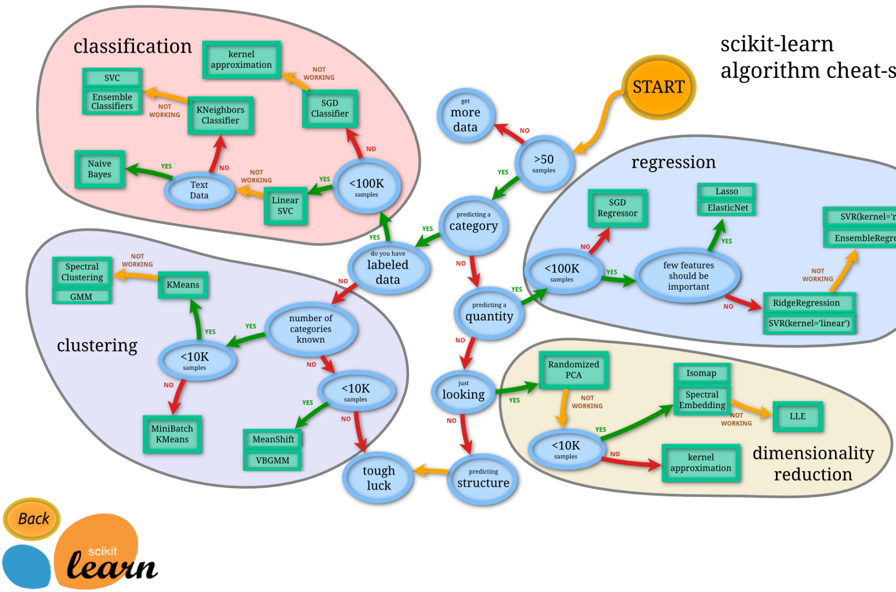


image from a blog...


scikit-learn
algorithm cheat-s
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
get more data
NO
>50 samples
YES
predicting a category
YES
YES
do you have labeled data
NO
NOT WORKING
Spectral Clustering
GMM
KMeans
clustering
YES
YES
<10K samples
number of categories known
NO
NO
MiniBatch KMeans
<10K samples
YES
MeanShift
VBGMM
NO
tough luck
predicting structure
NO
predicting a quantity
YES
NO
just looking
YES
NO
regression
SGD Regressor
Lasso
ElasticNet
NO
YES
<100K samples
YES
few features should be important
SVR(kernel='r
EnsembleRegre
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
Back
scikit learn