Intelligent Systems: Reasoning and Recognition

James L. Crowley

Ensimag 2 — Winter Semester 2018-2019
Lesson 5 — 20 February 2018

Support Vector Machines

Outline

Notation

x_d	A feature. An observed or measured value.
$\vec{X}$	A vector of D features.
D	The number of dimensions for the vector $\vec{X}$
$\{\vec{X}_m\}$	Training samples for learning.
$\{y_m\}$	The indicator variable for each training sample, $y_m = +1$ for examples of the target pattern (class 1) $y_m = -1$ for all other examples (class 2)
M	The number of training samples.
$\vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{pmatrix}$	The coefficients for a linear model the model.
b	bias so that $\vec{w}^T\vec{X}_m + b \geq 0$ for Class 1 and $\vec{w}^T\vec{X}_m + b < 0$ for Class 2.

IF $\vec{w}^T\vec{X}_m + b \geq 0$ THEN P ELSE N

IF $y_m\left(\vec{w}^T\vec{X}_m + b\right) \geq 0$ THEN T ELSE F

$\gamma = \min\left\{y_m \cdot \left(\vec{w}^T\vec{X}_m + b\right)\right\}$ Margin for a classifier

Classifier Margin for a linear classifier

For a 2-Class classifier, the "margin", γ, is the smallest separation between the two classes.

Consider a linear classifier is defined by a vector of weights, a bias and a decision function a decision function of sgn($\vec{w}^T\vec{X}_m + b$):

$$\vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{pmatrix} \text{ and } b \qquad \text{sgn}(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0 \end{cases}$$

This is equivalent to IF $\vec{w}^T\vec{X} + b \geq 0$ THEN Class 1 ELSE Class 2.

Note that $\vec{w}^T\vec{X}_m + b \geq 0$ is the same as $\vec{w}^T\vec{X}_m \geq -b$.
Thus b can be considered as a threshold on the product : $\vec{w}^T\vec{X}_m$

Assume a training set of M observations $\{\vec{X}_m\}$ $\{y_m\}$ where
$y_m = +1$ for examples of the target class (class 1)
$y_m = -1$ for all others (class 2)

The training data is **separable** if there exists a decision surfaces $\vec{w}$ with bias b, such that for all training data, $\vec{X}_m$,

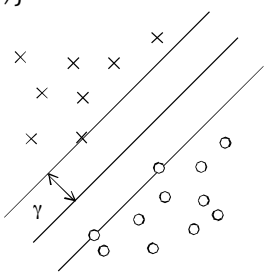
$\vec{w}^T\vec{X}_m + b \geq 0$ for Class 1 and $\vec{w}^T\vec{X}_m + w_0 < 0$ for Class 2.

A training sample is correctly classified if: $y_m \cdot \left(\vec{w}^T\vec{X}_m + b\right) \geq 0$

Classifier Margin

The margin, γ_m, for each sample, m, is $\gamma_m = y_m \cdot \left(\vec{w}^T\vec{x}_m + b\right)$

The margin for a classifier and a set of training data is the minimum margin of the training data $\gamma = \min\left\{y_m \cdot \left(\vec{w}^T\vec{x}_m + b\right)\right\}$



Support Vector Machines

Support Vector Machines (SVM) are a form of maximum margin classifier, where the model is defined by a small subset of the training data. SVM are popular for problems of classification, regression and novelty detection. The selection of the training samples that define the model corresponds to a convex optimization problem.

SVM's use a minimal subset of the training data (the "support vectors) to define the "best" decision surface between two classes. We will use the two class problem, K=2, to illustrate the principle. Multi-class solutions are possible.

The simplest case, the hard margin SVM, require that the training data be completely separated by at least one hyper-plane. This is generally achieved by using a Kernel to map the features into a high dimensional space were the two classes are separable.

To illustrate the principle, we will first examine a simple linear SVM where the data are separable. We will then generalize with Kernels and with soft margins.

We will assume that the training data is a set of M training samples $\{\vec{x}_m\}$ with an indicator variable, $\{y_m\}$, where , y_m is -1 or +1.

Hard-Margin SVMs - a simple linear classifier.

The simplest case is a linear classifier trained from separable data.

$g(\vec{x}) = \vec{w}^T\vec{x} + b$ with the decision rule is: IF $g(\vec{x}) \geq 0$ THEN P else N

For a hard margin SVM we assume that the two classes are separable for all of the training data:

$$\forall m: \; y_m(\vec{w}^T\vec{x}_m + b) \geq 0$$

We will use a subset S of the training samples, $\{\vec{x}_s\} \subset \{\vec{x}_m\}$ composed of M_s training samples to define the "best" decision surface $g(\vec{x}) = \vec{w}^T\vec{x} + b$.

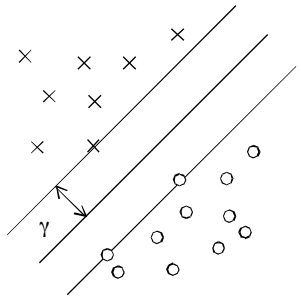
The minimum number of support vectors depends on the number of features, D:

$$M_s = D+1$$

The M_s selected training samples $\{\vec{x}_s\}$ are called the support vectors. For example, in a 2D feature space we need only 3 training samples to serve as support vectors.

To define the classifier we will look for the subset if S training samples that maximizes the margin (γ) between the two classes.

Margin: $\gamma = \min\left\{y_m \cdot \left(\vec{w}^T \vec{x}_m + b\right)\right\}$



Thus we seek a subset of $M_s=D+1$ training samples, $\{\vec{x}_s\} \subset \{\vec{x}_m\}$ to define a decision surface and the margin. We will use these samples as support vectors.

The algorithm must choose D+1 training samples $\{\vec{x}_s\}$ from the M Training samples in $\{\vec{x}_m\}$ such that the margin is a large as possible. This is equivalent to a search for a pair of parallel surfaces a distance γ from the decision surface.

Finding the Support Vectors

Assume that we have M training samples $\{\vec{X}_m\}$ and their indicator variable $\{y_m\}$, where, y_m is -1 or +1.

The decision surfaces is a hyper plane $\vec{W}^T \vec{X}_m + b = 0$

Assume that we have normalized the coefficients of the hyperplane such at

$$\|\vec{W}\| = 1$$

Then the distance of any sample point $\vec{X}_m$ from the hyper-plane, $\vec{W}$ is

$$d = y_m(\vec{W}^T \vec{X}_m + b)$$

A D dimensional decision surface is defined by at least D points. At least one additional point is required to define the margin. Thus we seek a subset of $M_s=D+1$ training samples, $\{\vec{X}_s\} \subset \{\vec{X}_m\}$ to define a decision surface and is margin.

Our algorithm must choose D+1 training samples $\{\vec{X}_s\}$ from the M Training samples in $\{\vec{X}_m\}$ such that the margin is a large as possible. This is equivalent to a search for a pair of parallel surfaces a distance γ from the decision surface.

The margin is the minimum distance

$$\gamma = \min\{y_n(\vec{W}^T\vec{X}_m + b)\}$$

For the D+1 support vectors $\vec{x}_s \in \{\vec{x}_s\}$ $\quad d_s = \gamma$
For all other training samples $\vec{x}_m \notin \{\vec{x}_s\}$: $d_m \geq \gamma$

The scale of the margin is determined by $\|\vec{w}\|$.

To find the support vectors, we can arbitrarily define the margin as $\gamma = 1$ and then renormalize $\|\vec{w}\| = 1$ once the support vectors have been discovered.

With $\gamma = 1$, we will look for two separate hyper-planes that "bound" the decision surface, such that for points on the surfaces: $\vec{w}^T\vec{x}_m + b = 1$ and $\vec{w}^T\vec{x}_m + b = -1$

The distance between these two planes is $\frac{2}{\|\vec{w}\|}$

We will add the constraint that for all support vectors

$$y_m\left(\vec{w}^T\vec{x}_m + b\right) = 1$$

while for all other samples:

$$y_m\left(\vec{w}^T\vec{x}_m + b\right) \geq 1$$

If we note that minimizing $\|\vec{W}\|$ is equivalent minimizing $\frac{1}{2}\|W\|^2$, we can set this up as a quadratic optimization problem, and use Lagrange Multipliers.

Our problem is to maximize $\arg\min_{\vec{w},b}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}$ while minimizing the number of active points, M_s. (The factor of ½ is a convenience for analysis.) This can be solved as a quadratic optimization problem using Lagrange Multipliers.

For a subset of $M_s \geq$ D+1 samples, $a_m > 0$. These are the support vectors.

For these vectors we set $a_m=1$.
For all other samples $a_m \le 0$. Se set these to $a_m = 0$

The Lagrangian function is: $L(\vec{w},b,\vec{a}) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{m=1}^{M} a_m \left\{ y_m(\vec{w}^T\vec{x}_m + b) - 1 \right\}$

Setting the derivatives to zero, we obtain:

$$\frac{\partial L}{\partial \vec{w}} = 0 \Rightarrow \quad \vec{w} = \sum_{m=1}^{M} a_m y_m \vec{X}_m \qquad\qquad \frac{\partial L}{\partial b} = 0 \Rightarrow \quad b = \frac{1}{M_s} \sum_{x_s \in \{x_s\}} \vec{w}^T\vec{X}_s - y_s$$

The normal of the decision surface is then: $\vec{W} = \sum_{m=1}^{M} a_m y_m \vec{X}_m$

where most of the a_m are zero. The M_s nonzero a_m select the support vectors.

and the offset can be found by solving for:

$$b = \frac{1}{M_S} \sum_{m \in S} \vec{W}^T\vec{X}_m - y_m$$

Giving $g(\vec{X}) = \vec{w}^T\vec{X} + b$.

Note that only S=D+1 of the coefficients a_m are non-zero. These S non-zero coefficients select the support vectors from within the complete set of training data.

We use the terms a_m to compose a set $\{\vec{X}_S\}$ of M_s support vectors.

The formula $b = \frac{1}{M_s} \sum_{x_m \in S} \vec{w}^T\vec{X}_m - y_m$ computes b as the average bias for all the support vectors (corrected by the indicator variable).

In fact, the bias should the same for any support vector.
Thus we can use any of the support vectors to compute the bias:

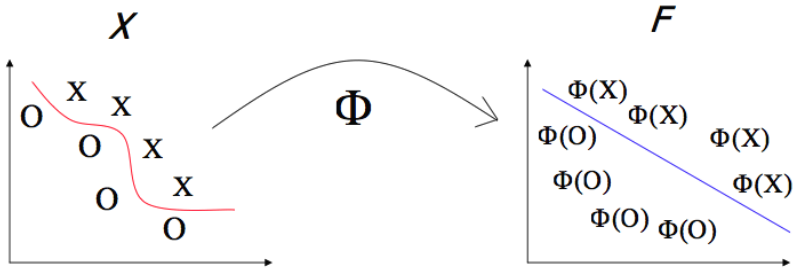
$$b = \vec{w}^T\vec{X}_s - y_s \quad \text{for any} \quad \vec{X}_S \in \{S\}$$

The solution can be generalized for use with non-linear decision surfaces using kernels.

SVM with Kernels

Support Vector Machines can be generalized for use with non-linear decision surfaces using kernel functions. This provides a very general and efficient classifier.

A "kernel" function, f(z), transforms the data into a space where the two classes are separate.



Instead of a decision surface: $g(\vec{x}) = \vec{w}^T \vec{x} + b$

We use a decision surface $g(\vec{x}) = \vec{w}^T \cdot \vec{f}(\vec{x}) + b$

the coefficients $\vec{w}$ and the bias b are provided by the D+1 support vectors.
For this we need to include the

where $\vec{w} = \sum_{m=1}^{M} a_m y_m \vec{f}(\vec{x}_m)$

is learned from the transformed training data.

As before, $S = \{\vec{x}_s\}$ are the support vectors, and a_m is a variable learned from the training data that is $a_m \geq 0$ for support vectors and 0 for all others.

Radial Basis Function Kernels

Radial Basis functions are a popular kernel function: A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin.

For Example, the Gaussian function $f(z) = e^{-\frac{z^2}{2\sigma^2}}$

is a popular Radial Basis Function, and is often used as a kernel for support vector machines, with radial basis functions: $z = \| \vec{X} - \vec{x}_s \|$ for each support vectors.

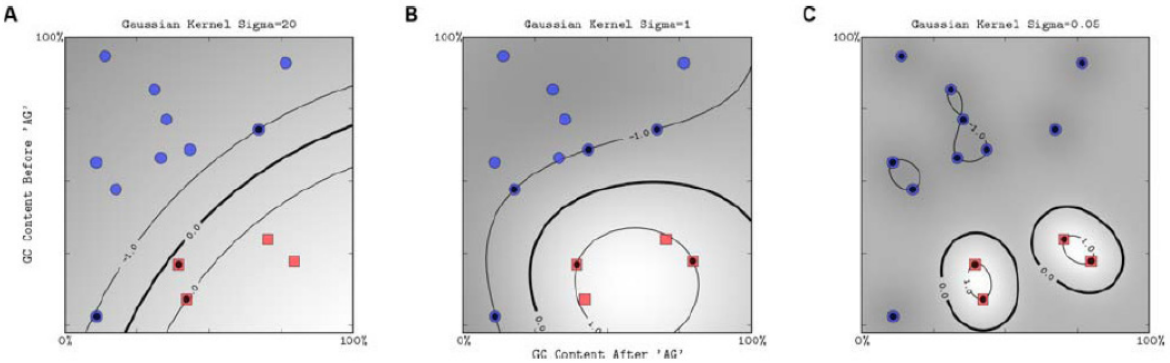
We can use a sum of M_s radial basis functions to define a discriminant function, where the support vectors are drawn from the M training samples.
This gives a discriminant function

$$g(\vec{X}) = \sum_{m=1}^{M} a_m y_m f(\| \vec{X} - \vec{x}_m \|) + b,$$

The training samples $\vec{X}_m$ for which $a_m \neq 0$ are the support vectors.

Depending on σ, this can provide a good fit or an over fit to the data. If σ is large compared to the distance between the classes, this can give an overly flat discriminant surface. If σ is small compared to the distance between classes, this will over-fit the samples.

A good choice for σ will be comparable to the distance between the closest members of the two classes.



(images from "A Computational Biology Example using Support Vector Machines", Suzy Fei, 2009)

Each Radial Basis Function is a dimension in a high dimensional basis space.

Kernel Functions for Symbolic Data

Kernel functions can be defined over graphs, sets, strings and text!

Consider for example, a non-vector space composed of a set of words {W}.
We can select a subset of discriminant words {S} ⊂ {W}

Now given a set of words (a probe), {A} ⊂ {W}

We can define a kernel function of A and S using the intersection operation.

$$k(A,S) = 2^{|A \cap S|}$$

where | . | denotes the cardinality (the number of elements) of a set.