Support Vector Machine (SVM) Classifier

WEEK 2

Reading Materials

Lecture Slides: https://math.berkeley.edu/~scanlon/m16bs04/ln/16b2lec3.pdf
Simple Lagrange Multiplier, Berkeley Math. (Simple, with examples to work on, no inequality constraints.)

Note:http://ocw.mit.edu/courses/mathematics/18-02sc-multivariable-calculus-fall-2010/2.-partial-derivatives/part-c-lagrange-multipliers-and-constrained-differentials/session-40-proof-of-lagrange-multipliers/MIT18_02SC_notes_22.pdf
Proof of Lagrange multiplier, MIT OpenCourseWare

Video: https://www.youtube.com/watch?v=_PwhiWxHK8o
Patrick Winston, MIT OpenCourseWare (Simpler introduction, no lagrange multiplier, no primal-dual form)

Video: https://www.youtube.com/view_play_list?p=A89DCFA6ADACE599 (Lecture 6 - 8)
Andrew Ng, Stanford

Lecture Notes: http://cs229.stanford.edu/notes/cs229-notes3.pdf
Andrew Ng

The Optimal Margin Classifier

The distance to each point is

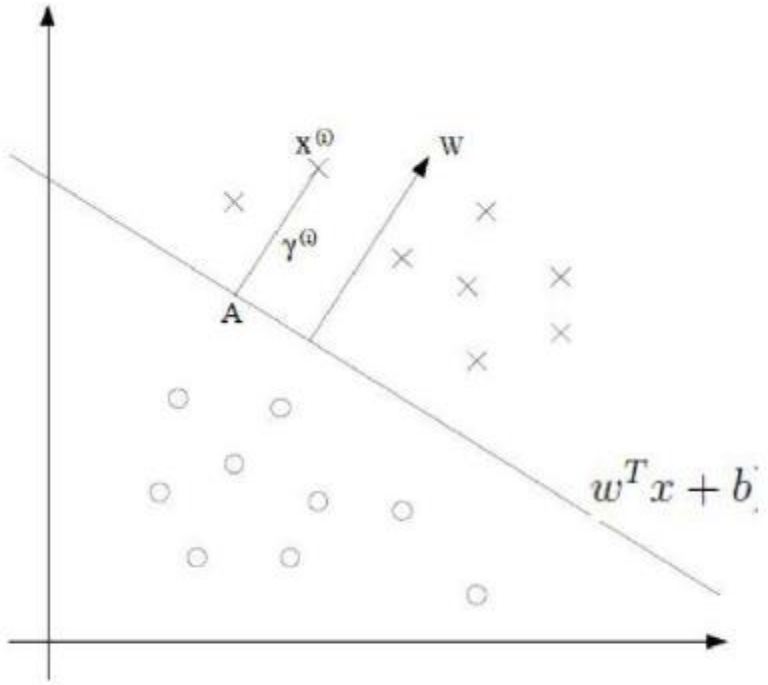
$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{w}{||w||} \right)^T x^{(i)} + \frac{b}{||w||} \right)$$

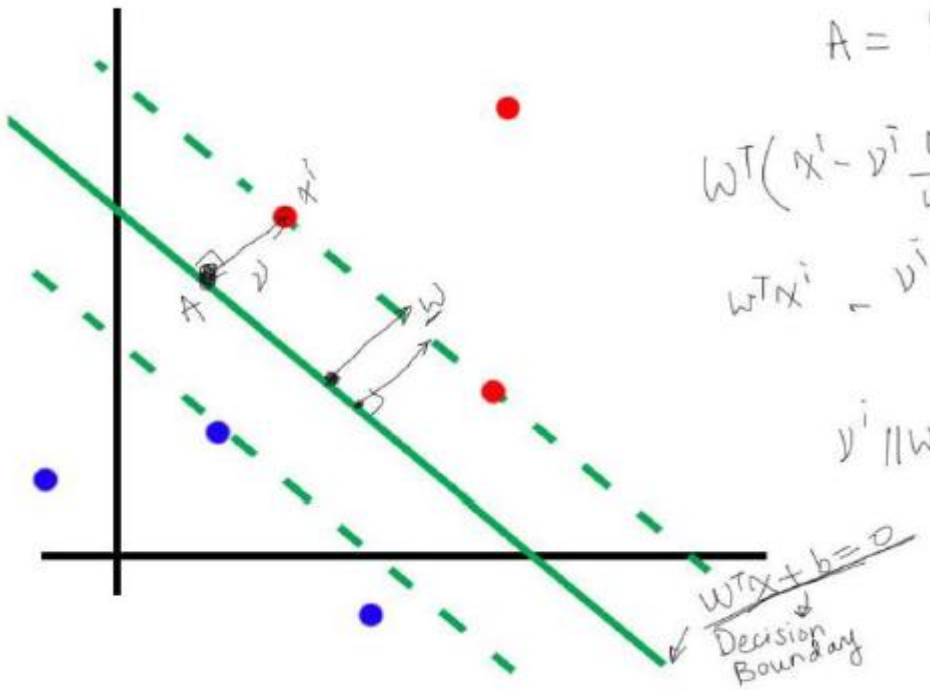
We want to maximize the worst case distanc $\gamma = \min_{i=1,\ldots,m} \gamma^{(i)}$

Which after simplification, becomes:

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m \end{aligned}$$

(see next slide for detailed derivation)





$$A = x^i - \nu^i \frac{w}{\|w\|}$$ → 因为 A 在 Decision Boundary 上：

$$w^T(A) + b = 0$$

$$w^T\left(x^i - \nu^i \frac{w}{\|w\|}\right) + b = 0$$

$$w^T x^i - \nu^i \frac{w^T w}{\|w\|} + b = 0 \qquad w^T w = \|w\|^2$$

$$\nu^i \|w\| = w^T x^i + b$$

$$\nu^i = \frac{w^T}{\|w\|} x^i + \frac{b}{\|w\|}$$

← when x^i is positive

When x_i is neg:

$$\nu^i = -\frac{w^T x^i}{\|w\|} + \frac{b}{\|w\|}$$

Let $y^i = \begin{cases} +1, & \text{if } x^i \text{ pos} \\ -1, & \text{if } x^i \text{ neg} \end{cases}$

Then:

$$\nu^i = y^i\left(\frac{w^T x^i}{\|w\|} + \frac{b}{\|w\|}\right)$$

Construct the optimization problem.

$\nu = \min \nu'$

$\max \nu$ s.t. 点到 the margin 的距离 $\geq \nu$

$$\max \frac{\nu}{\|w\|} \quad \text{s.t. } y^i\left(\frac{w^T x^i}{\|w\|} + \frac{b}{\|w\|}\right) \geq \frac{\nu}{\|w\|}.$$

because $\|w\|$ is a number.

Let $\nu = 1$

$$\max \frac{1}{\|w\|} \quad \text{s.t. } y^i\left(w^T x^i + b\right) \geq 1$$

$$\min \frac{1}{2}\|w\|^2 \quad \text{s.t. } y^i(w^T x^i + b) \geq 1, \text{ for all } i.$$

Generalized Lagrange Multiplier - Primal Problem

For an optimization problems with inequality constraints:

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \ldots, k \\ & h_i(w) = 0, \quad i = 1, \ldots, l. \end{aligned}$$

The Lagrangian (or generalized Lagrangian) is:

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w).$$

Define $\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta : \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$ then $\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta : \alpha_i \geq 0} f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$

$$\theta_{\mathcal{P}}(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{otherwise.} \end{cases}$$

Thus $\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha, \beta : \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$ has the same solution as the original problem.

(see next slide for the derivation used in class)

Generalized Lagrange Multiplier - Dual Problem

Define $\theta_{\mathcal{D}}(\alpha, \beta) = \min_{w} \mathcal{L}(w, \alpha, \beta)$ and consider the dual problem $\max_{\alpha,\beta:\alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha, \beta) = \max_{\alpha,\beta:\alpha_i \geq 0} \min_{w} \mathcal{L}(w, \alpha, \beta)$

Comparing the primal problem $\min_{w} \theta_{\mathcal{P}}(w) = \min_{w} \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$

and the dual problem $\max_{\alpha,\beta:\alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha, \beta) = \max_{\alpha,\beta:\alpha_i \geq 0} \min_{w} \mathcal{L}(w, \alpha, \beta)$

It can be shown that: $d^* = \max_{\alpha,\beta:\alpha_i \geq 0} \min_{w} \mathcal{L}(w, \alpha, \beta) \leq \min_{w} \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) = p^*$

$d^* = p^*$ nder certain conditions: (1) $g_i(w)$ are convex,
(2) $h_i(w)$ can be written as $h_i(w) = a_i^T w + b_i$

Then there exist w^*, α^*, β^* s.t $p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$

Generalized Lagrange Multiplier - Primal/Dual Problem

If the above conditions are satisfie w^*, α^*, β^* also satisfy the **KKT dual complementarity conditions**:

$$
\begin{aligned}
\frac{\partial}{\partial w_i}\mathcal{L}(w^*, \alpha^*, \beta^*) &= 0, \quad i = 1, \ldots, n \\
\frac{\partial}{\partial \beta_i}\mathcal{L}(w^*, \alpha^*, \beta^*) &= 0, \quad i = 1, \ldots, l \\
\alpha_i^* g_i(w^*) &= 0, \quad i = 1, \ldots, k \\
g_i(w^*) &\leq 0, \quad i = 1, \ldots, k \\
\alpha^* &\geq 0, \quad i = 1, \ldots, k
\end{aligned}
$$

When $\alpha_i^* > 0$, (implies $g_i(w^*) = 0$) this constraint is said to be **active.**

Back to our optimization

$$\min_{\gamma,w,b} \quad \frac{1}{2}||w||^2$$
$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m$$

After some derivation: *(see next two slides for derivation)*

$$\mathcal{L}(w, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)}.$$

and the problem becomes:

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle.$$
$$\text{s.t.} \quad \alpha_i \geq 0, \quad i = 1, \ldots, m$$
$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0,$$

Sequential Minimal Optimization for SVM

The dual problem:

$$\begin{aligned} \max_\alpha \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle. \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0, \end{aligned}$$

When updating the 1st variable, the constraints ensures that $\alpha_1 y^{(1)} = -\sum_{i=2}^{m} \alpha_i y^{(i)} \rightarrow \alpha_1 = -y^{(1)} \sum_{i=2}^{m} \alpha_i y^{(i)}$

- Update a pair of variable at one time.

Repeat till convergence {

1. Select some pair α_i and α_j to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum).
2. Reoptimize $W(\alpha)$ with respect to α_i and α_j, while holding all the other α_k's $(k \neq i, j)$ fixed.

}

Coordinate Ascent

(This method is used to solve the dual problem in the last slide)

Consider a maximization problem,

$$\max_{\alpha} W(\alpha_1, \alpha_2, \ldots, \alpha_m)$$

The coordinate ascent algorithm:

Loop until convergence: {

For $i = 1, \ldots, m$, {

$\alpha_i := \arg\max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_m)$

}

}

The Optimal Margin Classifier: Review

1 $\min_{\gamma,w,b} \quad \frac{1}{2}||w||^2$
s.t. $y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m$

..... Problem formulation from geometry.

2 $\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_i \alpha_i \left(y^i(w^T x^i + b) - 1\right)$

..... Construct Lagrangian.

3 $d^* = \max_{\alpha} \min_{w,b} \frac{1}{2}||w||^2 - \sum_i \alpha_i \left(y^i(w^T x^i + b) - 1\right)$

..... Construct the dual problem.

4 $d^* = \max_{\alpha} \mathcal{L}^*(w, b, \alpha) = \sum_i \alpha - \frac{1}{2}\sum_{i,j} y^i y^j \alpha_i \alpha_j \langle x^i, x^j \rangle$

..... Solver the inner optimization in the dual problem.

5 $\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle.$
s.t. $\alpha_i \geq 0, \quad i = 1, \ldots, m$
$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0,$

..... Reformulate the problem

The Optimal Margin Classifier: Review

$$\max_\alpha \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle.$$

$$\text{s.t.} \quad \alpha_i \geq 0, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0,$$

..... New problem to solve.

$$w = \sum_i \alpha_i y^i x^i$$

$$b = \frac{1}{2} \left(\max_{i:y^i=-1} w^T x^i + \min_{i:y^i=1} w^T x^i \right)$$

..... After solving α_i by using Sequential Minimal Optimization.

$$\hat{y}^i = \phi(w^T x^i + b)$$

$$= \phi \left(\sum_j \alpha_i y^i \langle x^j, x^i \rangle + b \right)$$

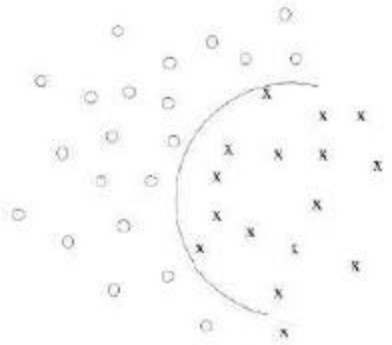
..... To classify a new data case

Kernels

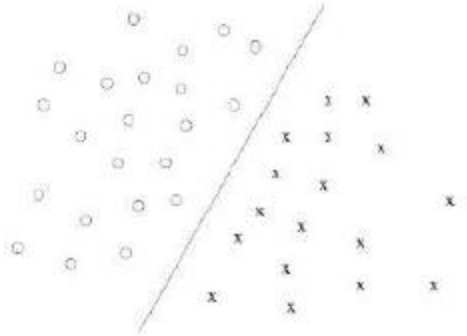
Some samples are non-linear separable.

We can either 1) use a non-linear separator,
or 2) transform the data so it is linear separable

For solution (2): Replace x by $\phi(x)$



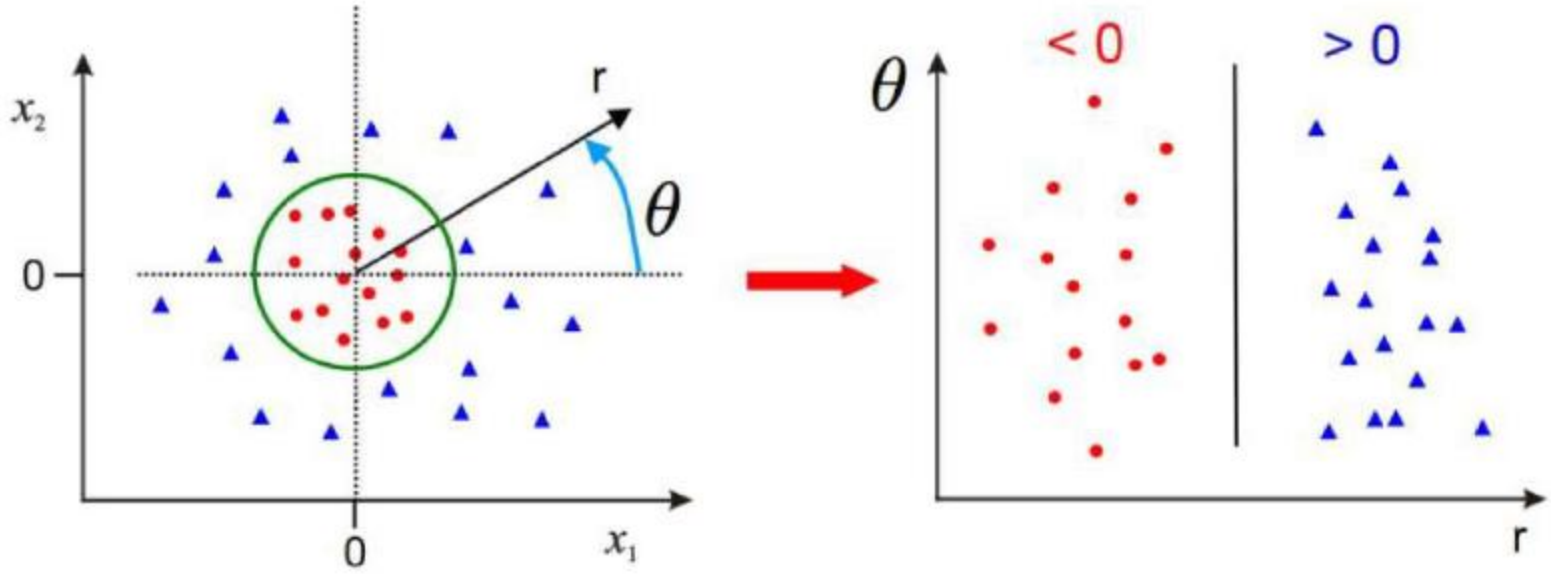
Non-linear separator in the original $\mathbf{x}$-space



Linear separator in the feature ϕ-space

[Tommi Jaakkola]

Kernels: Example



- Data is linearly separable in polar coordinates
- Acts non-linearly in original space

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} r \\ \theta \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

Kernels

Recall that: When we classify a new data case,

$$\begin{aligned}\hat{y}^i &= \phi(w^T x^i + b) \\ &= \phi\left(\sum_j \alpha_i y^i \langle x^j, x^i \rangle + b\right)\end{aligned}$$

Which only depends on the inner product of the input feature vectors.

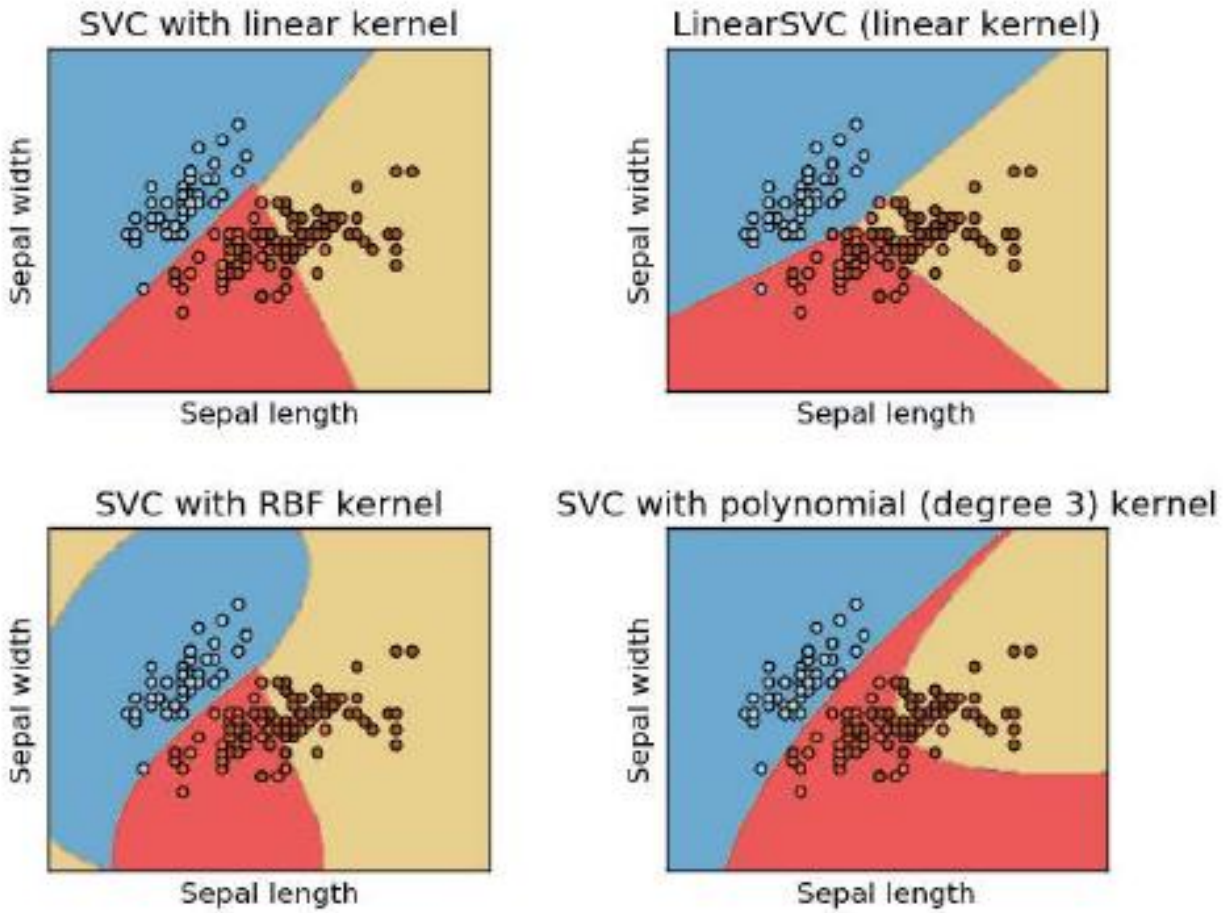
Instead of transforming x, we can just transform the inner product. Define **Kernel** as:

$$K(x, z) = \phi(x)^T \phi(z)$$

Some useful kernels include:

$$\text{Polynomial: } K(x, z) = (x^T z + 1)^d$$

$$\text{Gaussian (or Radial, or RBF): } K(x, z) = \exp(-\tfrac{||x-z||^2}{2\sigma^2})$$

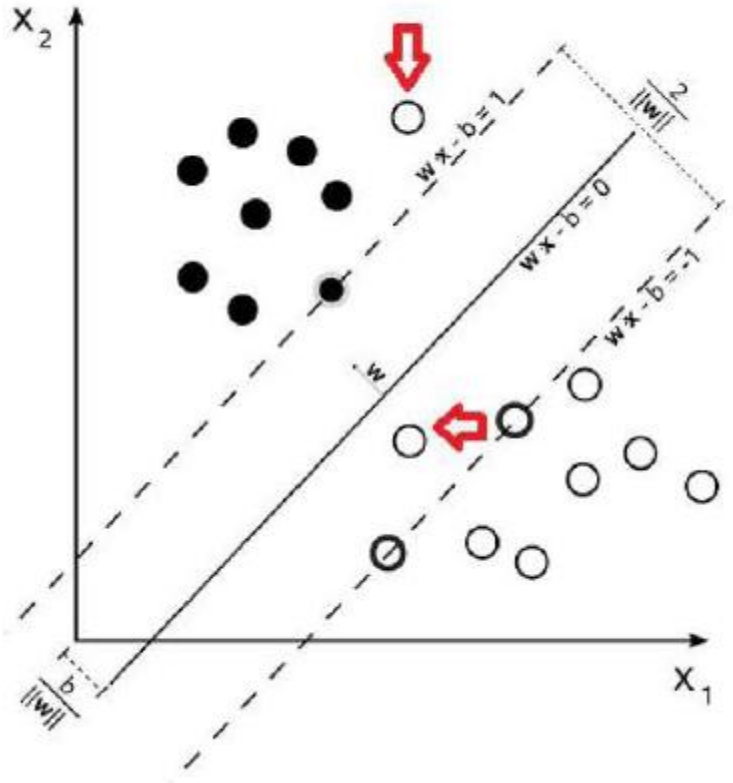

SVC with linear kernel
LinearSVC (linear kernel)
SVC with RBF kernel
SVC with polynomial (degree 3) kernel
Sepal width
Sepal length

SVM with Non-separable Case

Add a term ξ_i to allow samples pass the margin.

Similarly to the separable case:

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m \\ & \xi_i \geq 0, \quad i = 1, \ldots, m. \end{aligned}$$



SVM with Non-separable Case

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1,\ldots,m \\ & \xi_i \geq 0, \quad i = 1,\ldots,m. \end{aligned}$$

Lagrangian:

$$\mathcal{L}(w,b,\xi,\alpha,r) = \frac{1}{2}w^T w + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left[y^{(i)}(x^T w + b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i$$

Construct the dual problem by first set $\theta_{\mathcal{D}}(\alpha, r) = \min_{w,b,\xi} \mathcal{L}(w,b,\xi,\alpha,r)$

Solve by setting the 3 partial derivatives w.r.t w, b, ξ to 0, then the dual problem becomes

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,m \\ & \sum_{i=1}^{m}\alpha_i y^{(i)} = 0, \end{aligned}$$

This can then be solved by SMO.

SVC, NuSVC, SVR

SVC VS NuSVC

We introduce a new parameter nu which controls the number of support vectors and training errors. The parameter nu is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors.

SVC VS LinearSVC

They are just different implementations of the same algorithm. The SVM module (SVC, NuSVC, etc) is a wrapper around the libsvm library and supports different kernels while LinearSVC is based on liblinear and only supports a linear kernel.

SVC VS SVR

The technical difference is that in SVC classification the hinge loss is used as opposed to the epsilon-sensitive loss that is used in SVR regression.

Multi-class classification

SVC and NuSVC implement the “one-against-one” approach for multi- class classification. If n_class is the number of classes, then n_class * (n_class - 1) / 2 classifiers are constructed and each one trains data from two classes. To provide a consistent interface with other classifiers, the decision_function_shape option allows to aggregate the results of the “one-against-one” classifiers to a decision function of shape (n_samples, n_classes).

On the other hand, LinearSVC implements “one-vs-the-rest” multi-class strategy, thus training n_class models. If there are only two classes, only one model is trained:

Scores and probabilities

The SVC method decision_function gives per-class scores for each sample (or a single score per sample in the binary case). When the constructor option probability is set to True, class membership probability estimates (from the methods predict_proba and predict_log_proba) are enabled. In the binary case, the probabilities are calibrated using Platt scaling: logistic regression on the SVM's scores, fit by an additional cross-validation on the training data. In the multiclass case, this is extended as per Wu et al. (2004).

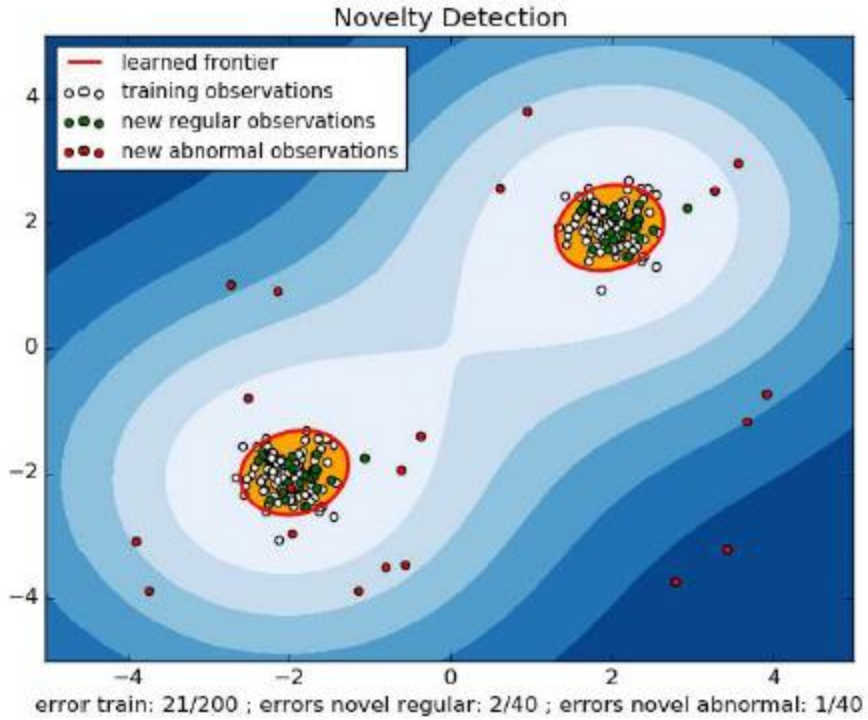
Unbalanced problems

In problems where it is desired to give more importance to certain classes or certain individual samples keywords class_weight and sample_weight can be used.

SVC (but not NuSVC) implement a keyword class_weight in the fit method. It's a dictionary of the form {class_label : value}, where value is a floating point number > 0 that sets the parameter C of class class_label to C * value.

SVC, NuSVC, SVR, NuSVR and OneClassSVM implement also weights for individual samples in method fit through keyword sample_weight. Similar to class_weight, these set the parameter C for the i-th example to C * sample_weight[i].

Density estimation, novelty detection



Tips on Practical Use

Kernel cache size

- For SVC, SVR, nuSVC and NuSVR, the size of the kernel cache has a strong impact on run times for larger problems. If you have enough RAM available, it is recommended to set cache_size to a higher value than the default of 200(MB), such as 500(MB) or 1000(MB).

Setting C

- C is 1 by default and it's a reasonable default choice. If you have a lot of noisy observations you should decrease it. It corresponds to regularize more the estimation.

Support Vector Machine algorithms are not scale invariant, so it is highly recommended to **scale your data**. For example, scale each attribute on the input vector X to [0,1] or [-1,+1], or standardize it to have mean 0 and variance 1.

Parameters of the RBF Kernel

When training an SVM with the Radial Basis Function (RBF) kernel, two parameters must be considered: C and gamma. The parameter C, common to all SVM kernels, trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. gamma defines how much influence a single training example has. The larger gamma is, the closer other examples must be to be affected.

Proper choice of C and gamma is critical to the SVM's performance. One is advised to use sklearn.grid_search.GridSearchCV with C and gamma spaced exponentially far apart to choose good values.