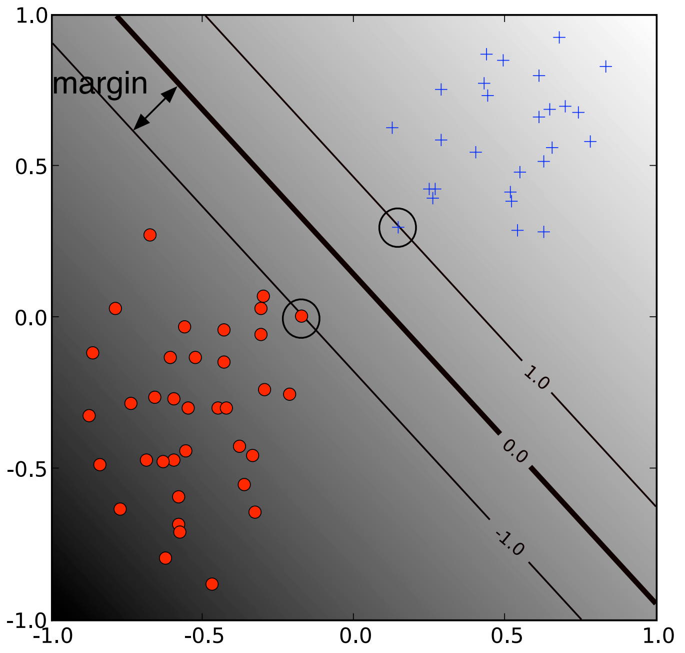
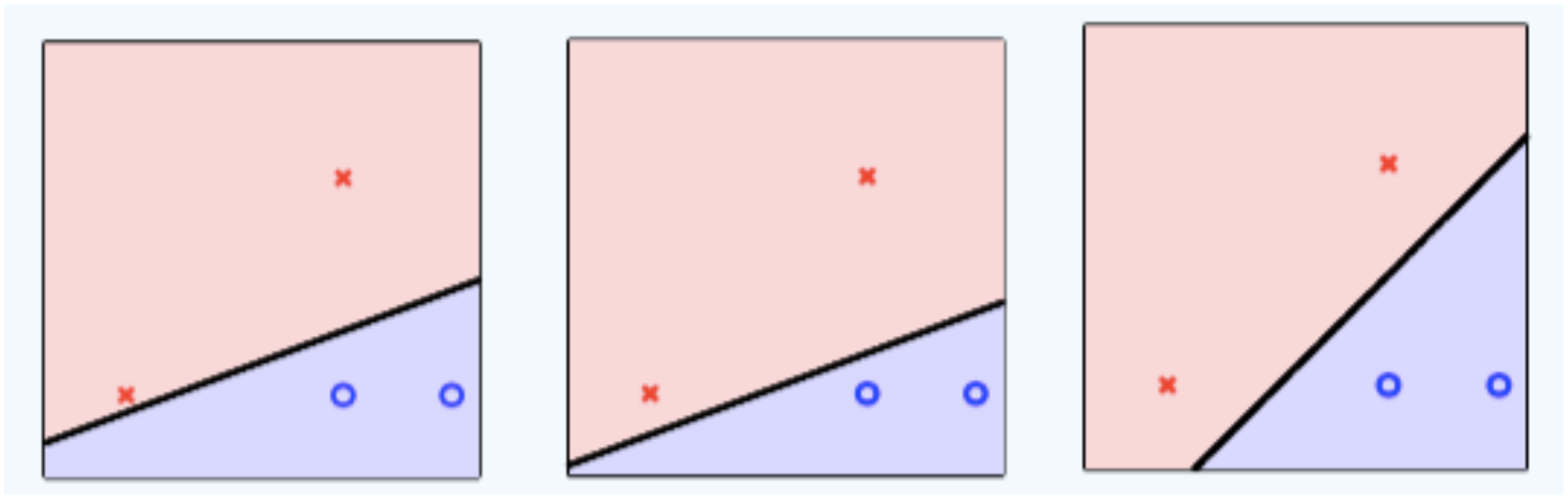
Support vector machines and large margin classification

Chapter e-8

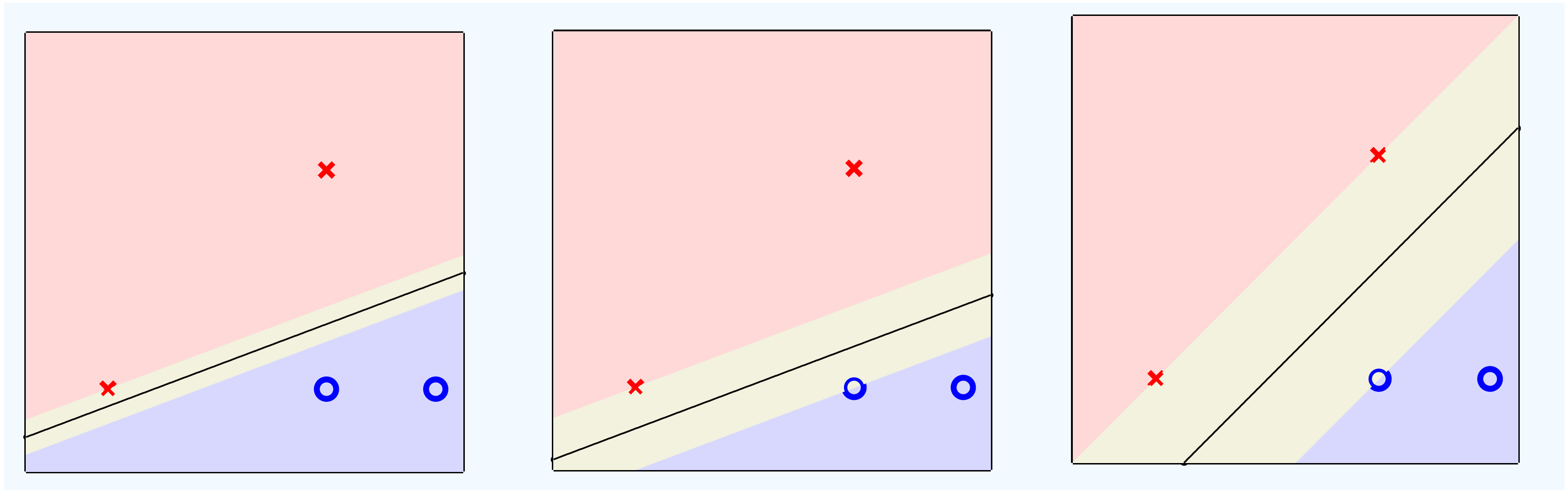


Which hyperplane is better?



Which of these hyperplanes is likely to yield a more accurate classifier?

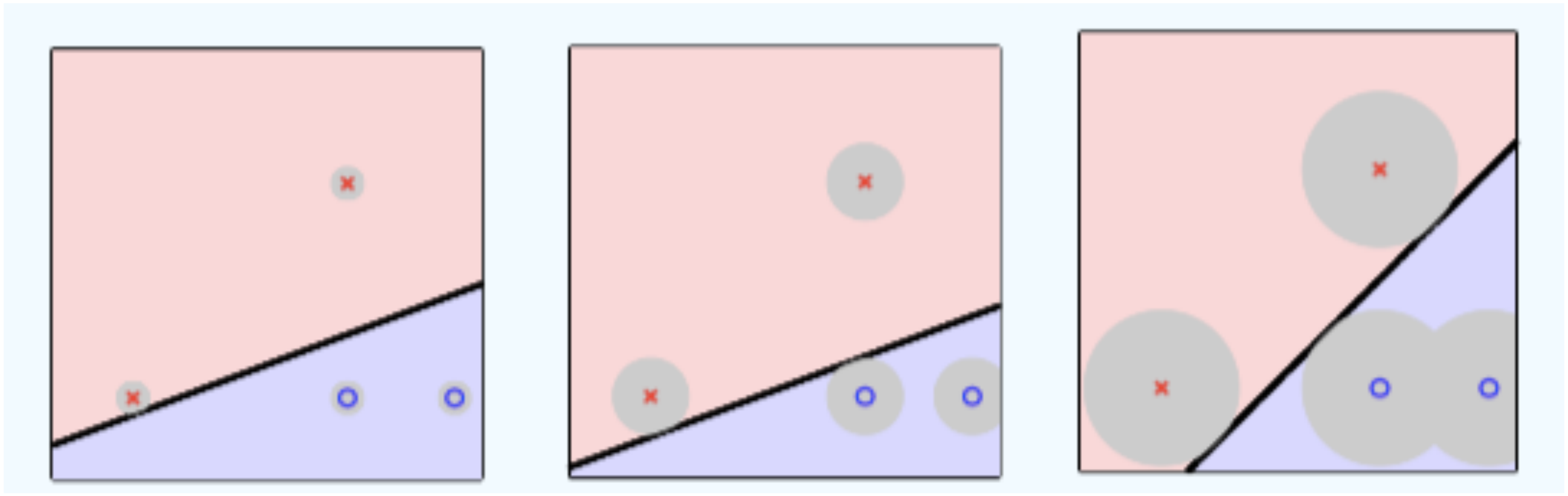
Which hyperplane is better?



The hyperplane on the right has the largest margin

The optimal margin hyperplane is introduced in section 8.1 in Chapter e-8 of the book

Which hyperplane is better?



It also provides the largest cushion against noise

The book calls such a hyperplane “fat”

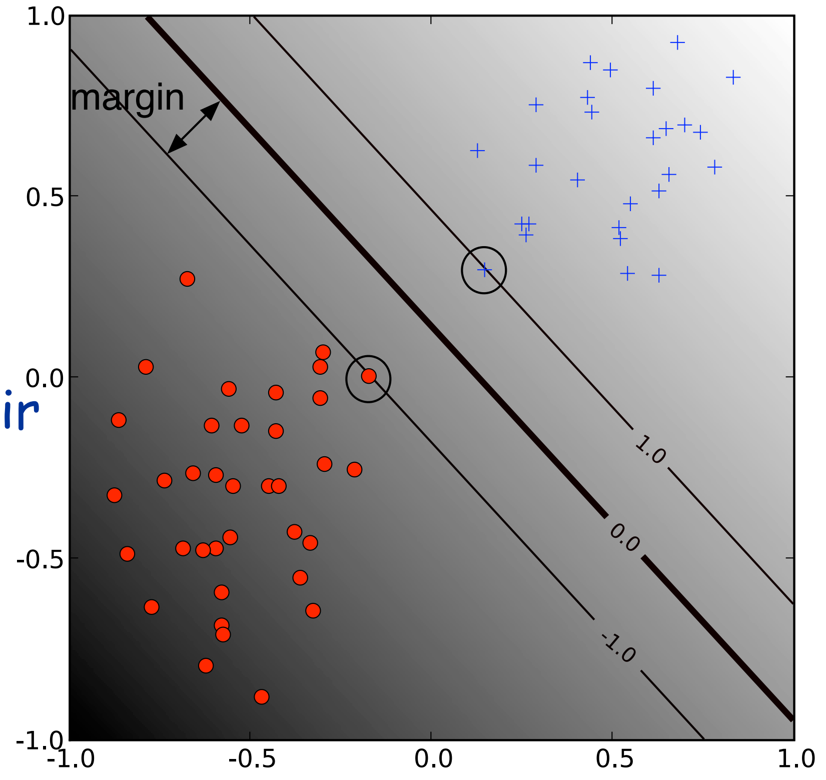
Large margin classifiers

Perceptron: find hyperplane that separates the two classes

Support Vector Machine (SVM): separating hyperplane with a large margin

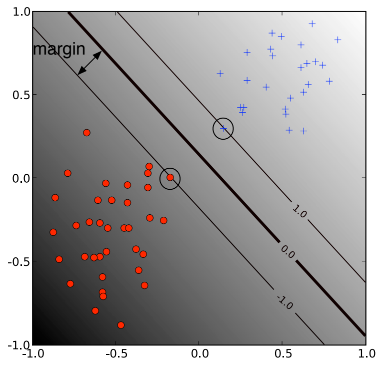
Intuitive concept that is backed by theoretical results
(statistical learning theory)

Has its origins in the work of Vladimir Vapnik



Vapnik, V., and A. Lerner. Pattern recognition using generalized portrait method. Automation and Remote Control, 24, 774–780, 1963.

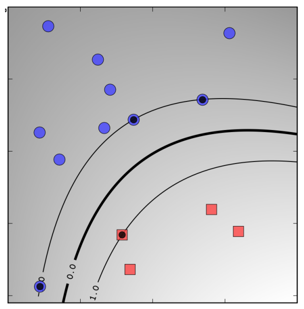
The history of SVMs



Large margin linear classifiers

- Vapnik, V., and A. Lerner. Pattern recognition using generalized portrait method. Automation and Remote Control, 24, 774–780, 1963.

Large margin non-linear classifiers



- B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In Fifth Annual Workshop on Computational Learning Theory, pages 144—152, 1992

SVMs for non-separable data

- C. Cortes and V. N. Vapnik, Support vector networks. Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.

Since then: lots of other large margin algorithms

Bring back the bias

Before:

$$\mathbf{x} \in \{1\} \times \mathbb{R}^d; \;\; \mathbf{w} \in \mathbb{R}^{d+1}$$

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}; \qquad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}.$$

$$\text{signal} = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

Now:

$$\mathbf{x} \in \mathbb{R}^d; \;\; b \in \mathbb{R}, \;\; \mathbf{w} \in \mathbb{R}^d$$

bias b

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}; \qquad \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}.$$

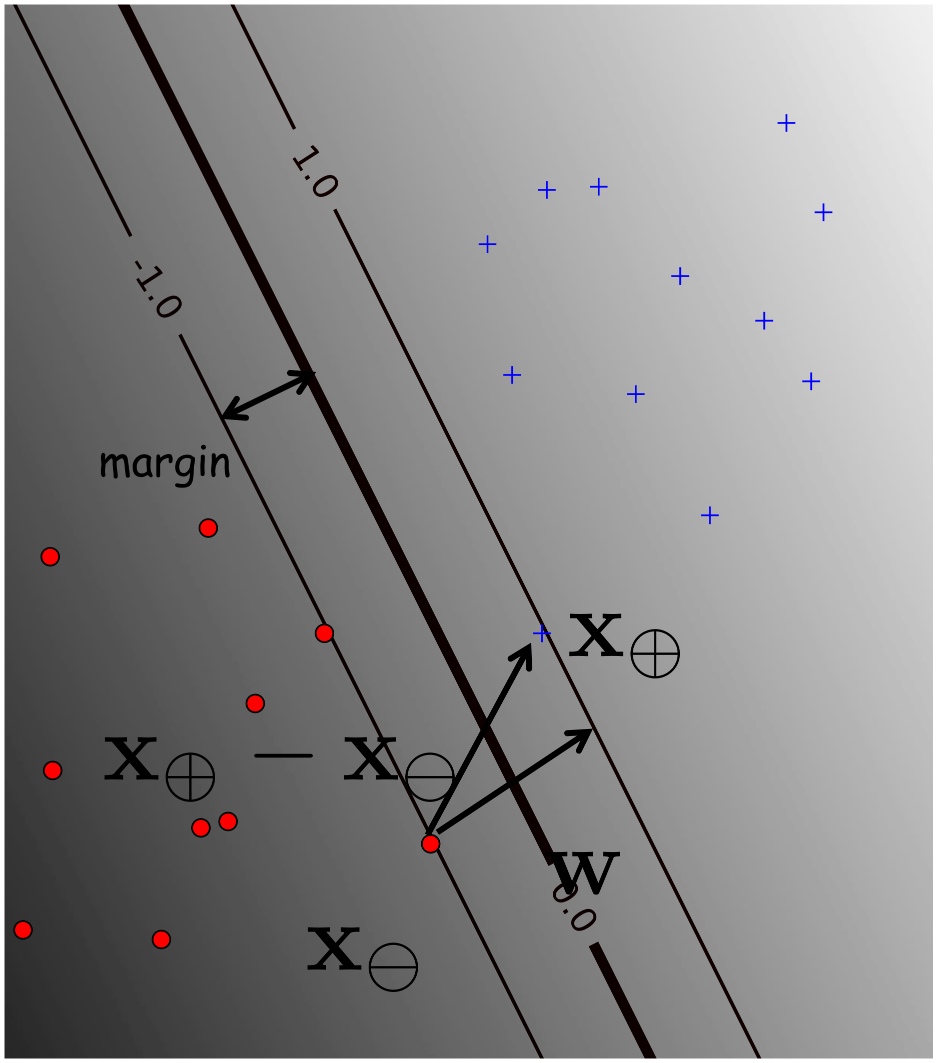
$$\text{signal} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$$

The geometric margin

The margin of a linear discriminant:

$$\frac{1}{2}\hat{\mathbf{w}}^{\mathsf{T}}(\mathbf{x}_{\oplus} - \mathbf{x}_{\ominus})$$

$\hat{\mathbf{w}}$ a unit vector in the direction of **w**



This exposition of the large margin hyperplane follows section 4.1 in
Asa Ben-Hur and Jason Weston. A User's guide to Support Vector Machines, 2009.
http://www.cs.colostate.edu/~asa/pdfs/howto.pdf

The geometric margin

Want to find:

$$\frac{1}{2}\hat{\mathbf{w}}^\intercal(\mathbf{x}_\oplus - \mathbf{x}_\ominus)$$

Suppose that x_+ and x_- are equidistant from the decision boundary:

$$\mathbf{w}^\intercal\mathbf{x}_\oplus + b = a$$

$$\mathbf{w}^\intercal\mathbf{x}_\ominus + b = -a$$

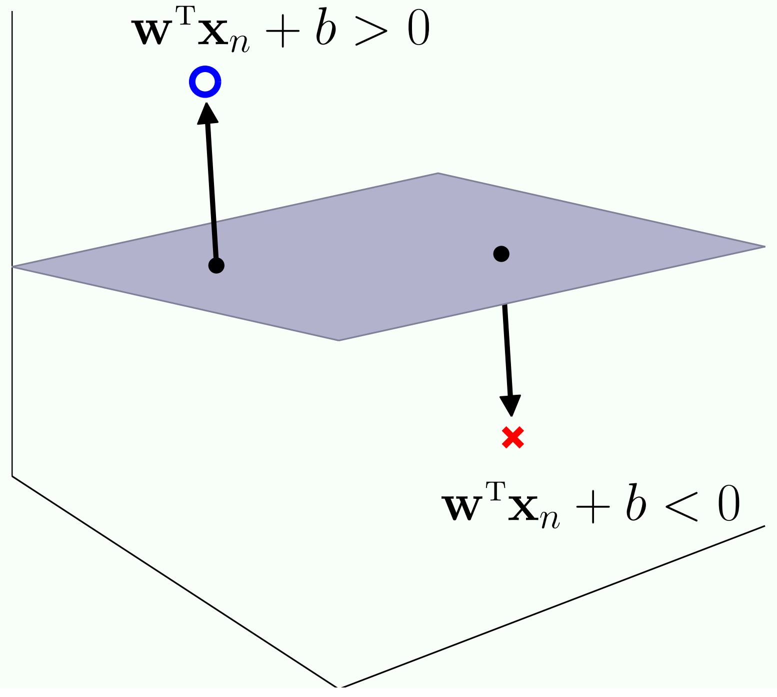
Subtracting the two equations:

$$\mathbf{w}^\intercal(\mathbf{x}_\oplus - \mathbf{x}_\ominus) = 2a$$

Divide by the norm of w:

$$\hat{\mathbf{w}}^\intercal(\mathbf{x}_\oplus - \mathbf{x}_\ominus) = \frac{2a}{||\mathbf{w}||}$$

Canonical separating hyperplane



$\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b > 0$

$\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b < 0$

Hyperplane $h = (b, \mathbf{w})$

h separates the data means:

$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) > 0$$

By rescaling the weights and bias,

$$\min_{n=1,\ldots,N} \; y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = 1$$

The geometric margin

To get a well-defined value we will use the canonical representation of a hyperplane.

Under this assumption we have that the margin equals $\frac{1}{||\mathbf{w}||}$

Maximizing the margin is therefore equivalent to minimizing $||\mathbf{w}||^2$

Motivation

Theoretical motivation: The VC dimension, which measures the complexity of a hypothesis, increases with decreasing margin.

The linear SVM

Objective: maximize the margin while correctly classifying all examples correctly

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to: } y_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geq 1 \quad i = 1, \ldots, N\,.$$

Digression: constrained optimization

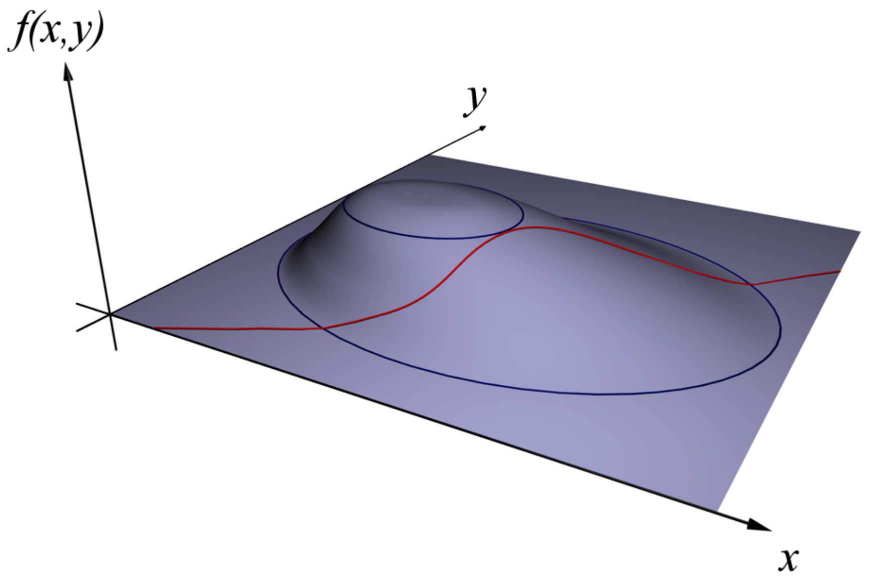
Before considering optimization problems with inequality constraints we will consider ones with equality constraints:

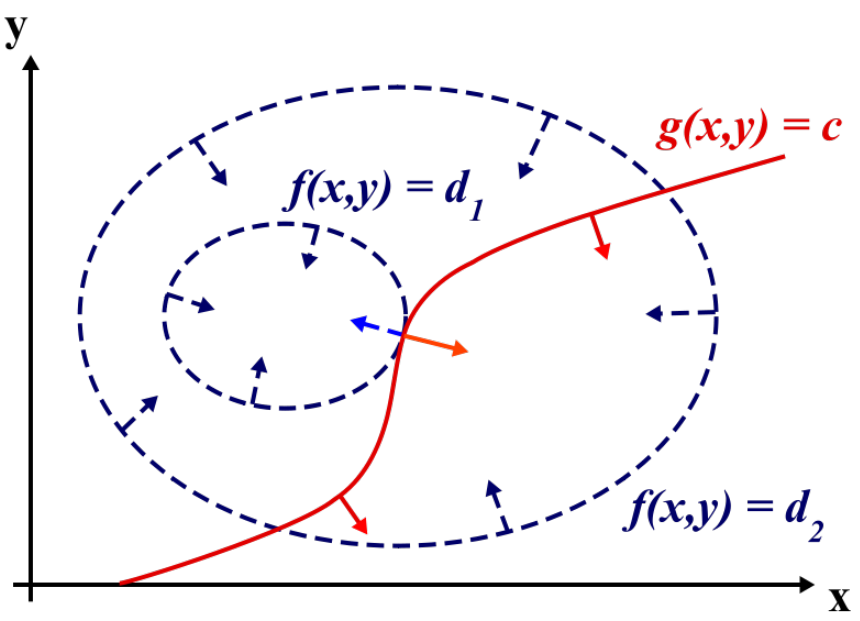
$$\text{minimize } f(\mathbf{x})$$
$$\text{subject to: } g_i(\mathbf{x}) = 0$$

And to make things even simpler, start with the case of a single constraint $g(\mathbf{x})$

$$\text{minimize } f(\mathbf{x})$$
$$\text{subject to: } g(\mathbf{x}) = 0$$

Digression: constrained optimization





Images from http://en.wikipedia.org/wiki/Lagrange_multiplier

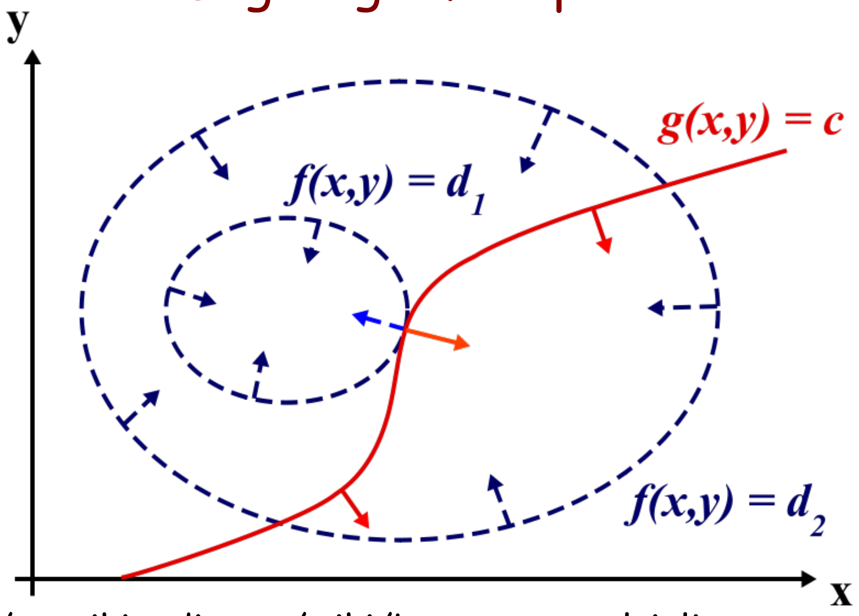
Digression: constrained optimization

Claim: A minimizer x^* of the constrained optimization problem must have the property that $\nabla f(\mathbf{x}^*)$ is orthogonal to the constraint surface.

Therefore there exists $\lambda \neq 0$ such that

$$\nabla f(\mathbf{x}^*) + \lambda \nabla g(\mathbf{x}^*) = 0$$

λ is known as the Lagrange multiplier



Images from http://en.wikipedia.org/wiki/Lagrange_multiplier

Lagrange multipliers

When there are multiple equality constraints:

$$\nabla f(\mathbf{x}^*) + \sum_i \lambda_i \nabla g_i(\mathbf{x}^*) = 0$$

The Lagrangian function:

$$\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_i \lambda_i g_i(\mathbf{x}) = f(\mathbf{x}) + \boldsymbol{\lambda}^{\mathsf{T}} g(\mathbf{x})$$

The above condition is obtained by setting

$$\nabla_{\mathbf{x}} \Lambda(\mathbf{x}, \boldsymbol{\lambda}) = 0$$

$\nabla_{\mathbf{x}}$ Denotes gradient with respect to x

And the condition $\nabla_{\boldsymbol{\lambda}} \Lambda(\mathbf{x}, \boldsymbol{\lambda}) = 0$ leads to the constraint equations.

Conclusion: the solution is a stationary point of the Lagrangian

Inequality constraints

$$\text{minimize } f(\mathbf{x})$$
$$\text{subject to: } g(\mathbf{x}) \leq 0$$

Two possible scenarios:

g(**x**) < 0 – the constraint is inactive

g(**x**) = 0 – the constraint is active

If the constraint is inactive the stationarity condition is $\nabla f(\mathbf{x}) = 0$

This corresponds to a stationary point of the Lagrangian with $\lambda = 0$

When the constraint is active, we have $\lambda \neq 0$

Both cases can be summarized by the condition

$$\lambda g(\mathbf{x}) = 0$$

The sign of λ is important: f(**x**) will be minimized only if its gradient is oriented away from the region g(x) < 0, i.e.

$$\nabla f(\mathbf{x}^*) = -\lambda \nabla g(\mathbf{x}^*) \text{ where } \lambda > 0$$

Constrained optimization with inequality constraints

Conclusion:

Our constrained optimization problem of minimizing f(**x**) such that g(**x**) ≤ 0 is solved by $\mathbf{x}, \lambda$ that satisfy:

$$\begin{aligned} \nabla \Lambda(\mathbf{x}, \lambda) &= 0 \\ g(\mathbf{x}) &\leq 0 \\ \lambda &\geq 0 \\ \lambda g(\mathbf{x}) &= 0 \end{aligned}$$

These are known as the KKT conditions

Constrained optimization with inequality constraints

With multiple constraints:

Our constrained optimization problem of minimizing f(**x**) such that g_i(**x**) ≤ 0 is solved by $\mathbf{x}, \boldsymbol{\lambda}$ that satisfy:

$$\begin{aligned} \nabla \Lambda(\mathbf{x}, \boldsymbol{\lambda}) &= 0 \\ g_i(\mathbf{x}) &\leq 0 \\ \boldsymbol{\lambda} &\geq 0 \\ \lambda_i g_i(\mathbf{x}) &= 0 \end{aligned}$$

These are known as the KKT conditions

Lagrangian duality

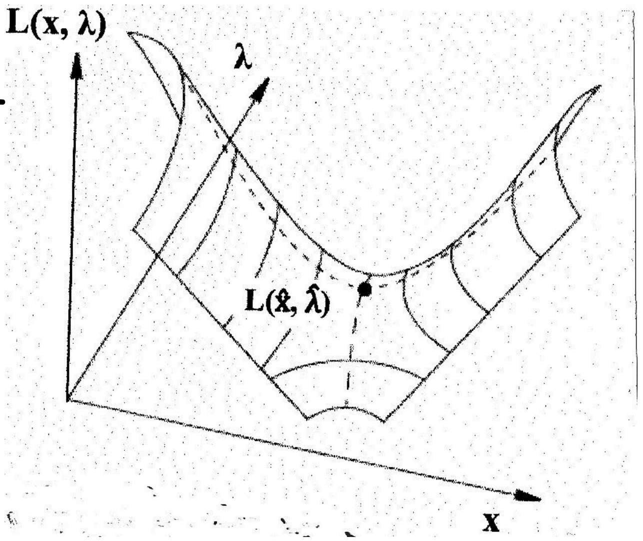
Claim: The problem of minimizing f($\mathbf{x}$) s.t. $g_i(\mathbf{x}) \le 0$ can be expressed as:

$$\min_{\mathbf{x}} \max_{\boldsymbol{\lambda}} \Lambda(\mathbf{x}, \boldsymbol{\lambda}) \text{ such that } \boldsymbol{\lambda} \geq 0$$

We can see this by performing the inner maximization:

$$\max_{\boldsymbol{\lambda}} f(\mathbf{x}) + \boldsymbol{\lambda}^{\mathsf{T}} g(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & g(\mathbf{x}) \leq 0 \\ \infty & g(\mathbf{x}) > 0 \end{cases}$$

Solution is a saddle point



Lagrangian duality

Claim: The problem of minimizing f($\mathbf{x}$) s.t. $g_i(\mathbf{x}) \leq 0$ can be expressed as:

$$\min_{\mathbf{x}} \max_{\boldsymbol{\lambda}} \Lambda(\mathbf{x}, \boldsymbol{\lambda}) \text{ such that } \boldsymbol{\lambda} \geq 0$$

Instead of using the primal formulation let's consider:

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{x}} \Lambda(\mathbf{x}, \boldsymbol{\lambda}) \text{ such that } \boldsymbol{\lambda} \geq 0$$

This is called the dual

Under certain conditions (convexity) the two problems have the same solution

Back to SVMs

Lagrangian for the SVM problem:

$$\Lambda(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \alpha_i \left[1 - y_i \left(\mathbf{w}^\intercal \mathbf{x}_i + b\right)\right]$$

original constraints:

$$y_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geq 1$$

Necessary conditions for the saddle point:

$$\frac{\partial \Lambda}{\partial \mathbf{w}} = \mathbf{w} + \sum_{i=1}^{n} \alpha_i(-y_i \mathbf{x}_i) = 0$$

$$\Rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial \Lambda}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0$$

How do we get b?

See sections 8.2.2,8.2.3 in Chapter e-8

Support Vectors

Let's use the KKT conditions:

$$\alpha_i \left[1 - y_i \left(\mathbf{w}^\intercal \mathbf{x}_i + b\right)\right] = 0$$

Implication:

Pick an i such that $\alpha_i > 0$

$$y_i \left(\mathbf{w}^\intercal \mathbf{x}_i + b\right) = 1$$

$$\Rightarrow b = y_i - \mathbf{w}^\intercal \mathbf{x}_i$$

Support Vectors

Let's use the KKT conditions:

$$\alpha_i \left[1 - y_i \left(\mathbf{w}^\intercal \mathbf{x}_i + b\right)\right] = 0$$

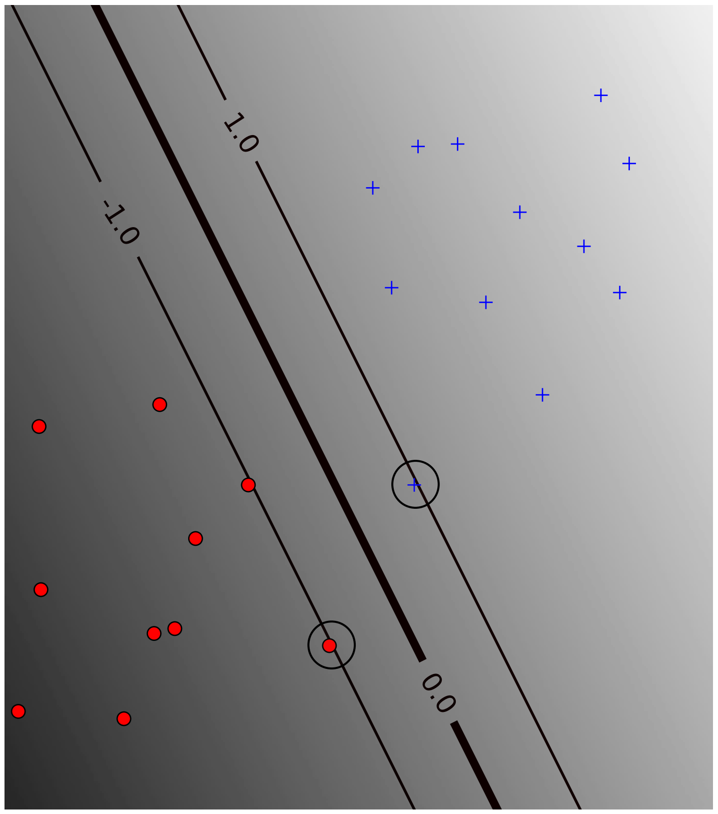
Implication:

Pick an i such that $\alpha_i > 0$

$$y_i \left(\mathbf{w}^\intercal \mathbf{x}_i + b\right) = 1$$

$$\Rightarrow b = y_i - \mathbf{w}^\intercal \mathbf{x}_i$$

The correspond $\mathbf{x}_i$ are called support vectors

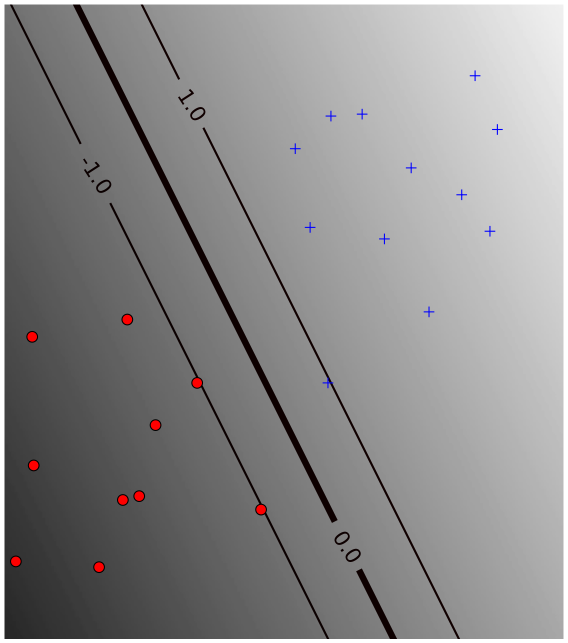


Support Vectors

Claim: The fraction of support vectors is an upper bound on the estimated Leave-One-Out error (see page 17 in chapter 8)

$$E_{\text{cv}}(\text{SVM}) = \frac{1}{N}\sum_{n=1}^{N} \mathsf{e}_n \le \frac{\#\ \text{support vectors}}{N}.$$

Reasoning: if we take out a data point which is not a support vector, the decision boundary remains the same.



The dual

$$\Lambda(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \alpha_i \left[1 - y_i \left(\mathbf{w}^\intercal \mathbf{x}_i + b\right)\right]$$

The dual:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$W(\alpha) = \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\right)^\intercal \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j$$

$$+ \sum_{i=1}^{n} \alpha_i - b \sum_{i=1}^{n} \alpha_i y_i$$

$$- \sum_{i=1}^{n} \alpha_i y_i \left(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^\intercal \mathbf{x}_i\right)$$

The dual

$$
\begin{aligned}
W(\alpha) = & \frac{1}{2}\left(\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i\right)^{\intercal}\sum_{j=1}^{n}\alpha_j y_j \mathbf{x}_j \\
& + \sum_{i=1}^{n}\alpha_i - b\sum_{i=1}^{n}\alpha_i y_i \\
& - \sum_{i=1}^{n}\alpha_i y_i\left(\sum_{j=1}^{n}\alpha_j y_j \mathbf{x}_j^{\intercal}\mathbf{x}_i\right) \\
= & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^{\intercal}\mathbf{x}_j
\end{aligned}
$$

The dual

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\intercal} \mathbf{x}_j$$

$$\text{subject to: } \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Comments: quadratic programming problem (no local minima!)

Usually a sparse solution (many alphas equal to 0)

Compare to the primal:

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2} ||\mathbf{w}||^2$$

$$\text{subject to: } \; y_i(\mathbf{w}^{\intercal}\mathbf{x}_i + b) \geq 1 \quad i = 1, \ldots, n\,.$$

The non-seaprable case: the soft margin SVM

In order to allow for misclassifications we replace the constraints

$$y_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geq 1$$

with
$$y_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geq 1 - \xi_i$$

$\xi_i \geq 0$ are called slack variables

Need to incorporate the slack variables in the optimization problem because we want to discourage their. We'll do it with:

$\displaystyle\sum_{i=1}^{n} \xi_i$ which is a bound on the number of misclassified examples

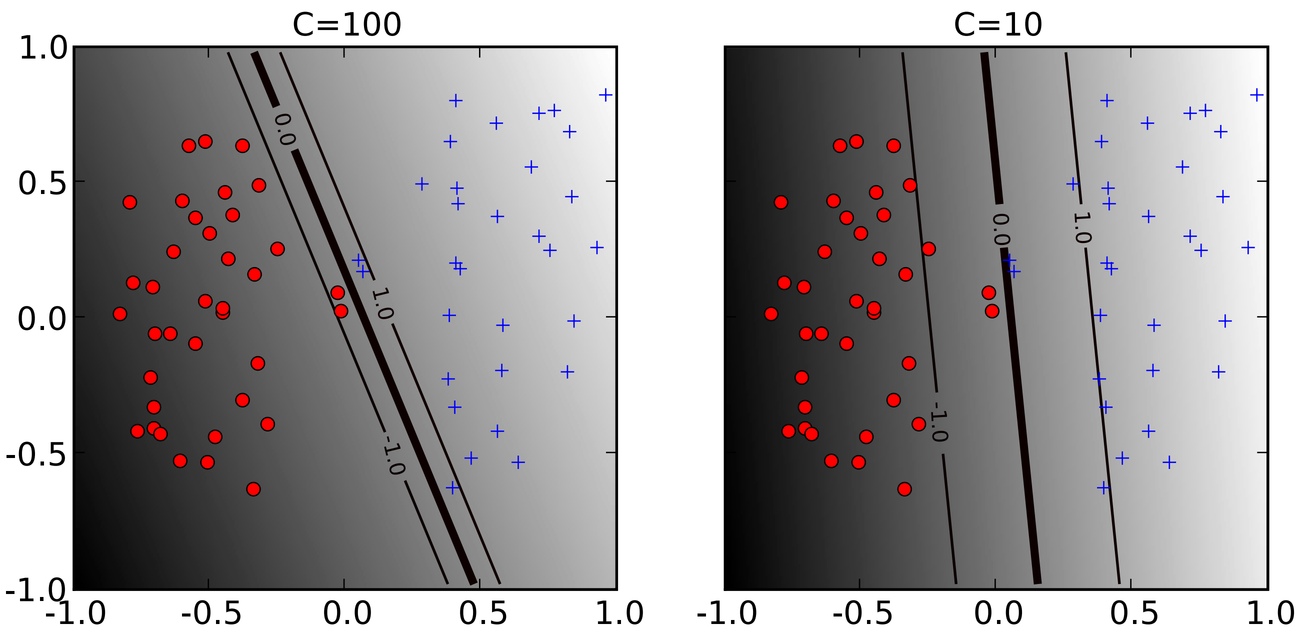
Section 8.4 in Chapter e-8

Soft margin SVM

Our optimization problem for the non-separable case:

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to: } \; y_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geq 1 - \xi_i, \; \xi_i \geq 0, \quad i = 1, \ldots, n\,.$$



Useful even if the data is linearly separable!

SVMs for non-separable data

Our optimization problem for the non-separable case:

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to: } \; y_i(\mathbf{w}^\intercal\mathbf{x}_i + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0, \quad i = 1, \ldots, n\,.$$

Let's form the Lagrangian:

$$\Lambda(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left[1 - \xi_i - y_i\left(\mathbf{w}^\intercal\mathbf{x}_i + b\right)\right] - \sum_{i=1}^{n}\beta_i\xi_i$$

Saddle point equations:

$$\frac{\partial\Lambda}{\partial\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i = 0$$

$$\frac{\partial\Lambda}{\partial b} = \sum_{i=1}^{n} y_i\alpha_i = 0$$

$$\frac{\partial\Lambda}{\partial\xi_i} = C - \alpha_i - \beta_i = 0$$

The dual

Plugging into the Lagrangian we get the following dual formulation:

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\intercal \mathbf{x}_j$$

$$\text{subject to: } \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\beta_i \geq 0, \quad C - \alpha_i - \beta_i = 0$$

Beta appears only in the constraints. Replace it with the constraint

$$0 \leq \alpha_i \leq C$$

The dual

The final form of the dual becomes

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\intercal \mathbf{x}_j$$

$$\text{subject to: } 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

SVM: dual and primal

Primal:

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to:} \;\; y_i(\mathbf{w}^\intercal\mathbf{x}_i + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0, \quad i = 1, \ldots, n\,.$$

Dual:

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\intercal\mathbf{x}_j$$

$$\text{subject to: } 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

The dual has simpler constraints and will allow us to use SVMs as non-linear classifiers

SVM solvers

Primal:

- Fast
- Software:
- liblinear – linear SVMs
- PEGASOS – subgradient descent that also works for nonlinear SVM

Dual:

- You can solve it by generic quadratic programming solvers
- SVM-specific solvers: SMO (optimize two alphas at a time)
- Software: libsvm (a flavor of SMO)

Scikit-learn uses liblinear and libsvm

SMO

Sequential Minimal Optimization (SMO): A solver for the SVM dual problem.

When you choose two variables, the resulting problem can be solved analytically!

Issues and tricks:

- Which two variables to choose?
- Shrinking: temporarily remove variables that are less likely to be chosen (at upper/lower bounds). Need occasional "unshrinking".

Platt, John (1998), *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*

Demo

Show demo in 2-d