



# Quadratic Programming Problem

Formulation:	Formulation:	Quadratic programming with linear constraints				Formulation:
minimize $\frac{1}{2} \ \mathbf{w}\ ^2$	minimize $\frac{1}{2} \ \mathbf{w}\ ^2$	minimize $\frac{1}{2} \ \mathbf{w}\ ^2$ s.t. $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$		minimize $L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2} \ \mathbf{w}\ ^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1)$ s.t. $\alpha_i \geq 0$		minimize $L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2} \ \mathbf{w}\ ^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1)$ s.t. $\alpha_i \geq 0$
such that						
For $y_i = +1$ , $\mathbf{w}^T \mathbf{x}_i + b \geq 1$						
For $y_i = -1$ , $\mathbf{w}^T \mathbf{x}_i + b \leq -1$						
	subject to			$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$		
	$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$			$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0$		
		Lagrangian Function				Lagrangian dual problem
		minimize $L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2} \ \mathbf{w}\ ^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1)$ s.t. $\alpha_i \geq 0$				maximize $\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ s.t. $\alpha_i \geq 0$ , and $\sum_{i=1}^n \alpha_i y_i = 0$

## Linear Discriminant Function

The linear discriminant function is:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{i \in SV} \alpha_i \mathbf{x}_i^T \mathbf{x} + b$$

It is a weighted *dot product* between the test point  $\mathbf{x}$  and the support vectors  $\mathbf{x}_i$

Solving the optimization problem involved computing the *dot products*  $\mathbf{x}_i^T \mathbf{x}_j$  between all pairs of training samples

### Feature Space

General idea: the original input space can be mapped to a higher-dimensional feature space where the training set is linearly separable

- For some functions  $K(\mathbf{x}_i, \mathbf{x}_j)$  checking that  $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$  can be cumbersome.
  - Mercer's theorem:
    - Every semi-positive definite symmetric function is a kernel
    - Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:
- |                                 |                                 |                                 |     |                                 |
|---------------------------------|---------------------------------|---------------------------------|-----|---------------------------------|
| $K(\mathbf{x}_1, \mathbf{x}_1)$ | $K(\mathbf{x}_1, \mathbf{x}_2)$ | $K(\mathbf{x}_1, \mathbf{x}_3)$ | ... | $K(\mathbf{x}_1, \mathbf{x}_N)$ |
| $K(\mathbf{x}_2, \mathbf{x}_1)$ | $K(\mathbf{x}_2, \mathbf{x}_2)$ | $K(\mathbf{x}_2, \mathbf{x}_3)$ |     | $K(\mathbf{x}_2, \mathbf{x}_N)$ |
| ...                             | ...                             | ...                             | ... | ...                             |
| $K(\mathbf{x}_P, \mathbf{x}_1)$ | $K(\mathbf{x}_N, \mathbf{x}_2)$ | $K(\mathbf{x}_N, \mathbf{x}_3)$ | ... | $K(\mathbf{x}_N, \mathbf{x}_N)$ |

## SVM Learning

- Choose a kernel function
- Choose a value for  $C$
- Solve the quadratic programming problem (many software packages available)
- Construct the discriminant function from the support vectors

- Choice of kernel
  - Gaussian or polynomial kernel is default
  - if ineffective, more elaborate kernels are needed
  - domain experts can give assistance in formulating appropriate similarity measures
- Choice of kernel parameters
  - e.g.  $\sigma$  in Gaussian kernel
  - $\sigma$  is the distance between closest points with different classifications
  - In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters.
- Optimization criterion – Hard margin v.s. Soft margin
  - a lengthy series of experiments in which various parameters are tested

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) - \sum_{i=1}^l (\eta_i \xi_i + \eta_i^* \xi_i^*)$$
$$- \sum_{i=1}^l \alpha_i (\varepsilon + \xi_i - y_i + \mathbf{w} \cdot \mathbf{x}_i + b)$$
$$- \sum_{i=1}^l \alpha_i^* (\varepsilon + \xi_i^* + y_i - \mathbf{w} \cdot \mathbf{x}_i - b)$$

$$\min_{\mathbf{w}, b, \xi} \max_{\alpha, \eta} L$$
$$\text{subject to } \alpha, \eta \geq 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^l (\alpha_i^* - \alpha_i) = 0$$
$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^l (\alpha_i - \alpha_i^*) \mathbf{x}_i = 0$$
$$\frac{\partial L}{\partial \xi_i^{(*)}} = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0$$

$$\max_{\alpha, \eta} -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \mathbf{x}_i \cdot \mathbf{x}_j - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*)$$
$$\text{subject to } \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]$$

## Least-squares SVM

Proposed by Suykens and Vandewalle at KUL in 1999.

Let  $y_i = \mathbf{w}^T \varphi(\mathbf{x}_i) + b + e_i$

Primal problem formulation

$$\min_{\mathbf{w}, b, e_i} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2$$
$$y_i = \mathbf{w}^T \varphi(\mathbf{x}_i) + b + e_i, \quad i = 1, \dots, N.$$

## Least-squares SVM

Lagrangian

$$\mathcal{L} = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2 - \sum_{i=1}^N \alpha_i (\mathbf{w}^T \varphi(\mathbf{x}_i) + b + e_i - y_i)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0, \frac{\partial \mathcal{L}}{\partial b} = 0, \frac{\partial \mathcal{L}}{\partial e_i} = 0, \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \text{ gives}$$

$$\left[ \begin{array}{c|c} \Omega + \gamma^{-1} \mathbf{I} & \mathbf{1} \\ \hline \mathbf{1}^T & 0 \end{array} \right] \left[ \begin{array}{c} \alpha \\ b \end{array} \right] = \left[ \begin{array}{c} \mathbf{y} \\ 0 \end{array} \right]$$

## Maximal Margin Classifier

### Soft Constraints

Problem formulation:

$$\text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$
$$\text{subject to}$$
$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

Parameter C is a weight between marginal maximization and misclassification minimization.

Problem reformulation: (Lagrangian dual problem)

$$\text{maximize } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$
$$\text{Subject to}$$
$$0 \leq \alpha_i \leq C$$
$$\sum_{i=1}^n \alpha_i y_i = 0$$

## Nonlinear SVM: Optimization

Problem reformulation: (Lagrangian dual problem)

$$\text{maximize } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$
$$\text{subject to } 0 \leq \alpha_i \leq C$$
$$\sum_{i=1}^n \alpha_i y_i = 0$$

The solution of the discriminant function is

$$g(\mathbf{x}) = \sum_{i \in SV} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$$

The optimization technique is the same.

### Design Issues

### SVM Summary

- Maximal Margin Classifier
  - Better generalization ability & less over-fitting
- The Kernel Trick
  - Map data points to a higher dimensional space to make them linearly separable.
  - Since only dot product is used, we do not need to represent the mapping explicitly.

### Support Vector Regression

Problem formulation

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

$$\text{subject to } \|y_i - (\mathbf{w} \cdot \mathbf{x}_i - b)\| \leq \varepsilon$$

### Support Vector Regression

Problem reformulation

$$\text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$$
$$\text{subject to } \begin{cases} y_i - \mathbf{w} \cdot \mathbf{x}_i - b \leq \varepsilon + \xi_i \\ \mathbf{w} \cdot \mathbf{x}_i + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

## Least-squares SVM

Final model

$$y(\mathbf{x}) = \sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$$