Kernel Methods

Introduction to Machine Learning – GIF-7015

Professor : Christian Gagné

Week 6

UNIVERSITÉ LAVAL

6.1 Review of linear discriminants

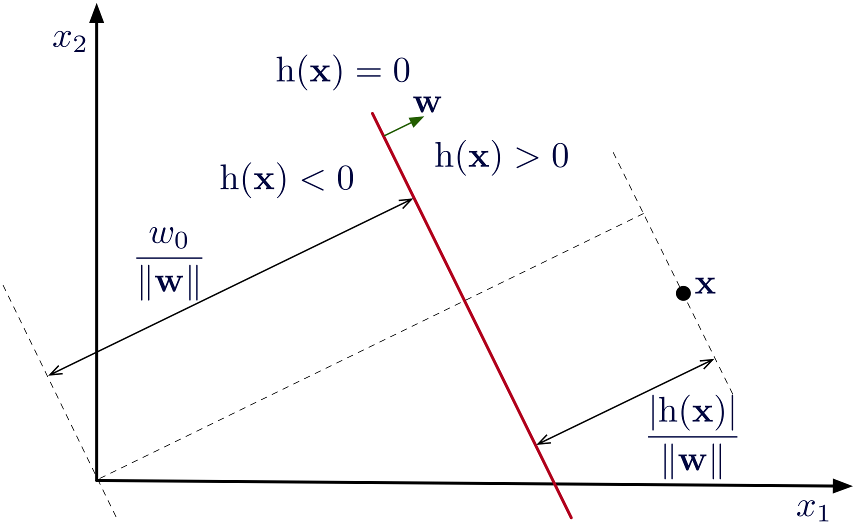
Linear discriminants

- Equation of a linear discriminant

$$\mathrm{h}_i(\mathbf{x}|\mathbf{w}_i,w_{i,0}) = \sum_{j=1}^{D} w_{i,j}x_j + w_{i,0}$$

- Two-class model
  - Only one equation $\mathrm{h}(\mathbf{x}|\mathbf{w},w_0)$
  - if $\mathrm{h}(\mathbf{x}) \geq 0$ then $\mathbf{x}$ belongs to C_1
  - Otherwise (when $\mathrm{h}(\mathbf{x}) < 0$) $\mathbf{x}$ belongs to C_2
  - Weight $\mathbf{w}$ determines the orientation of the separating hyperplane
  - Bias w_0 determines the position of the separating hyperplane in the input space

Geometry of linear discriminants



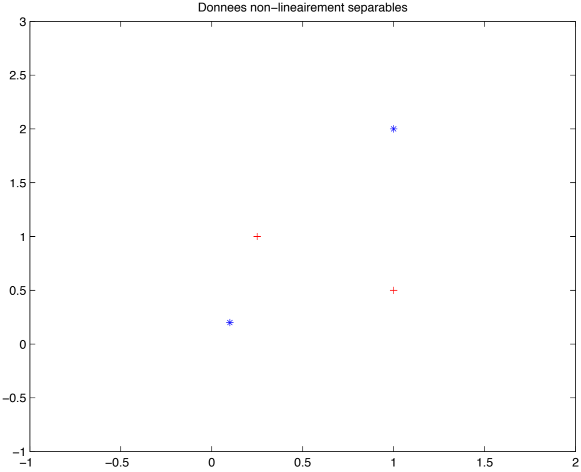
Perceptron criterion

- Perceptron criterion

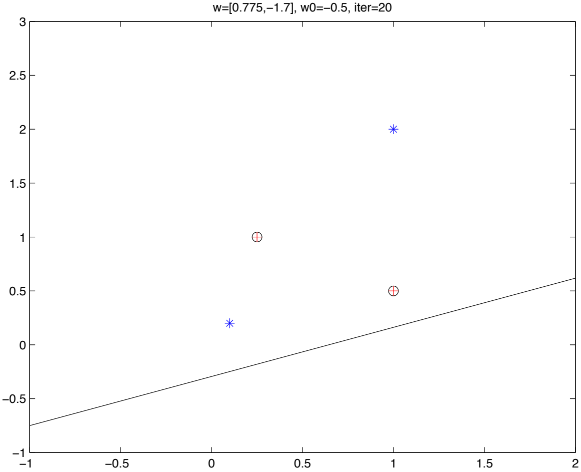
$$
\begin{aligned}
E_{percp}(\mathbf{w}, w_0 | \mathcal{X}) &= -\sum_{\mathbf{x}^t \in \mathcal{Y}} r^t \mathrm{h}(\mathbf{x}^t | \mathbf{w}, w_0) \\
\mathcal{Y} &= \{\mathbf{x}^t \in \mathcal{X} | r^t \mathrm{h}(\mathbf{x}^t | \mathbf{w}, w_0) < 0\}
\end{aligned}
$$

- Weak link between the error and the nature of the errors
  - The classifier may diverge on nonlinearly separable data

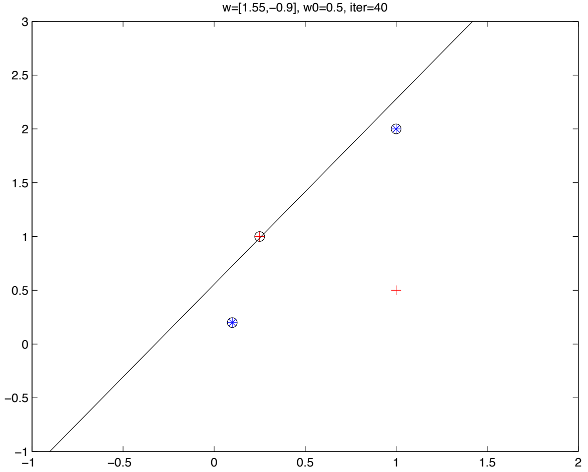
Perceptron on nonlinearly separable data



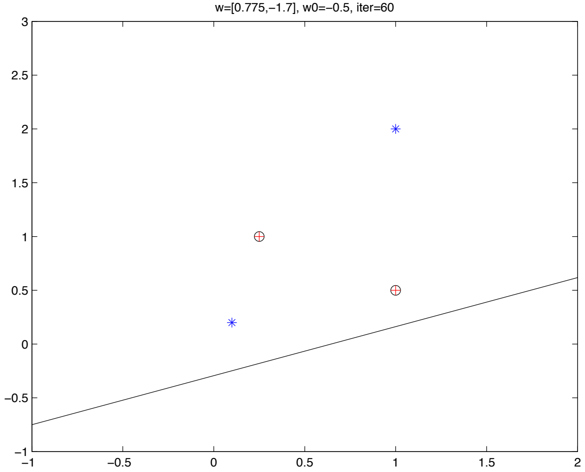
Perceptron on nonlinearly separable data



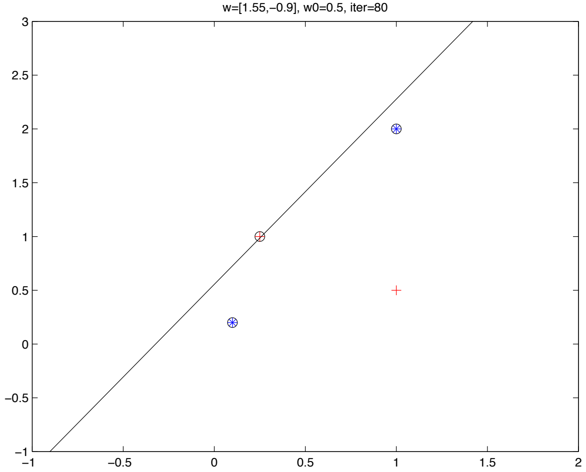
Perceptron on nonlinearly separable data



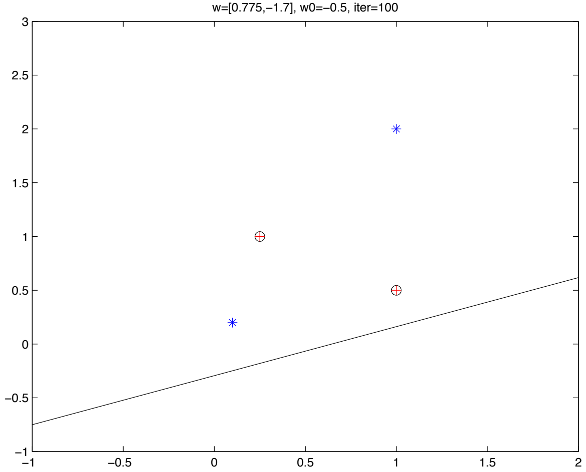
Perceptron on nonlinearly separable data



Perceptron on nonlinearly separable data



Perceptron on nonlinearly separable data



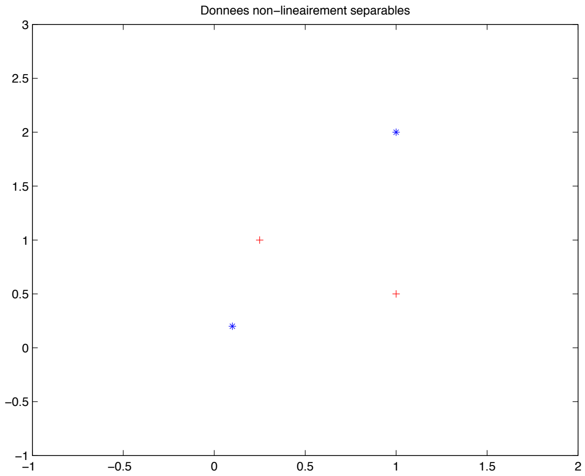
Least squares criterion

- Least squares criterion: regression for classification

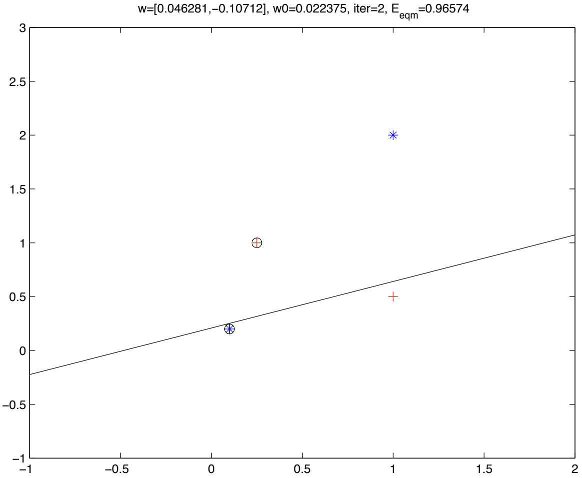
$$E_{quad}(\mathbf{w}, w_0|\mathcal{X}) = \frac{1}{2} \sum_{\mathbf{x}^t \in \mathcal{X}} (r^t - (\mathbf{w}^\top \mathbf{x}^t + w_0))^2$$

- Tends to minimize the distance from the $\mathrm{h}(\mathbf{x})$ to the r^t value.
  - Better management of nonlinearly separable data
  - Emphasis on data far from the separating hyperplane

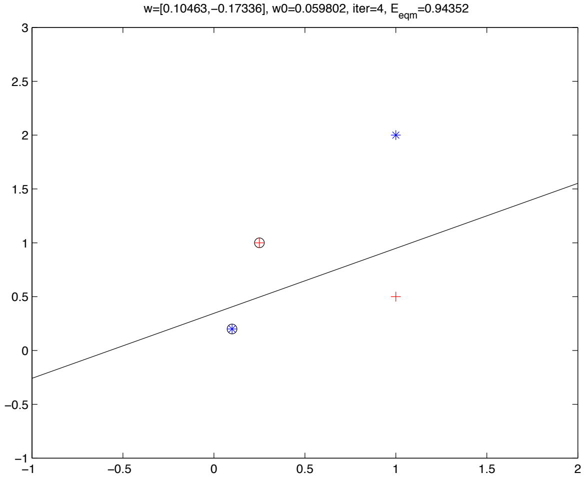
Least squares on nonlinearly separable data



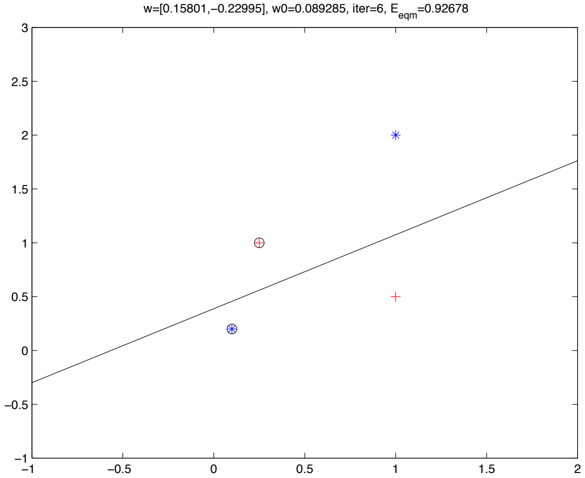
Least squares on nonlinearly separable data



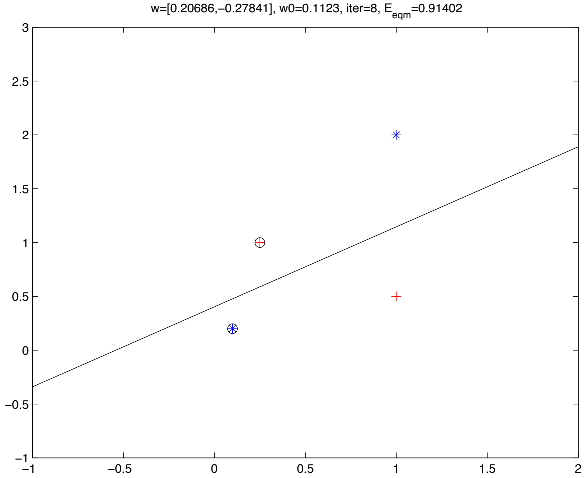
Least squares on nonlinearly separable data



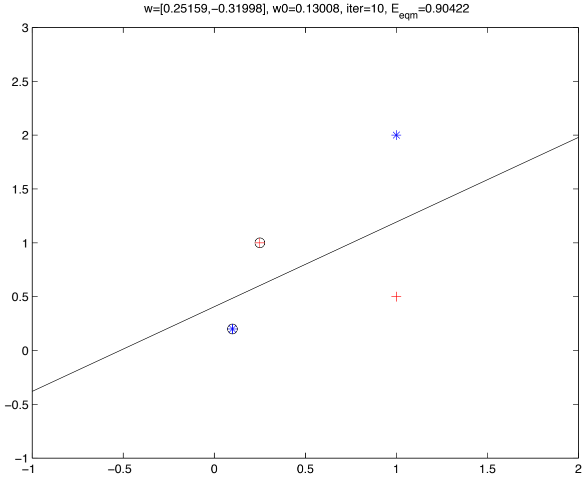
Least squares on nonlinearly separable data



Least squares on nonlinearly separable data



Least squares on nonlinearly separable data

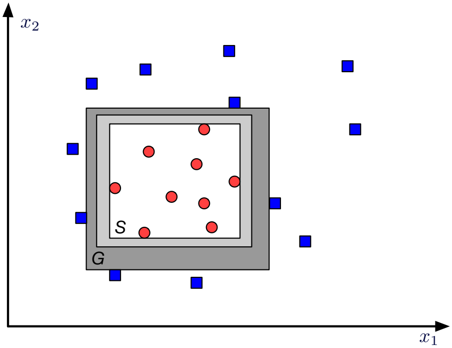


6.2 Support vector machine

Support vector machine (SVM)

- SVM: Support vector machine
- Maximization of geometric margins
  - Aims to find the optimal position for the separating hyperplane
  - It is argued in computational learning theory that this criterion minimizes error (cf. version space)
- Development for a linear discriminant
  - Can be extended to nonlinear models by using kernel functions

Version space



- G: most general hypothesis
- S: most specific hypothesis
- Hypotheses in $\mathcal{H}$ between S and G are part of the *version space*

Maximization of geometric margins

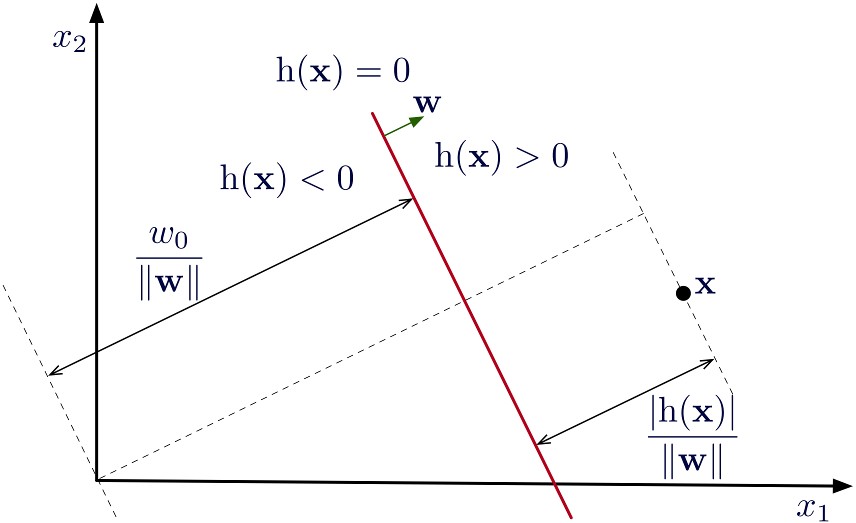
- Searching for weights $\mathbf{w}$ and w_0 maximizing the geometric margin for a dataset $\mathcal{X} = \{\mathbf{x}^t, r^t\}$, where $r^t \in \{-1, +1\}$
- Distances to the data separating hyperplane

$$\frac{|\mathbf{w}^\top \mathbf{x}^t + w_0|}{\|\mathbf{w}\|} = \frac{r^t(\mathbf{w}^\top \mathbf{x}^t + w_0)}{\|\mathbf{w}\|}$$

- We want this distance to be greater than a ρ threshold (margin) for all data

$$\frac{r^t(\mathbf{w}^\top \mathbf{x}^t + w_0)}{\|\mathbf{w}\|} \geq \rho, \ \forall t$$

Linear discriminants geometry



Maximizing geometric margins

- $\mathbf{w}^\top \mathbf{x}^t + w_0$ is undetermined, there are an infinity of solutions

$$\underset{w_0 = 1}{\mathbf{w}^\top = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix}} \equiv \underset{w_0 = 0.5}{\mathbf{w}^\top = \begin{bmatrix} 1 \\ 0.25 \end{bmatrix}} \equiv \underset{w_0 = 10}{\mathbf{w}^\top = \begin{bmatrix} 20 \\ 5 \end{bmatrix}}$$

- We set $\rho\|\mathbf{w}\| = 1$, which gives:

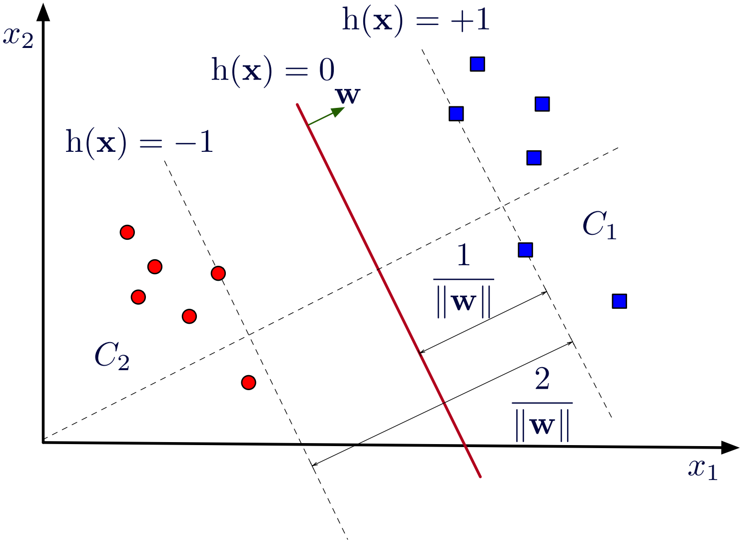
$$\begin{aligned} \mathbf{w}^\top \mathbf{x}^t + w_0 &\geq +1 \quad \text{for} \quad r^t = +1 \\ \mathbf{w}^\top \mathbf{x}^t + w_0 &\leq -1 \quad \text{for} \quad r^t = -1 \end{aligned}$$

- Equivalent formulation

$$r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) \geq +1$$

- Minimizing $\|\mathbf{w}\|$ allows to maximize the ρ margin

Maximizing geometric margins



6.3 SVM optimization problem

Lagrange multipliers

- Method for solving optimization problems under constraints
  - Example: maximize $f(\mathbf{x})$ under constraint that $g(\mathbf{x}) = 0$
  - There is a parameter $\lambda \neq 0$ that allows to obtain

$$\nabla f + \lambda \nabla g = 0$$

  - Corresponding equation with Lagrange multiplier

$$L(\mathbf{x},\lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$$

- Maximum obtained by solving $\nabla L(\mathbf{x},\lambda) = 0$
  - If we are only interested in $\mathbf{x}$, we can eliminate λ without having to evaluate it

Example with the Lagrange multiplier

- Maximize $f(x_1,x_2) = 1 - x_1^2 - x_2^2$ subject to constraint $g(x_1,x_2) = x_1 + x_2 - 1 = 0$
- Formulation with Lagrange multiplier

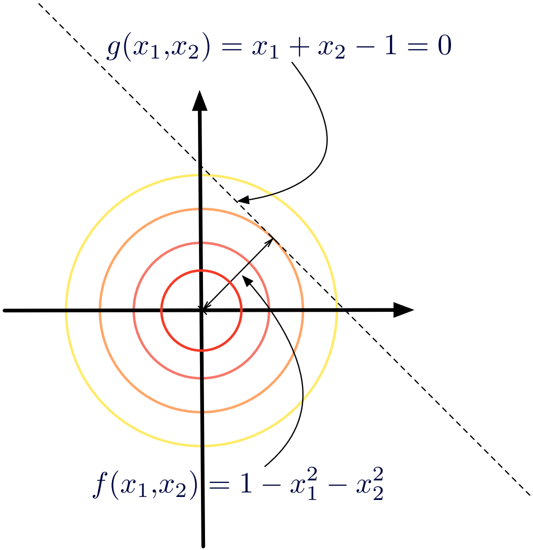
$$L(x_1,x_2,\lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1)$$

- Solve $\nabla L(x_1,x_2,\lambda) = 0$

$$\begin{aligned}
\frac{\partial L}{\partial x_1} &= -2x_1 + \lambda = 0 \\
\frac{\partial L}{\partial x_2} &= -2x_2 + \lambda = 0 \\
\frac{\partial L}{\partial \lambda} &= x_1 + x_2 - 1 = 0
\end{aligned}$$

- Solution to the system of equations: $x_1 = 0.5$, $x_2 = 0.5$ and $\lambda = 1$

Example with the Lagrange multiplier



Lagrange multipliers with inequalities

- If constraints are inequalities $g(\mathbf{x}) \geq 0$
  - Possibility 1: inactive constraint, $f(\mathbf{x})$ is maximum for $g(\mathbf{x}) > 0$, so maximum at $\nabla f(\mathbf{x}) = 0$, which implies $\lambda = 0$
  - Possibility 2: active constraint, $f(\mathbf{x})$ is maximum for $g(\mathbf{x}) = 0$
    - In that case, $\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x})$ and $\lambda > 0$
- Corresponding conditions (Karush-Kuhn-Tucker)

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

- Formulation where we minimize $f(\mathbf{x})$, subject to $g(\mathbf{x}) \geq 0$ (subtraction of the constraint)

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x}), \text{ with } \lambda \geq 0$$

Formulation of the SVM optimization problem

- SVM optimization problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) \geq +1, \ \forall t \end{aligned}$$

- Typical form of a quadratic programming problem
  - Methods (and solvers) exist to find an exact resolution for this problem
- Reformulation of the problem using Lagrange multipliers (α^t)

$$\begin{aligned} L_p &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_t \alpha^t [r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) - 1] \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_t \alpha^t r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) + \sum_t \alpha^t \end{aligned}$$

Primal and dual formulations

- L_p is the primal formulation of the problem

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_t \alpha^t r^t (\mathbf{w}^\top \mathbf{x}^t + w_0) + \sum_t \alpha^t$$

- Resolution of L_p require to minimize according to $\{\mathbf{w}, w_0\}$ and maximize according to $\alpha^t \geq 0$
    - Saddle point solution according to $\{\mathbf{w}, w_0\}$ and α^t
- Simplification by dual formulation of the problem
    - Eliminate $\mathbf{w}$ using the partial derivatives of L_p according to $\{\mathbf{w}, w_0\}$ equal to zero

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0, \quad \frac{\partial L_p}{\partial w_0} = 0$$

Passing to dual formulation

$$
\begin{aligned}
L_p &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_t \alpha^t r^t (\mathbf{w}^\top \mathbf{x}^t + w_0) + \sum_t \alpha^t \\
\frac{\partial L_p}{\partial \mathbf{w}} &= \mathbf{w} - \sum_t \alpha^t r^t \mathbf{x}^t = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_t \alpha^t r^t \mathbf{x}^t \\
\frac{\partial L_p}{\partial w_0} &= \sum_t \alpha^t r^t = 0 \\
L_d &= \frac{1}{2}(\mathbf{w}^\top \mathbf{w}) - \mathbf{w}^\top \sum_t \alpha^t r^t \mathbf{x}^t - w_0 \sum_t \alpha^t r^t + \sum_t \alpha^t \\
&= -\frac{1}{2}(\mathbf{w}^\top \mathbf{w}) + \sum_t \alpha^t \\
&= -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\mathbf{x}^t)^\top \mathbf{x}^s + \sum_t \alpha^t
\end{aligned}
$$

Problem formulation with Lagrange multipliers

- Dual formulation with Lagrange multipliers

$$\begin{aligned} \text{maximize} \quad & -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\mathbf{x}^t)^\top \mathbf{x}^s + \sum_t \alpha^t \\ \text{subject to} \quad & \sum_t \alpha^t r^t = 0 \quad \text{and} \quad \alpha^t \geq 0, \forall t \end{aligned}$$

- New problem formulation
  - Problem size depends on the size of the dataset (N) rather than on the dimensionality (D)
- Form always resolvable by quadratic programming
  - Guarantee to obtain the global optimum in polynomial time
  - Complexity in time $O(N^3)$, complexity in space $O(N^2)$.
- This formulation allows to use kernel functions (presented later this week)

Support vectors

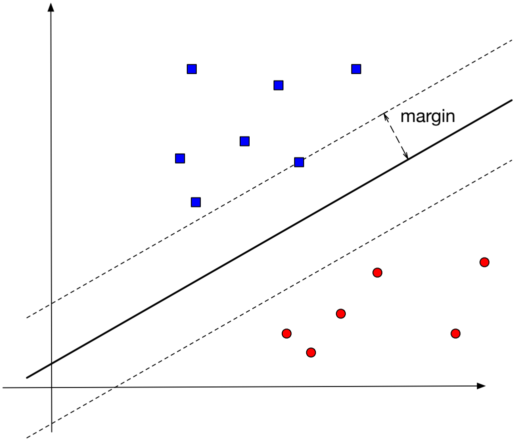
- New formulation: one α^t per training data
  - Usually, a majority of $\alpha^t = 0$
  - The data for which $\alpha^t > 0$ are the *support vectors*
- Calculation of w_0 from the support vectors, $\mathcal{M} = \{\alpha^t | \alpha^t > 0, \forall t\}$

$$w_0 = \mathbb{E}[r^t - \mathbf{w}^\top \mathbf{x}^t] = \frac{1}{|\mathcal{M}|} \sum_{\alpha^t \in \mathcal{M}} \left(r^t - \sum_{\alpha^s \in \mathcal{M}} \alpha^s r^s (\mathbf{x}^t)^\top \mathbf{x}^s \right)$$

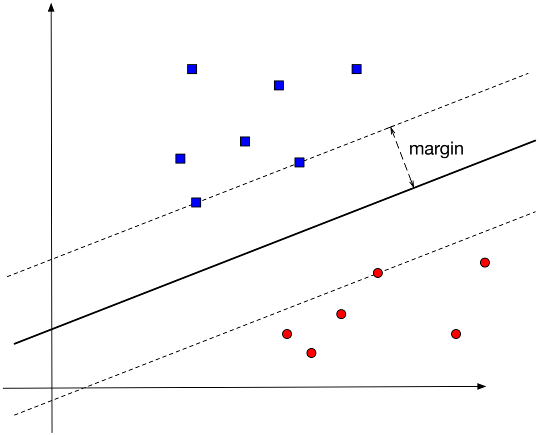
- Post-training data evaluation

$$\mathrm{h}(\mathbf{x}) = \sum_t \alpha^t r^t (\mathbf{x}^t)^\top \mathbf{x} + w_0$$

Illustration of support vectors



Sub-optimal margin



Maximum margin

6.4 Soft margins

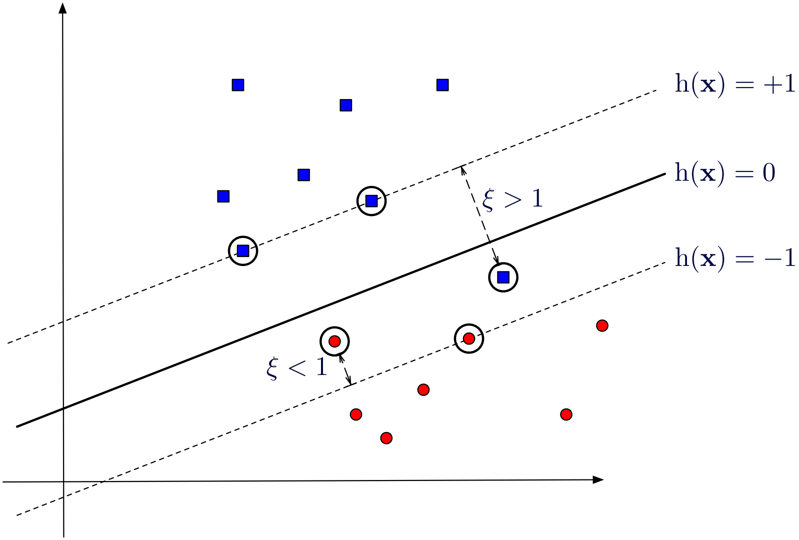
Soft margins

- Under the current formulation, SVM remains a linear discriminant
  - With nonlinearly separable data, no valid solution can be obtained by quadratic programming
- Introduction of *slacks* variables ($\xi^t \geq 0$) for each data $\mathbf{x}^t$
  - If $\xi^t = 0$, no problem with the $\mathbf{x}^t$ variable
  - If $\xi^t > 0$, deviation of the $\mathbf{x}^t$ variable from the margin
    - $0 < \xi^t < 1$: data on the right side, but in the margin
    - $\xi^t > 1$: data on the wrong side of the hyperplane, misclassified
  - Rewriting the SVM optimization criterion

$$r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) \geq 1 - \xi^t$$

- Allows error tolerance
  - Error associated with data in the margin: $\sum_t \xi^t$

Soft margins



Reformulation with soft margins

- Primal formulation with soft margins

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_t \xi^t - \sum_t \alpha^t[r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) - 1 + \xi^t] - \sum_t \mu^t \xi^t$$

  - μ^t: Lagrange multipliers for constraints $\xi^t \geq 0$
  - C: Penalty factor for regularization according to errors ξ^t

- Dual formulation with soft margins

$$\begin{aligned} &\text{maximize} \quad && -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\mathbf{x}^t)^\top \mathbf{x}^s + \sum_t \alpha^t \\ &\text{subject to} \quad && \sum_t \alpha^t r^t = 0 \quad \text{and} \quad 0 \leq \alpha^t \leq C, \forall t \end{aligned}$$

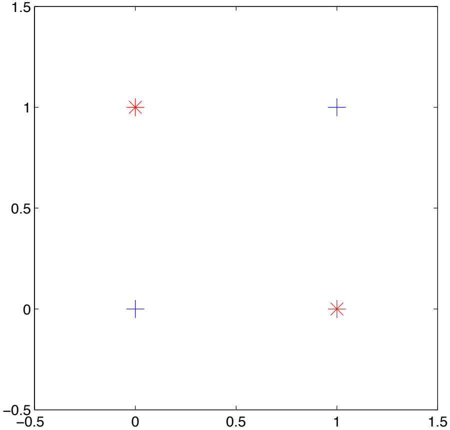
6.5 Basis functions review

XOR problem

- XOR problem

$$
\begin{aligned}
\mathbf{x}_1 &= [0\ 0]^\top & r_1 &= 0 \\
\mathbf{x}_2 &= [0\ 1]^\top & r_2 &= 1 \\
\mathbf{x}_3 &= [1\ 0]^\top & r_3 &= 1 \\
\mathbf{x}_4 &= [1\ 1]^\top & r_4 &= 0
\end{aligned}
$$

- Example of nonlinearly separable data



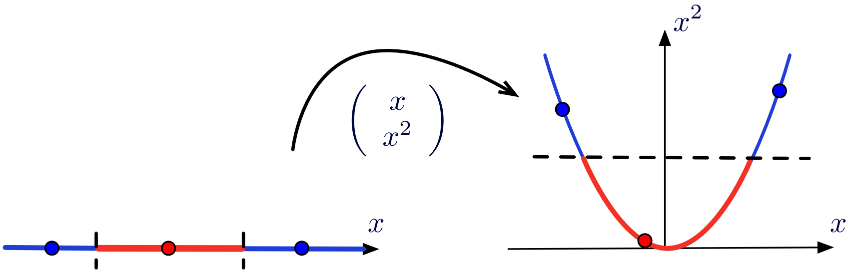
Basis functions

- Discriminant with basis function
  - Nonlinear transformation $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^K$ written in a linear form

$$\mathrm{h}_i(\mathbf{x}) = \sum_{j=1}^{K} w_j \phi_{i,j}(\mathbf{x}) + w_0$$

  - Example of basis functions
    - $\phi_{i,j}(\mathbf{x}) = x_j$
    - $\phi_{i,j}(\mathbf{x}) = x_1^{j-1}$
    - $\phi_{i,j}(\mathbf{x}) = \exp(-(x_2 - m_j)^2/c)$
    - $\phi_{i,j}(\mathbf{x}) = \exp(-\|\mathbf{x} - \mathbf{m}_j\|^2/c)$
    - $\phi_{i,j}(\mathbf{x}) = \mathrm{sgn}(x_j - c_j)$

Projection with a basis function



- In 1D: nonlinearly separable
- With 2D projection: linearly separable

Basis functions

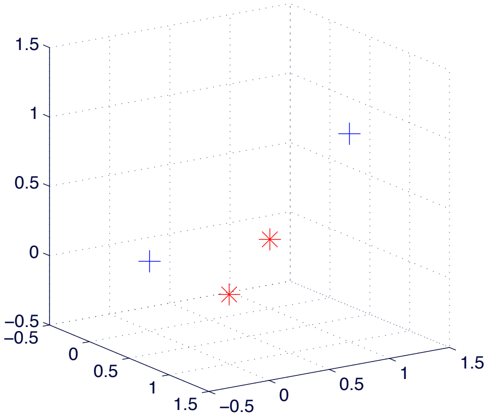
- Resolution of the XOR problem with a basis function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$

$$\phi(\mathbf{x}) = [x_1 \ x_2 \ (x_1 x_2)]^\top$$

- Transformation results

$$\begin{aligned}
\mathbf{z}_1 &= [0\ 0\ 0]^\top & r_1 &= 0\\
\mathbf{z}_2 &= [0\ 1\ 0]^\top & r_2 &= 1\\
\mathbf{z}_3 &= [1\ 0\ 0]^\top & r_3 &= 1\\
\mathbf{z}_4 &= [1\ 1\ 1]^\top & r_4 &= 0
\end{aligned}$$

- Data is linearly separable in the new space!



Radial Basis Functions

- Radial Basis Functions (RBF)

$$\phi_i(\mathbf{x}) = \exp\left[-\frac{\|\mathbf{x} - \mathbf{m}_i\|^2}{2s_i^2}\right]$$

- Consists of a Gaussian function centered on $\mathbf{m}_i$ with a local influence parameterized by s_i
  - Strictly speaking, this is not a probability density for a multivariate law ($\int_{-\infty}^{\infty} \phi_i(\mathbf{x}) d\mathbf{x} \neq 1$)
- The idea is: each Gaussian function captures a group of data in a certain neighbourhood
- With K Gaussian functions, projection in a space with K dimensions

$$\phi = [\phi_1 \ \ldots \ \phi_K]^\top : \mathbb{R}^D \to \mathbb{R}^K$$

6.6 Kernel SVM

Basis functions and SVM

- Nonlinear transformation $\phi : \mathbb{R}^D \to \mathbb{R}^K$ with basis functions

$$\mathbf{z}(\mathbf{x}) = \phi(\mathbf{x})$$

- Linear discrimination in nonlinear space

$$\begin{aligned} \mathrm{h}(\mathbf{z}) &= \mathbf{w}^\top \mathbf{z} + w_0 \\ &= \mathbf{w}^\top \phi(\mathbf{x}) + w_0 = \sum_{j=1}^{K} w_j \phi_j(\mathbf{x}) + w_0 \end{aligned}$$

- Reformulation in dual form

$$\begin{aligned} \mathbf{w} &= \sum_t \alpha^t r^t \mathbf{z}^t = \sum_t \alpha^t r^t \phi(\mathbf{x}^t) \\ \mathrm{h}(\mathbf{x}) &= \sum_t \mathbf{w}^\top \phi(\mathbf{x}) + w_0 = \sum_t \alpha^t r^t (\phi(\mathbf{x}^t))^\top \phi(\mathbf{x}) + w_0 \end{aligned}$$

Kernel functions

- Kernel function: $K(\mathbf{x},\mathbf{y}) = (\phi(\mathbf{x}))^\top \phi(\mathbf{y})$
- SVM with kernel function

$$\mathrm{h}(\mathbf{x}) = \sum_t \alpha^t r^t K(\mathbf{x}^t,\mathbf{x}) + w_0$$

- Kernel trick: no computation directly in the space generated by $\phi(\mathbf{x})$
    - Allows to process kernel functions generating high dimensionality spaces (possibly infinite), without working directly in these spaces.
- Commonly used kernels
    - Scalar product: $K(\mathbf{x},\mathbf{y}) = <\mathbf{x},\mathbf{y}> = \mathbf{x}^\top \mathbf{y}$
    - Polynomial of order q: $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^q$
    - Gaussian: $K(\mathbf{x},\mathbf{y}) = \exp\left[-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right]$
    - Sigmoid: $K(\mathbf{x},\mathbf{y}) = \tanh(2\mathbf{x}^\top \mathbf{y} + 1)$

SVM kernel

- Training on a dataset $\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$
  - Calculation of α^t by quadratic programming

$$\begin{aligned} \text{maximize} \qquad & L_d = -\frac{1}{2}\sum_{t=1}^{N}\sum_{s=1}^{N}\alpha^t\alpha^s r^t r^s K(\mathbf{x}^t, \mathbf{x}^s) + \sum_t \alpha^t \\ \text{subject to} \qquad & \sum_t \alpha^t r^t = 0 \quad \text{and} \quad 0 \leq \alpha^t \leq C, \forall t \end{aligned}$$

  - Calculation of the bias w_0 with support vectors, $\mathcal{M} = \{\alpha^t | \alpha^t \geq 0, \forall t\}$

$$w_0 = \frac{1}{|\mathcal{M}|}\sum_{\alpha^t \in \mathcal{M}}\left(r^t - \sum_{\alpha^s \in \mathcal{M}}\alpha^s r^s K(\mathbf{x}^t, \mathbf{x}^s)\right)$$

- Evaluating a data $\mathbf{x}$

$$\mathrm{h}(\mathbf{x}) = \sum_t \alpha^t r^t K(\mathbf{x}^t, \mathbf{x}) + w_0$$

Polynomial kernel

- Polynomial kernel of order q

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^q$$

- Example in dimension $D = 2$ and order $q = 2$

$$\begin{aligned} K(\mathbf{x},\mathbf{y}) &= (\mathbf{x}^\top \mathbf{y} + 1)^2 = (x_1 y_1 + x_2 y_2 + 1)^2 \\ &= 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2 \end{aligned}$$

  - Corresponding basis functions

$$\begin{aligned} \phi(\mathbf{x}) &= [1 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2 \;\; \sqrt{2}x_1 x_2 \;\; x_1^2 \;\; x_2^2]^\top \\ K(\mathbf{x},\mathbf{y}) &= \phi(\mathbf{x})^\top \phi(\mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^2 \end{aligned}$$

Gaussian kernel

- Gaussian kernel with spread σ

$$K(\mathbf{x},\mathbf{y}) = \exp\left[-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right]$$

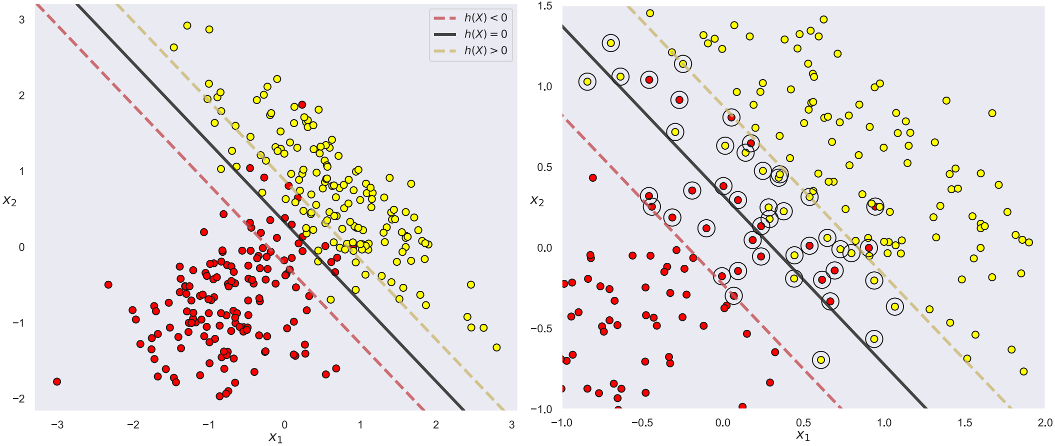
- SVM with Gaussian kernel is a network of RBF functions trained in a particular way

$$\mathrm{h}(\mathbf{x}) = \sum_{t=1}^{N} \alpha^t r^t K(\mathbf{x}^t,\mathbf{x}) + w_0 = \sum_{t=1}^{N} w_t \exp\left[-\frac{\|\mathbf{x}-\mathbf{x}^t\|^2}{\sigma^2}\right] + w_0$$

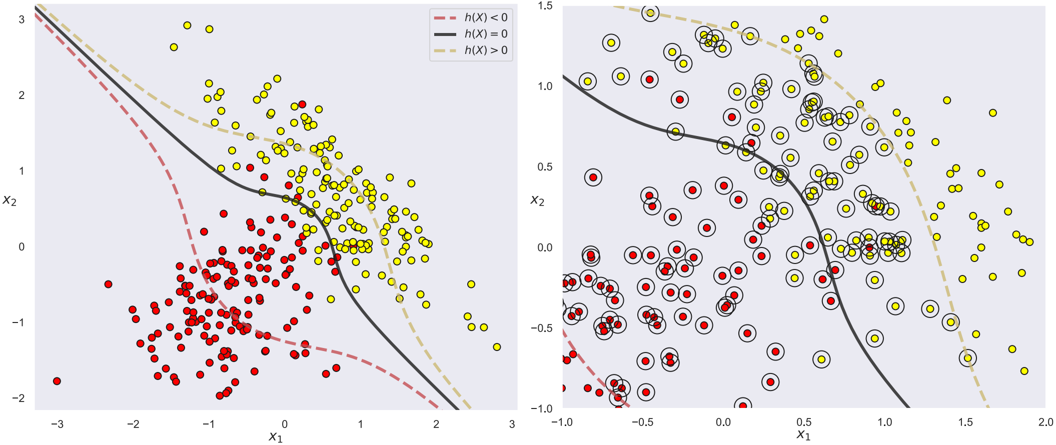
- Density estimation with the kernel method: SVM with kernel and $\alpha^t = 1, \forall t$

$$\mathrm{h}(\mathbf{x}) = \sum_{t=1}^{N} r^t K(\mathbf{x}^t,\mathbf{x})$$

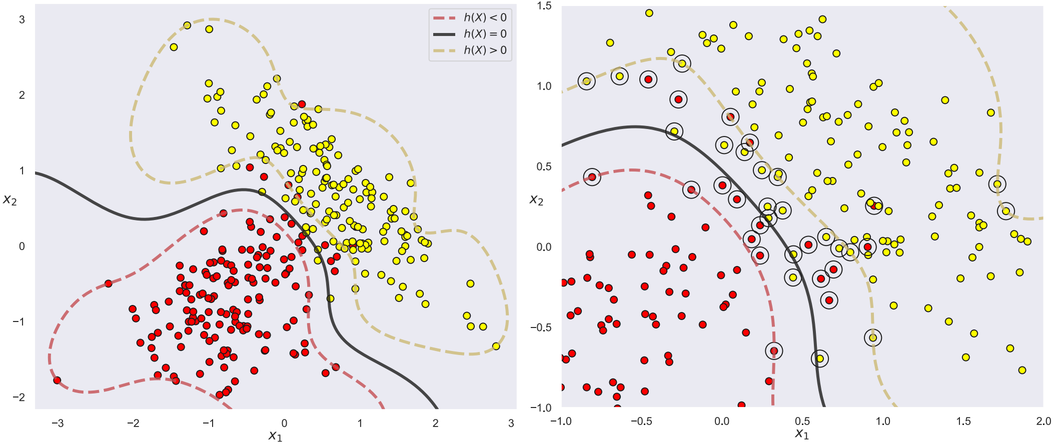
Overlapping data: linear SVM



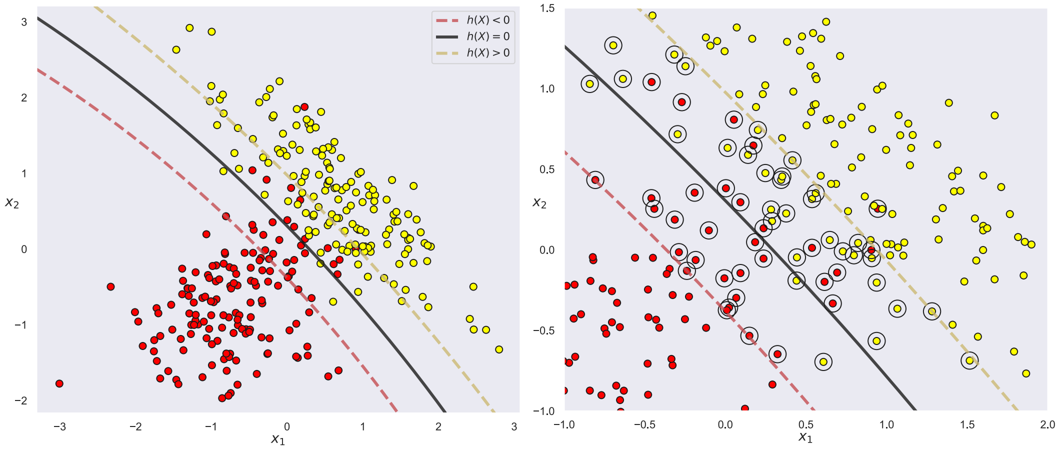
Overlapping data: polynomial kernel



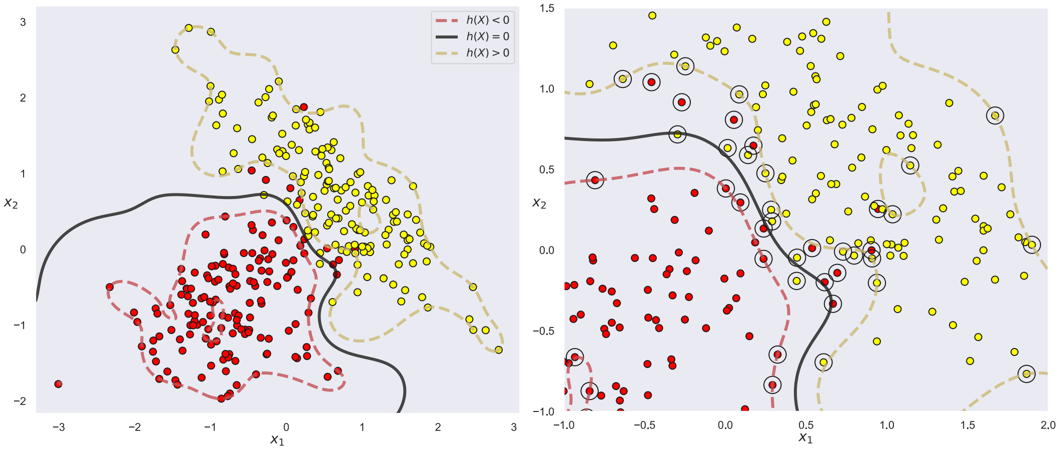
Overlapping data: Gaussian kernel



Overlapping data: Gaussian kernel with large σ



Overlapping data: Gaussian kernel with small σ



6.7 SVM hyperparameters

SVM parameters

- SVM is a complex machine, where the choice of parameters can greatly influence the results.
  - With Gaussian kernel, parameters C (regularization) and σ (kernel reach) have a significant impact on performance
  - For different values of these parameters, results can vary greatly (and sometimes be catastrophic)
  - Empirical adjustment is required, case by case
- Rule of thumb for SVM training with Gaussian kernel
  - Values to be tested for parameter C: $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$
  - Values to be tested for parameter σ: $\{\sigma_{\mathrm{min}}, 2\sigma_{\mathrm{min}}, 4\sigma_{\mathrm{min}}, \ldots, 64\sigma_{\mathrm{min}}\}$ where σ_{min} is the minimum Euclidean distance measured between two data in the data set (excluding zero distances): $\sigma_{\mathrm{min}} = \min_{\forall \mathbf{x}^i \neq \mathbf{x}^j} \|\mathbf{x}^i - \mathbf{x}^j\|$
- Adjustment of these parameters with a grid search

Grid search

- Grid search: adjustment of pairs of parameters, based on results from a validation dataset
  1. Partition the dataset $\mathcal{X}$ into two subsets, $\mathcal{X}_T$ and $\mathcal{X}_V$ (usually 50%-50%)
  2. Train a classifier with $\mathcal{X}_T$ for each pair of parameters considered
  3. Select the pair of parameters where the error is minimal on $\mathcal{X}_V$
  4. Use this pair of parameters for full training on the whole dataset $\mathcal{X}$
- Classical method to determine C and σ of a SVM with Gaussian kernel
  - Applicable for all pairs of parameters for which joint effect is important in the training of classifiers

6.8 Gradient descent for SVM

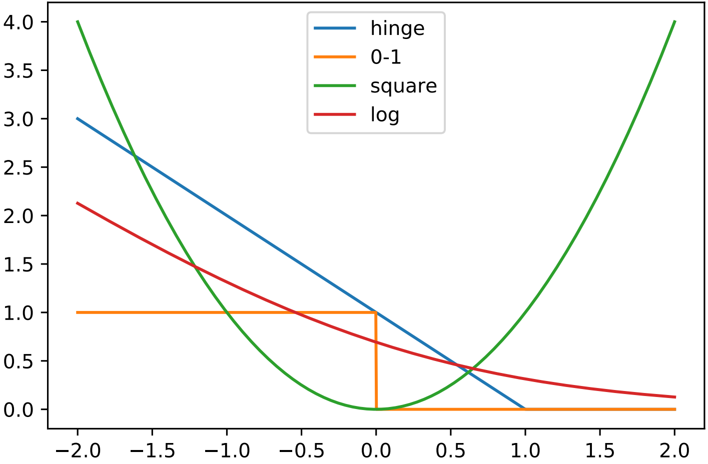
Error function

- SVM: linear discriminant with Hinge loss function

$$\mathcal{L}_{hinge}(y^t, r^t) = \max(1 - y^t r^t, 0)$$

    - $y^t = \mathrm{h}(\mathbf{x}^t | \mathbf{w}, w_0)$
    - Penalizes data on the right side of the hyperplane, but in the margin $(y^t r^t < 1)$
- Each error criterion makes a different tradeoff depending on the nature of the errors
    - 0/1 loss function
    - Quadratic error
    - Cross entropy

Comparison of different error criteria



Gradient descent with kernel

- Discriminant in the space generated by a kernel

$$\mathrm{h}(\mathbf{x}) = \sum_{\mathbf{x}^s \in \mathcal{X}} \alpha^s r^s K(\mathbf{x}^s,\mathbf{x}) + w_0$$

- Learning of the parameters α^t and w_0 can be done using a gradient descent
  - Corrections to be applied to the parameters

$$\Delta\alpha^t = -\eta \frac{\partial E(\boldsymbol{\alpha},w_0|\mathcal{X})}{\partial \alpha^t}, \quad \Delta w_0 = -\eta \frac{\partial E(\boldsymbol{\alpha},w_0|\mathcal{X})}{\partial w_0}$$

  - Updated value, with constraint $\alpha^t \geq 0$, $\forall \alpha^t$:

$$\begin{aligned} \alpha^t &= \begin{cases} 0 & \text{if } \alpha^t + \Delta\alpha^t < 0 \\ \alpha^t + \Delta\alpha^t & \text{otherwise} \end{cases}, \\ w_0 &= w_0 + \Delta w_0. \end{aligned}$$

Error function for gradient descent

- Hinge loss function with regularization for discriminant with kernel

$$
\begin{aligned}
E_{hinge}(\boldsymbol{\alpha}, w_0 | \mathcal{X}) &= \sum_{\mathbf{x}^t \in \mathcal{Y}} (1 - r^t \mathrm{h}(\mathbf{x}^t | \boldsymbol{\alpha}, w_0)) + \lambda \frac{1}{2} \sum_{\alpha^s \in \boldsymbol{\alpha}} (\alpha^s)^2, \\
\mathcal{Y} &= \{\mathbf{x}^t \in \mathcal{X} \mid r^t \mathrm{h}(\mathbf{x}^t | \boldsymbol{\alpha}, w_0) < 1\}.
\end{aligned}
$$

- Maximizes geometric margins in kernel space
  - Value $r^t \mathrm{h}(\mathbf{x}^t \mid \boldsymbol{\alpha}, w_0) \in [0,1]$: data classified properly, but in the margin
- Regularization is necessary
  - Otherwise, α^t values explode!
  - Regularization parameter λ must be adjusted empirically for each dataset (grid search with the σ for Gaussian kernel)

6.9 Kernel functions and distances

Kernel functions and distances

- Kernel function: similarity measurement
- Distance measurement: dissimilarity measurement
- Euclidean distance in space generated by kernel (space $\phi(\mathbf{x})$)

$$d(\mathbf{x},\mathbf{y})^2 = K(\mathbf{x},\mathbf{x}) + K(\mathbf{y},\mathbf{y}) - 2K(\mathbf{x},\mathbf{y})$$

- Example with scalar product type kernel, $K(\mathbf{x},\mathbf{y}) = \mathbf{x}^\top \mathbf{y}$

$$\begin{aligned} d(\mathbf{x},\mathbf{y})^2 &= \|\mathbf{x}-\mathbf{y}\|^2 = (\mathbf{x}-\mathbf{y})^\top(\mathbf{x}-\mathbf{y}) \\ &= \mathbf{x}^\top\mathbf{x} + \mathbf{y}^\top\mathbf{y} - 2\mathbf{x}^\top\mathbf{y} \\ &= K(\mathbf{x},\mathbf{x}) + K(\mathbf{y},\mathbf{y}) - 2K(\mathbf{x},\mathbf{y}) \end{aligned}$$

- Allows to use k-nearest neighbours classifications with kernel functions!
    - Support vectors = prototype selection

Gram matrix

- Gram matrix $G(\mathcal{X})$: measure of similarities between all the data of $\mathcal{X} = \{\mathbf{x}^t\}_{t=1}^N$

$$G(\mathcal{X}) = \begin{bmatrix} K(\mathbf{x}^1,\mathbf{x}^1) & K(\mathbf{x}^1,\mathbf{x}^2) & \cdots & K(\mathbf{x}^1,\mathbf{x}^N) \\ K(\mathbf{x}^2,\mathbf{x}^1) & K(\mathbf{x}^2,\mathbf{x}^2) & \cdots & K(\mathbf{x}^2,\mathbf{x}^N) \\ \dots & \dots & \ddots & \dots \\ K(\mathbf{x}^N,\mathbf{x}^1) & K(\mathbf{x}^N,\mathbf{x}^2) & \cdots & K(\mathbf{x}^N,\mathbf{x}^N) \end{bmatrix}$$

- Symmetrical matrix
- Shape similar to a distance matrix or a covariance matrix

6.10 SVM in scikit-learn

Scikit-learn

- `svm.SVC`: SVM with kernel as seen during this course
  - Some standard kernels supported (linear, Gaussian, polynomial, sigmoid), Gram matrix can also be provided
  - Not so scalable, does not work well with $N > 100\,000$
- `svm.NuSVC`: SVM kernel variant
  - Regularization directly controlling the number of support vectors
- `svm.LinearSVC`: linear SVM
  - Optimized for linear SVM, better resource utilization and scalability
- `linear_model.SGDClassifier`: stochastic gradient descent
  - Can emulate linear SVM with good loss function configuration and regularization
  - Efficient in the use of resources, allows online processing