Machine Learning
Part 2: The Breadth of ML ideas

Marc Toussaint

Machine Learning & Robotics Lab, U Stuttgart

03/06/2013

Table of contents

The Breadth of ML ideas

- A. Ideas about features & data preprocessing
  – centering & whitening
  – PCA
  – PLS (for classification?)
- B. Ideas about local learners
  – local & lazy learning
  – kNN
  – kd-trees
- C. Ideas about combining weak or randomized learners
  – Bootstrap, bagging, and model averaging
  – Boosting
  – (Boosted) decision trees & stumps
  – Random forests
- D. Ideas about other loss functions
  – hinge loss, linear programming, SVMs
- E. Ideas about deep learners

D. Ideas about other loss functions

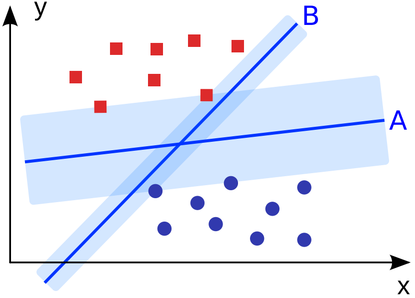
– hinge loss, linear programming, SVMs

Support Vector Machine

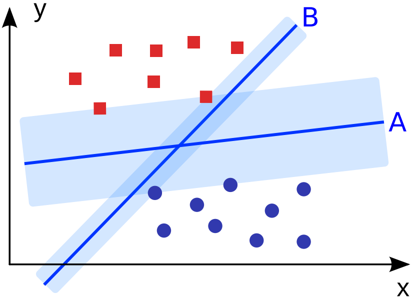
(see Hastie 12.3.2)

- binary linear classifier: $y \in \{-1, +1\}$
- SVM builds a hyperplane to separate two classes.
  A hyperplane is defined by means of β as $\{x \| f(x) = \phi(x)^\top \beta + \beta_0 = 0\}$. The separating hyperplane is linear in the feature space $\phi(x)$, but non-linear in the input space x.
- classification: $x \mapsto \text{sign}(\phi(x)^\top \beta + \beta_0)$
  (linear discriminative function like ridge regression)
- (Warning: offset β_0 requires special attention in kernel methods, but we ignore this issue in the following.)

Support Vector Machine

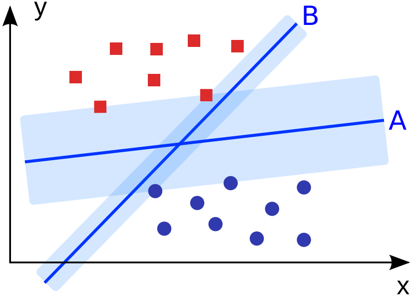


Support Vector Machine



- why maximize the margin? (fat margin vs. thin margin)

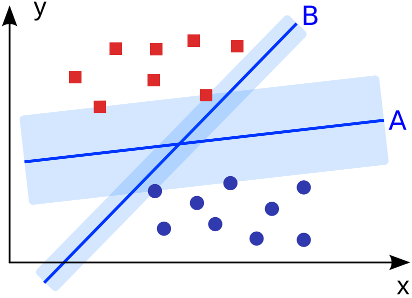
Support Vector Machine



- why maximize the margin? (fat margin vs. thin margin)
- compute the margin? (x_k is the nearest point to the plane)

$$M = \frac{y_k(\phi(x_k)^\top \beta + \beta_0)}{\|\beta\|}$$

Support Vector Machine



- why maximize the margin? (fat margin vs. thin margin)
- compute the margin? (x_k is the nearest point to the plane)

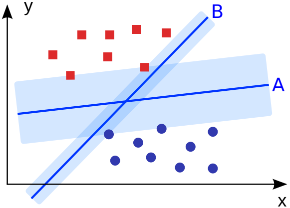
$$M = \frac{y_k(\phi(x_k)^\top \beta + \beta_0)}{\|\beta\|}$$

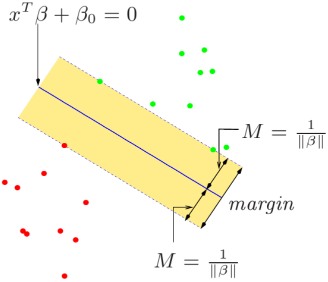
- maximize the margin?

$$\max_{\beta,\beta_0} \min_{x_k} \frac{y_k(\phi(x_k)^\top \beta + \beta_0)}{\|\beta\|}$$

Support Vector Machine (Case 1: linearly separable)

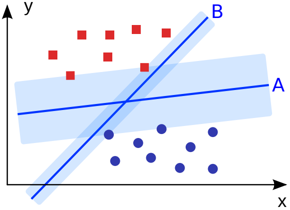
- normalize β: $|\phi(x_k)^\top \beta + \beta_0| = 1$

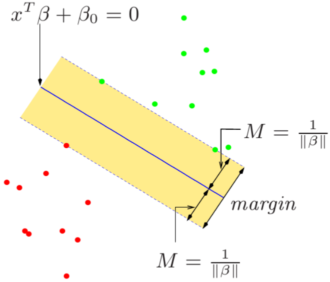




Support Vector Machine (Case 1: linearly separable)

- normalize β: $|\phi(x_k)^\top \beta + \beta_0| = 1$





- can be rephrased as

$$\min_\beta \|\beta\| \qquad \text{subject to } y_i(\phi(x_i)^\top \beta + \beta_0) \geq 1, \quad i = 1, \dots, n$$

Ridge regularization like ridge regression, but different loss

Support Vector Machine (Case 1: linearly separable)

SVM as a Penalization Method

- **Difference** to ridge regression: Hinge loss

Support Vector Machine (Case 1: linearly separable)

SVM as a Penalization Method

- **Difference** to ridge regression: Hinge loss
- SVM uses the *hinge loss* instead of neg-log-likelihood:

$$L^{\text{hinge}(\beta)} = \sum_{i=1}^{n} [1 - y_i \phi(x_i)^\top \beta - \beta_0]_+ + \lambda \|\beta\|^2$$

subscript $+$ indicates the positive part

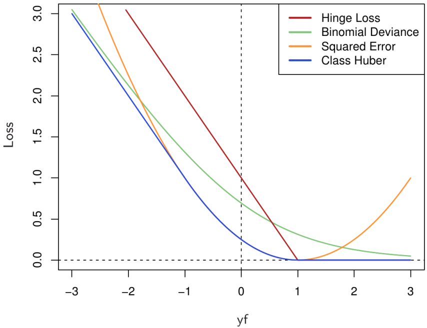
Support Vector Machine (Case 1: linearly separable)

SVM as a Penalization Method

- **Difference** to ridge regression: Hinge loss
- SVM uses the *hinge loss* instead of neg-log-likelihood:

$$L^{\text{hinge}(\beta)} = \sum_{i=1}^{n}[1 - y_i\phi(x_i)^\top\beta - \beta_0]_+ + \color{red}{\lambda\|\beta\|^2}$$

  subscript $+$ indicates the positive part
- Hinge loss is zero if $f(x_i) = \phi(x_i)^\top\beta + \beta_0 \geq 1$ if $y_i = 1$ and $f(x_i) = \phi(x_i)^\top\beta + \beta_0 \leq -1$ if $y_i = -1$.



Support Vector Machine (Case 1: linearly separable)

The Lagrange (primal) function, to be minimized w.r.t. β and β_0 is

$$\boxed{L(\beta, \beta_0) = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\phi(x_i)^\top \beta + \beta_0) - 1]}$$

where $\alpha_i \geq 0$

Support Vector Machine (Case 1: linearly separable)

The Lagrange (primal) function, to be minimized w.r.t. β and β_0 is

$$\boxed{L(\beta, \beta_0) = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\phi(x_i)^\top \beta + \beta_0) - 1]}$$

where $\alpha_i \geq 0$
Setting the derivatives to zero, we obtain:

$$\beta = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i)$$

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

Support Vector Machine (Case 1: linearly separable)

Substituing β and β_0, obtain a simpler convex optimization problem (dual)

$$\boxed{\begin{aligned} L(\alpha) = \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k \phi(x_i)^\top \phi(x_k) \\ \text{such that } \alpha_i \geq 0, \text{ and } 0 = \sum_{i=1}^{N} \alpha_i y_i \end{aligned}}$$

Support Vector Machine (Case 1: linearly separable)

- Using the Lagrangian *dual form*, the solution for $f(x)$ can be written as

$$\begin{aligned} f(x) &= \phi(x)^\top \beta + \beta_0 \\ &= \sum_{i=1}^{n} \alpha_i y_i \phi(x)^\top \phi(x_i) \\ &= \sum_{i=1}^{n} \alpha_i y_i k(x, x_i) + \beta_0 \end{aligned}$$

  one α_i for each training point $(x_i, y_i) \rightarrow \alpha_i, x_i, y_i$ define β implicitly

Support Vector Machine (Case 1: linearly separable)

- Using the Lagrangian *dual form*, the solution for $f(x)$ can be written as

$$\begin{aligned} f(x) &= \phi(x)^\top \beta + \beta_0 \\ &= \sum_{i=1}^{n} \alpha_i y_i \phi(x)^\top \phi(x_i) \\ &= \sum_{i=1}^{n} \alpha_i y_i k(x, x_i) + \beta_0 \end{aligned}$$

  one α_i for each training point $(x_i, y_i) \to \alpha_i, x_i, y_i$ define β implicitly
- The Hinge loss introduces *sparsity*: Most α_i will be zero. Non-zero α_i only for those training points i for which the constraint $y_i(\phi(x_i)^\top \beta + \beta_0) \geq 1$ is exactly met: $\alpha_i \neq 0 \to y_i(\phi(x_i)^\top \beta + \beta_0) = 1$

Support Vector Machine (Case 1: linearly separable)

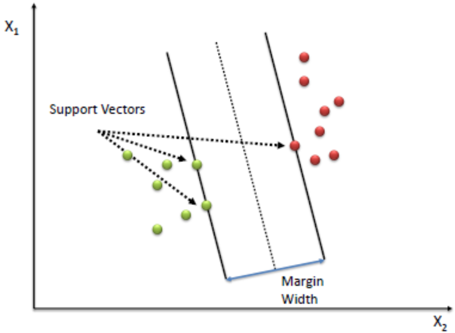
- Using the Lagrangian *dual form*, the solution for $f(x)$ can be written as

$$\begin{aligned} f(x) &= \phi(x)^\top \beta + \beta_0 \\ &= \sum_{i=1}^{n} \alpha_i y_i \phi(x)^\top \phi(x_i) \\ &= \sum_{i=1}^{n} \alpha_i y_i k(x, x_i) + \beta_0 \end{aligned}$$

  one α_i for each training point $(x_i, y_i) \rightarrow \alpha_i, x_i, y_i$ define β implicitly
- The Hinge loss introduces *sparsity*: Most α_i will be zero. Non-zero α_i only for those training points i for which the constraint $y_i(\phi(x_i)^\top \beta + \beta_0) \geq 1$ is exactly met: $\alpha_i \neq 0 \rightarrow y_i(\phi(x_i)^\top \beta + \beta_0) = 1$
- Efficient optimization techniques (quadratic programming with column generation) exploit this.

Support Vector Machine (Case 1: linearly separable)

- Non-zero α_i define the hyperplane / the decision function f. The corresponding x_i are called *support vectors*.



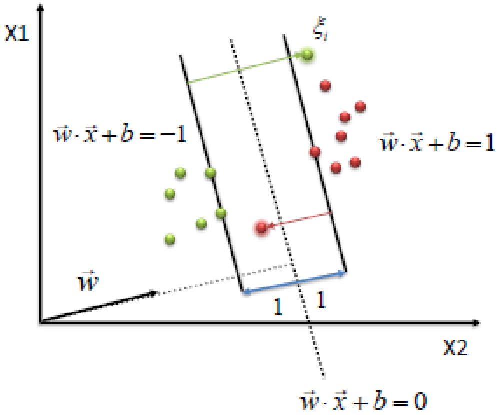
- SVM exploits the *kernel trick* to achieve non-linear classification by reasoning implicitly in non-linear feature space (as in ridge regression).

Support Vector Machine (Case 2: non-linearly separable)

Allow classes to overlap in feature space using *slack variables*

- Hinge loss

$$L^{\text{hinge}(\beta)} = \sum_{i=1}^{n}[1 - y_i\phi(x_i)^\top\beta - \beta_0 - \xi_i]_+ + \sum_{i=1}^{n}\xi_i + \lambda\|\beta\|^2$$



Support Vector Machine (Case 2: non-linearly separable)

Optimization problem:

$$\boxed{\begin{aligned} &\min_{\beta} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\qquad\qquad \text{subject to: } y_i(\phi(x_i)^\top\beta + \beta_0) \geq 1 - \xi_i, \quad i = 1, \ldots, n, \\ &\qquad\qquad\qquad\qquad \text{and } \xi_i \geq 0 \end{aligned}}$$

Support Vector Machine (Case 2: non-linearly separable)

The Lagrange (primal) function is

$$\boxed{L = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\phi(x_i)^\top\beta + \beta_0) - (1 - \xi_i)]}$$

Support Vector Machine (Case 2: non-linearly separable)

The Lagrange (primal) function is

$$\boxed{L = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\phi(x_i)^\top\beta + \beta_0) - (1-\xi_i)]}$$

Derivatives with respect to β, β_0, ξ_i

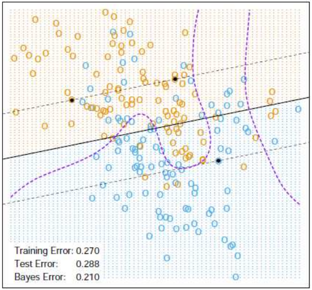
$$\beta = \sum_{i=1}^{N}\alpha_i y_i \phi(x_i)$$

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

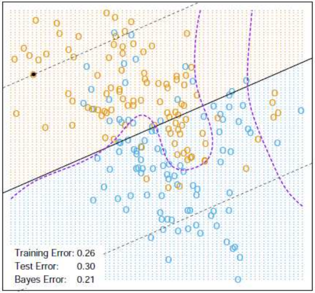
$$\alpha_i = C - \mu_i, \forall i$$

Support Vector Machine (Case 2: non-linearly separable)

In the left panel 62% of the observations are support points, while in the right panel 85% are.



$C = 10000$



$C = 0.01$

(from Hastie 12.2)