STATG019 – Selected Topics in Statistics 2015

Lecture 2

The Support Vector Machine

Dr Franz J. Király

Course organization

In-Course-Assessment

Two take-home ICA, one on kernels, one on point processes

Each counts 50% towards your final grade

Handing out: no.1 on Feb 9, no.2 on Mar 23 (on moodle)
Submission: no.1 on Mar 4, no.2 on Apr 29 (via moodle/TurnitIn)

Further details will be announced via the moodle news forum

Tutorials and/or practical sessions

Thursday, 11am - 1pm location varies

Given topic survey: next tutorial will be mainly on kernlab with R

Please install R and kernlab on your laptops (cluster rooms are full)

We do not have a room yet for tomorrow – do you know of any place?

Did you try the exercises? What do you think about them?

A short overview of the kernel SVM

the Support Vector Machine

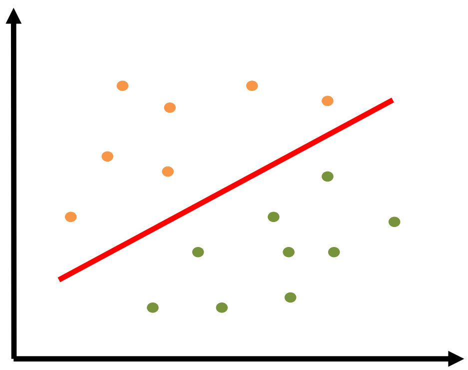
Input: $x_1, \ldots, x_N \in \mathbb{R}^n, y_1, \ldots, y_N \in \{-1, +1\}$

Output: separator/decision function

$$f(x) = \text{sgn}\left(b + w^\top x\right)$$

$w = \sum_{i=1}^N \alpha_i y_i x_i$ solves a QP involving $X^\top X$

$\alpha_i, 1 \leq i \leq N$ are "dual" variables

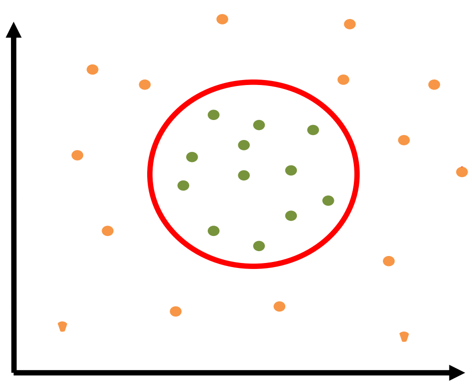


"Kernelized", non-linear variant:

$$f(x) = \text{sgn}\left(b + \sum_{i=1}^N \alpha_i k(x_i, x)\right) = \alpha^\top \kappa(x)$$

α solves a QP involving kernel matrix and vector

$$K = (k(x_i, x_j))_{ij} \qquad \kappa(x) = (k(x_i, x))_i$$



The linear
Support Vector Machine

(V. Vapnik, 1993)

The support vector machine

Goal: linearly separate two classes of points *with maximum margin*

Mathematically:

find a hyperplane $H = \{z \in \mathbb{R}^n \;:\; \langle a, z\rangle = c\}$

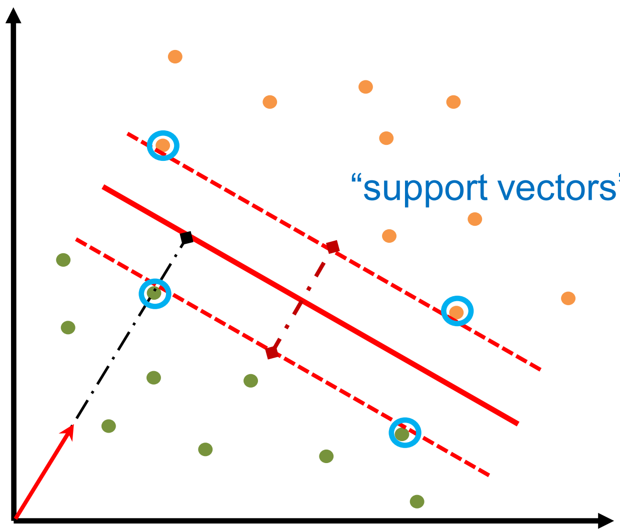
a the normal vector $\qquad c$ the offset

separating the green and the orange points

with the largest symmetric margin of $\pm\eta$

$\langle a, x\rangle \geq c + \eta \qquad$ for all orange points x

$\langle a, x\rangle \leq c - \eta \qquad$ for all green points x



For data $x_1, \ldots, x_N \in \mathbb{R}^n$ and labels $y_1, \ldots, y_N \in \{-1, 1\}$

This can be reformulated as the **optimization problem**

$$\max_{a,c,\eta} \eta \quad \text{s.t. } y_i \cdot (\langle a, x_i\rangle - c) \geq \eta \quad \text{for all } 1 \leq i \leq N$$

Issue: overparamterization!
a, c, η can be multiplied by common factor

Solution: make substitutions $w := \dfrac{a}{\eta}, \quad b := \dfrac{c}{\eta}$ thus $\eta = \dfrac{\|a\|}{\|w\|} = \dfrac{1}{\|w\|}$

to obtain $$\max_{w,b} \frac{1}{\|w\|} \quad \text{s.t. } y_i \cdot (\langle w, x_i\rangle - b) \geq 1 \quad \text{for all } 1 \leq i \leq N$$

Issue: objective function inverse of square root
bad optimization behaviour

Solution: replace $\displaystyle\max_{w,b} \frac{1}{\|w\|}$ by the equivalent $\displaystyle\min_{w,b} \langle w, w\rangle = \|w\|^2$

The support vector machine

Goal: linearly separate two classes of points *with maximum margin*

Quadratic optimization program:

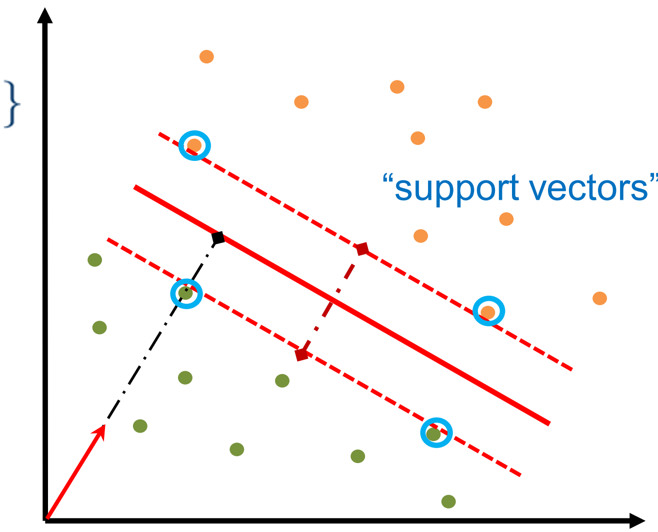
For data $x_1, \ldots, x_N \in \mathbb{R}^n$ and labels $y_1, \ldots, y_N \in \{-1, 1\}$

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

s.t. $y_i \cdot (\langle w, x_i \rangle - b) \geq 1$ for all $1 \leq i \leq N$

margin is $\pm\frac{1}{\|w\|}$

hyperplane is $H = \{z \in \mathbb{R}^n \ : \ \langle w, z \rangle = b\}$



To an unseen data point $x \in \mathbb{R}^n$ assign the label $\operatorname{sgn}(\langle w, x \rangle - b)$

"Naïve" Kernelization (Chapelle, 2006):

Replace x_i by $\phi(x_i)$ and w by $\sum_{i=1}^N \alpha_i \phi(x_i)$ (w is of this form by the representer theorem)

obtain:

$$\min_{\alpha} \frac{1}{2} \alpha^\top \cdot K \cdot \alpha$$

s.t. $y_i \cdot \left(\alpha^\top \kappa(x_i) - b\right) \geq 1$ for all $1 \leq i \leq N$

where $k(y, z) = \langle \phi(y), \phi(z) \rangle$

$K = \left(k(x_i, x_j)\right)_{ij}$ kernel matrix

$\kappa(x) = \left(k(x_i, x)\right)_i$ empirical kernel vector

This is a convex optimization problem in α

The following is about an alternative ("dual" SVM) which is more popular for historical reasons.

Lagrange duality

Lagrange duality

Start with an optimization program *"primal program"*

$$\min_{x \in \mathbb{R}^n} \; f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the loss function to be minimized

$$\text{s.t.} \quad g(x) \le 0$$

(component-wise)

where $g : \mathbb{R}^n \to \mathbb{R}^N$ are N boundary conditions

$x^\star$ optimal solution (assumed to exist)

Definition: the *Lagrangian* for the above problem is

$$L(x, \alpha) := f(x) + \alpha^\top g(x) \qquad L : \mathbb{R}^n \times \mathbb{R}^N \to \mathbb{R}$$

the *Lagrange dual function* is

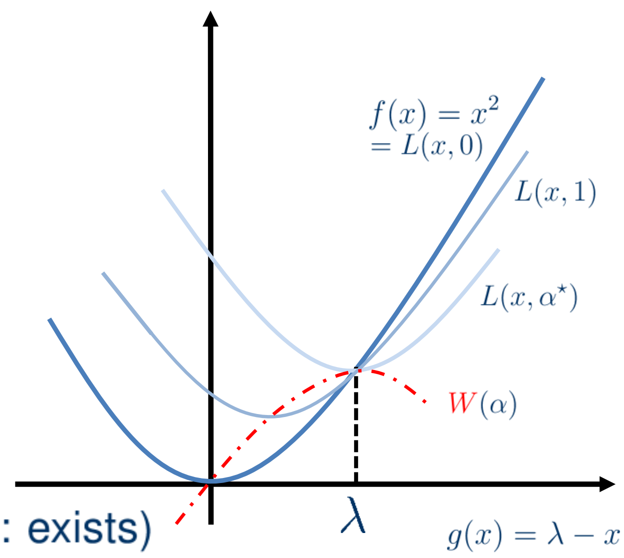
$$W(\alpha) := \inf_{x \in \mathbb{R}^n} L(x, \alpha) \qquad W : \mathbb{R}^N \to \mathbb{R}$$

the (*Lagrange*) *dual program* is

$$\max_{\alpha \in \mathbb{R}^N} W(\alpha)$$

$$\text{s.t. } \alpha \ge 0$$

$\alpha^\star$ optimal solution (one can show: exists)



Proposition: it holds that $W(\alpha^\star) \le f(x^\star)$ "duality gap"

Theorem (Slater): If the primal problem is *convex*, i.e., if f and g are, and there is $x \in \mathbb{R}^N$ such that $g(x) < 0$, then

$$W(\alpha^\star) = L(x^\star, \alpha^\star) = f(x^\star)$$

"strong duality holds"

Lagrange duality

primal program

$$\min_{x\in\mathbb{R}^n} f(x)$$

s.t. $g(x) \leq 0$

$$L(x,\alpha) := f(x) + \alpha^\top g(x)$$

$$W(\alpha) := \inf_{x\in\mathbb{R}^n} L(x,\alpha)$$

dual program

$$\max_{\alpha\in\mathbb{R}^N} W(\alpha)$$

s.t. $\alpha \geq 0$

Theorem (Slater): If the primal problem is *convex*, i.e., if f and g are, and there is $x \in \mathbb{R}^N$ such that $g(x) < 0$, then

$$W(\alpha^\star) = L(x^\star, \alpha^\star) = f(x^\star)$$

"strong duality holds"

Observation: $L(x,\alpha)$ is convex for fixed α

so hopefully $x^\star$ can be efficiently obtained as the minimum of $L(x,\alpha^\star)$ in x

Example 1: linearly constrained linear program

primal program

$$\min_{x\in\mathbb{R}^n} \langle c, x\rangle$$

s.t. $A \cdot x \leq b$

$c \in \mathbb{R}^n, A \in \mathbb{R}^{N\times n}$

$$L(x,\alpha) = c^\top x + \alpha^\top \cdot (Ax - b)$$
$$= (c^\top + \alpha^\top A)x - \alpha^\top b$$

$$W(\alpha) = \begin{cases} -\alpha^\top b, & c = A^\top \alpha \\ -\infty & \text{otherwise} \end{cases}$$

dual program

$$\min_{\alpha\in\mathbb{R}^N} \langle b, \alpha\rangle$$

s.t. $\alpha \geq 0$

$A^\top \alpha = c$

dual program is in standard form

Lagrange duality

primal program

$$\min_{x \in \mathbb{R}^n} f(x)$$

s.t. $g(x) \leq 0$

$$L(x, \alpha) := f(x) + \alpha^\top g(x)$$

$$W(\alpha) := \inf_{x \in \mathbb{R}^n} L(x, \alpha)$$

dual program

$$\max_{\alpha \in \mathbb{R}^N} W(\alpha)$$

s.t. $\alpha \geq 0$

Example 2: the support vector machine

primal program

$$\min_{(w,b) \in \mathbb{R}^{n+1}} \frac{1}{2}\langle w, w \rangle \quad \text{s.t.} \quad y_i \cdot (\langle w, x_i \rangle - b) \geq 1 \quad \text{for all } 1 \leq i \leq N$$

note: this is a convex program

$$L((w, b), \alpha) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{N} [\alpha_i - \alpha_i y_i (\langle w, x_i \rangle - b)]$$

due to convexity, minimizer is zero of first derivatives:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{N} \alpha_i y_i x_i \qquad \frac{\partial L}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i$$

implies $w^\star = \sum_{i=1}^{N} \alpha_i^\star y_i x_i$

(compare representer theorem!)

therefore $W(\alpha) = L\left(\sum_{i=1}^{N} \alpha_i y_i x_i, b, \alpha\right) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \cdot x_i^\top x_j$ (if $\frac{\partial L}{\partial b} = 0$, othw. $-\infty$)

dual program $\max_{\alpha \in \mathbb{R}^N} W(\alpha)$ s.t. $\alpha \geq 0, \quad 0 = \sum_{i=1}^{N} \alpha_i y_i$

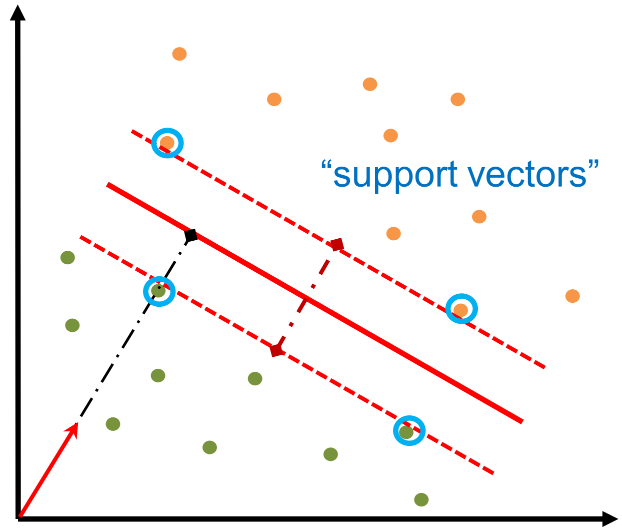
Kernelizable!

The kernel
Support Vector Machine

The kernel support vector machine

The linear support vector machine

For data $x_1, \ldots, x_N \in \mathbb{R}^n$ and labels $y_1, \ldots, y_N \in \{-1, 1\}$



Primal formulation:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y_i \cdot (\langle w, x_i \rangle - b) \geq 1 \quad \text{for all } 1 \leq i \leq N$$

Dual formulation:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \cdot x_i^\top x_j$$

$$\text{s.t.} \quad \alpha \geq 0, \quad 0 = \sum_{i=1}^{N} \alpha_i y_i$$

Obtain $w = \sum_{i=1}^{N} \alpha_i y_i x_i, \quad b = \frac{1}{N} \sum_{i=1}^{N} (w^\top x_i - y_i)$

Predicted label is $y(x) := \text{sgn}\ (\langle w, x \rangle - b)$

The Kernel SVM

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \cdot k(x_i, x_j) \quad \text{s.t.} \quad \alpha \geq 0, \quad 0 = \sum_{i=1}^{N} \alpha_i y_i$$

$$= \|\alpha\|_1 - \frac{1}{2} \cdot \alpha^\top \tilde{K} \alpha$$

"sparse norm"

where $\tilde{K} = (y_i y_j k(x_i, x_j))_{ij}$ (this is a psd kernel matrix)

Obtain $b = \frac{1}{N} \sum_{i=1}^{N} k(w, x_i) - y_i$

Predict $\text{sgn}\ (\alpha^\top \tilde{\kappa}(x) - b)$ where $\tilde{\kappa}(x) = (y_i k(x_i, x))_i$

Intuition on the non-linear kernel SVM

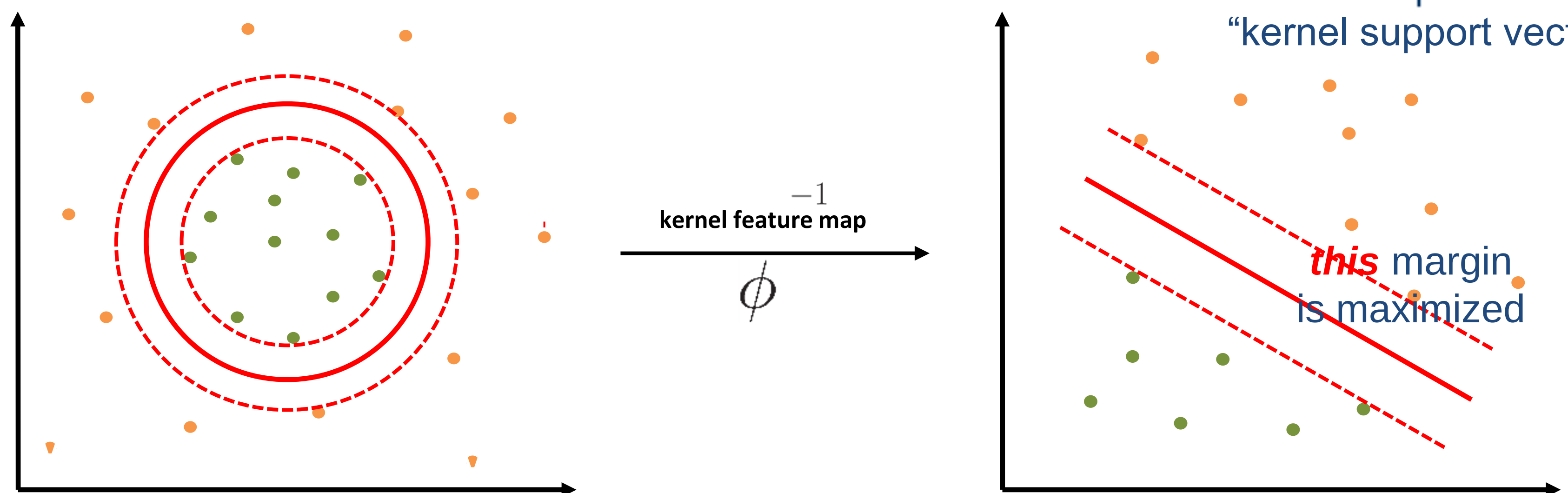
For data $x_1, \ldots, x_N \in \mathbb{R}^n$ and labels $y_1, \ldots, y_N \in \{-1, 1\}$ $\qquad \tilde{K} = (y_i y_j k(x_i, x_j))_{ij}$

$$\max_{\alpha} W(\alpha) = \|\alpha\|_1 - \frac{1}{2} \cdot \alpha^\top \tilde{K} \alpha \quad \text{s.t.} \quad \alpha \geq 0, \quad 0 = \sum_{i=1}^{N} \alpha_i y_i \qquad \tilde{\kappa}(x) = (y_i k(x_i, x))_i$$

Obtain $\quad b = \frac{1}{N} \sum_{i=1}^{N} k(w, x_i) - y_i$ $\qquad$ Predict $\quad y(x) :=$ sgn $\left(\alpha^\top \tilde{\kappa}(x) - b\right)$

Observation: separator is manifold of form $\mathcal{M} = \{z \ : \ \alpha^\top \tilde{\kappa}(z) = b\}$

with α sparse
“kernel support vectors”



For polynomial kernel: $\alpha^\top \tilde{\kappa}(z)$ is a polynomial $\sum_{i=1}^{N} \pm \alpha_i \left(\langle z, x_i \rangle\right)^d$ “sparse Waring form”

For Gaussian kernel: $\alpha^\top \tilde{\kappa}(z) = \sum_{i=1}^{N} \pm \alpha_i \exp\left(\frac{\|z - x_i\| 2}{2\sigma^2}\right)$ due to feature space infinite:
α often not very sparse
(though many entries will be small)

A dictionary of kernel Support Vector Classifiers

name	how it differs from the vanilla-SVM	pros/cons	type in R kernlab-`ksvm`
(hard-margin) SVC	The original formulation (previous slide)	Very outlier sensitive, solution may not exist. **Don't use.**	`C-svc` set `C`=**0**
soft-margin SVC or C-SVC (Cortes, Vapnik)	$\min_{w,b,\xi} \frac{1}{2}\lVert w\rVert^2 + C\sum_{i=1}^N \xi_i$ s.t. $y_i \cdot (\langle w, x_i\rangle - b) \geq 1 - \xi_i$, $\xi_i \geq 0$ data points can lie outside the margin but violation is punished infinitesimally by C	**Working horse** of support vector classification. P: easy, sparse solution C: somewhat outlier sensitive	`C-svc`
nu-SVC (Schölkopf et al)	$\min_{w,\xi,\rho,b} \frac{1}{2}\lVert w\rVert^2 - \nu\rho + \frac{1}{2}\sum_{i=1}^N \xi_i$ parameter $\nu \in (0,1)$ s.t. $y_i \cdot (\langle w, x_i\rangle - b) \geq \rho - \xi_i$, $\xi_i \geq 0$, $\rho \geq 0$ data points can lie outside the margin fraction of violations of margin is hard-bounded above by ν	support vector classification which emphasizes *number* of outliers above *severity* P: good for severe outliers C: tends to overfit more	`nu-svc`
bound-constraint SVC (Mangasarian et al)	$\min_{w,b,\xi} \frac{1}{2}\lVert w\rVert^2 + \frac{1}{2}b^2 + C\sum_{i=1}^N \xi_i$ otherwise same as C-SVC equivalent to homogenous & no intercept	minor variant of C-SVC P: can be worth trying C: sometimes it isn't	`C-bsvc`
multi-class SVC	Weston/Watkins: sum of C-SVC $(\langle w, x_i\rangle - b)_{c(i)} \geq (\langle w, x_i\rangle - b + 2 - \xi_i)_{\neg c(i)}$ Crammer/Singer: $\min_{M,\xi} \frac{\beta}{2}\lVert M\rVert_F^2 + \sum_{i=1}^N \xi_i$	better than naïve one-vs-one or one-vs-all multiclassifiers W/W-risk is closer to C-SVC C/S-risk is closer to nu-SVC	`kbb-svc` `spoc-svc`

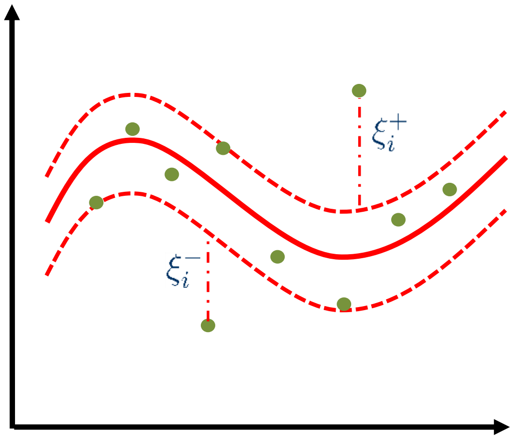
Regression and Outlier Detection with the SVM

Kernel support vector regression

Classification: For data $x_1, \ldots, x_N \in \mathbb{R}^n$ and labels $y_1, \ldots, y_N \in \{-1, 1\}$
learn a classifier f such that $f(x_i) \approx y_i$

Regression: For data $x_1, \ldots, x_N \in \mathbb{R}^n$ and labels $y_1, \ldots, y_N \in \mathbb{R}$
learn a regressor f such that $f(x_i) \approx y_i$

Idea: instead of mis-classification, penalize $\|f(x_i) - y_i\|$ with a *margin*



C-SVC: $\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^N \xi_i$ s.t. $y_i \cdot (\langle w, x_i\rangle - b) \geq 1 - \xi_i$, $\xi_i \geq 0$

eps-SVR: $\min_{w,b,\xi^+,\xi^-} \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^N \left[\xi_i^+ + \xi_i^-\right]$ parameters $C, \varepsilon \in \mathbb{R}^+$

s.t. $(\langle w, x_i\rangle - b) - y_i \leq \varepsilon + \xi_i^+, \quad \xi_i^+ \geq 0$
$y_i - (\langle w, x_i\rangle - b) \leq \varepsilon + \xi_i^-, \quad \xi_i^- \geq 0$ for all $1 \leq i \leq N$

nu-SVR = eps-SVR + nu-SVC

$$\min_{w,b,\xi^+,\xi^-} \frac{1}{2}\|w\|^2 + C\left(\nu\varepsilon + \frac{1}{N}\sum_{i=1}^N \left[\xi_i^+ + \xi_i^-\right]\right)$$

parameters $C, \varepsilon \in \mathbb{R}^+$, $\nu \in (0, 1)$

s.t. $(\langle w, x_i\rangle - b) - y_i \leq \varepsilon + \xi_i^+, \quad \xi_i^+ \geq 0$
$y_i - (\langle w, x_i\rangle - b) \leq \varepsilon + \xi_i^-, \quad \xi_i^- \geq 0$

(pros/cons analogous to SVCs)

bound-constraint SVR = eps-SVR + b²

kernlab-`ksvm:`

`eps-svr, nu-svr, eps-bsvr`

Novelty detection with the one-class SVM

Classification: For data $x_1, \ldots, x_N \in \mathbb{R}^n$ and labels $y_1, \ldots, y_N \in \{-1, 1\}$
learn a classifier f such that $f(x_i) \approx y_i$

Novelty detection: Most data $x_1, \ldots, x_N \in \mathbb{R}^n$ so far are "normal"
Learn a classifier that is able to identify "abnormal" points

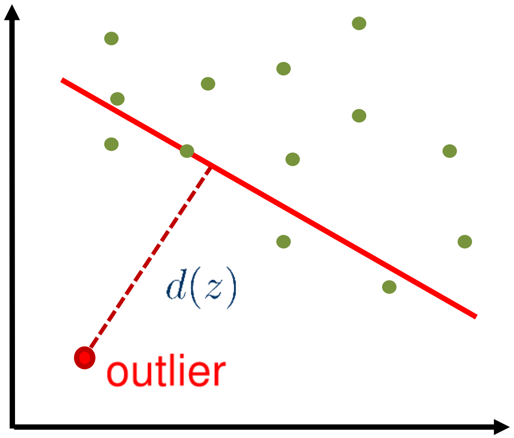
Idea: "normal" points should be separated from origin by boundary (with slack)

C-SVC: $\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^N \xi_i$ s.t. $y_i \cdot (\langle w, x_i\rangle - b) \geq 1 - \xi_i$
$\xi_i \geq 0$

1-SVC: $\min_{w,\xi,\rho} \frac{1}{2}\|w\|^2 - \rho + \frac{1}{\nu \cdot N}\sum_{i=1}^N \xi_i$ parameter $\nu \in (0,1)$

s.t. $\langle w, x_i\rangle \geq \rho - \xi_i, \quad \xi_i \geq 0$ for all $1 \leq i \leq N$

$\nu =$ upper bound on training points past the margin



Outlier score = distance to "normal" side $d(z) = \frac{w^\top x - \rho}{\|w\|}$

1-SVC after **kernelization:**

$$\min_\alpha \frac{1}{2}\alpha^\top K\alpha \quad \text{and} \quad d(z) = \frac{\alpha^\top \kappa(x) - \rho}{\sqrt{\alpha^\top K\alpha}} \quad \text{where} \quad K = (k(x_i, x_j))_{ij}, \quad \kappa(x) = (k(x_i, x))_i$$

s.t. $0 \leq \alpha \leq \frac{1}{\nu N}, \quad \|\alpha\|_1 = 1$

kernlab-`ksvm`: `one-svc`
improved variants exist (but not in R)

Using the SVM in kernlab

Support vector learning in kernlab

`ksvm` encapsulates all methods of support vector learning discussed today

Prior to use, kernlab has to be loaded with `library(kernlab)`

usage for training:

```
svmmodel <- ksvm(vartopredict~.,data=traindata,…)
```

trains a SVM, stored in the output variable `svmmodel` of type `ksvm`

important parameters for `ksvm`:

`type`	determines the kind of learning machine, e.g. `"C-svc"`
`kernel`	determines the kernel used, e.g. `"rbf-dot"`
`kpar`	a list of kernel parameters, e.g. `list(sigma=1)`
`C,nu,epsilon`	regularization parameters for the various methods

output contains model as `alpha,b,alphaindex`:

usage for prediction:

```
predicted <- predict(svmmodel, testdata)
```

yields a vector `predicted` of predictions for `testdata`

Documentation: http://cran.r-project.org/web/packages/kernlab/kernlab.pdf

Type `help(ksvm)` or `example(ksvm)` for more details and examples

Methods for model selection

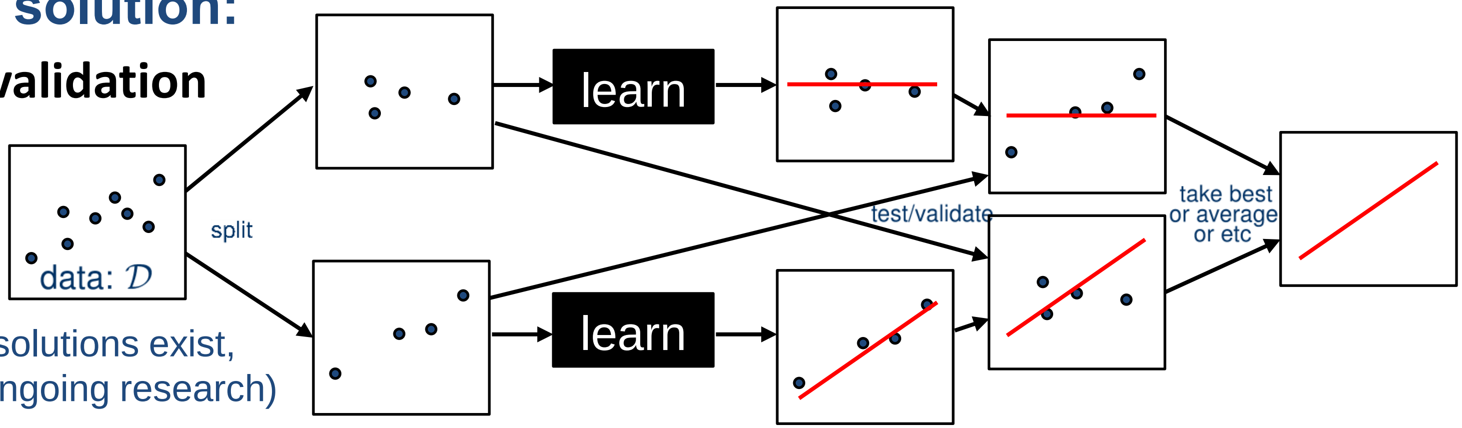
Major issue:

In kernels and in the support vector learning methods,
there are regularisation parameters to choose, such as

σ in Gaussian k. C in C-SVC ν in nu-SVC ε in SVR etc.

"universal" solution:

Cross-validation



(param-specific solutions exist,
also topic of ongoing research)

for `ksvm`:

error estimation by k-fold cross-validation by setting param. `cross = k`
estimated cross-validation error is returned as `cross`

cross-validation for class probabilities with parameter `prob.model = T`
cross-validation for other parameters must be done manually (e.g. `cvTools`)

Outlook

Lecture 1: Introduction to kernels

Main concepts and theoretical results, learning guarantees

Kernel PCA and kernel ridge regression

Some notes on R and kernlab

Lecture 2: the kernel support vector machine

The linear support vector machine, duality

Hard- and soft-margin two-class SVM

The one-class SVM

Support vector regression

Lecture 3: Gaussian processes and kernel learning

Potential further lecture topics:

Algorithms: kernel discriminants, kernel k-means, kernel quantile regression
kernel CCA, kernel MMD, kernel relevance vector machine

Large-scale learning with kernels, subset methods and Nyström-approximation

Combinatorial kernels: string kernels, graph kernels

Invariance kernels

Vapnik-Chervonenkis learning theory

Outlier detection, novelty detection

On-line kernel learning

Next week: Gaussian Processes