On the discardability of data in Support Vector Classification problems

Simone Del Favero, **Damiano Varagnolo**, Francesco Dinuzzo,
Luca Schenato, Gianluigi Pillonetto

Department of Information Engineering – Padova, Italy
Max Planck Institute – Tübingen, Germany

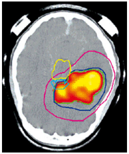
December 13$^{\text{th}}$, 2011 – 50$^{\text{th}}$ IEEE CDC

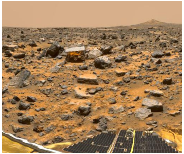
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

MAX-PLANCK-GESELLSCHAFT

Support Vector Classification is . . .

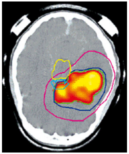
. . . transform numbers into labels . . .

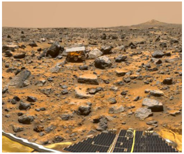




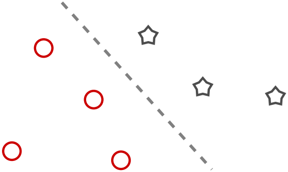
Support Vector Classification is . . .

. . . transform numbers into labels . . .





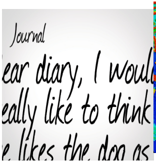
. . . minimizing the structural risk

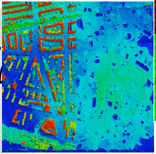


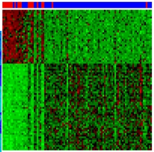
Cortés, Vapnik (1995)
Support-Vector Networks
Machine Learning

Support Vector Classifiers in the real world

several examples
of successful applications!

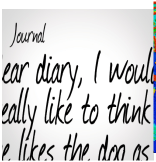


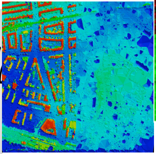
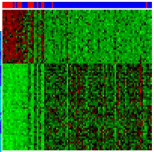




Support Vector Classifiers in the real world

several examples
of successful applications!

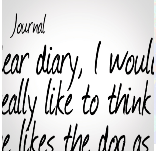


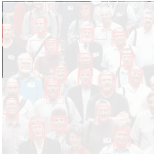




possible
bottlenecks

Support Vector Classifiers in the real world

several examples
of successful applications!





possible bottlenecks → training may be slow

Support Vector Classifiers in the real world

several examples
of successful applications!

possible bottlenecks

- training may be slow
- evaluation of the decision function may be slow

Counter-measures for the bottlenecks (1/2)

Several strategies to **enhance the training phase**:

Counter-measures for the bottlenecks (1/2)

Several strategies to **enhance the training phase**:

- chunking — Vapnik (1982) Estim. of Depend. Based on Emp. Data *Springer-Verlag*

Counter-measures for the bottlenecks (1/2)

Several strategies to **enhance the training phase**:

- chunking — Vapnik (1982) Estim. of Depend. Based on Emp. Data *Springer-Verlag*
- SMO — Platt (1998) SMO: a fast alg. for training SVMs *Adv. in Ker. Meth.*

Counter-measures for the bottlenecks (1/2)

Several strategies to **enhance the training phase**:

- chunking — Vapnik (1982) Estim. of Depend. Based on Emp. Data *Springer-Verlag*
- SMO — Platt (1998) SMO: a fast alg. for training SVMs *Adv. in Ker. Meth.*
- Active sets — Musicant Feinberg (2004) Active set SV regr. *IEEE Trans. on N.N.*

Counter-measures for the bottlenecks (1/2)

Several strategies to **enhance the training phase**:

- chunking — Vapnik (1982) Estim. of Depend. Based on Emp. Data *Springer-Verlag*
- SMO — Platt (1998) SMO: a fast alg. for training SVMs *Adv. in Ker. Meth.*
- Active sets — Musicant Feinberg (2004) Active set SV regr. *IEEE Trans. on N.N.*
- new QPs — Mangasarian Musicant (2001) Lagrangian SVM *J. of Mach. L. Res.*

Counter-measures for the bottlenecks (1/2)

Several strategies to **enhance the training phase**:

- chunking — Vapnik (1982) Estim. of Depend. Based on Emp. Data *Springer-Verlag*
- SMO — Platt (1998) SMO: a fast alg. for training SVMs *Adv. in Ker. Meth.*
- Active sets — Musicant Feinberg (2004) Active set SV regr. *IEEE Trans. on N.N.*
- new QPs — Mangasarian Musicant (2001) Lagrangian SVM *J. of Mach. L. Res.*
- new kernel matrix — Fine Scheinberg (2001) Eff. SVM train. using low rank ker. rep. *J. of Mach. L. Res.*
  Williams Seeger (2001) Using the Nyström meth. to speed up ker. mach. *NIPS*

Counter-measures for the bottlenecks (2/2)

Several strategies to **reduce the dataset** / **compress the evaluation function**:

Counter-measures for the bottlenecks (2/2)

Several strategies to **reduce the dataset** / **compress the evaluation function**:

Before training

- k-NN — Li (2004) Dist. based select. of Pot. SVs by ker. mat. *Adv. in N.N.*
- FDA — Lei Long (2011) Locate Pot. SVs for faster SMO *IEEE Conf. on Nat. Comp.*

Counter-measures for the bottlenecks (2/2)

Several strategies to **reduce the dataset** / **compress the evaluation function**:

Before training

- k-NN — Li (2004) Dist. based select. of Pot. SVs by ker. mat. *Adv. in N.N.*
- FDA — Lei Long (2011) Locate Pot. SVs for faster SMO *IEEE Conf. on Nat. Comp.*

While training

- reduced sets — Burges Schölkopf (1997) Improv. acc. and speed of SV learn. mach. *NIPS*
- huller — Bordes Bottou (2005) The huller: a simple and efficient online SVM *ECML*

Counter-measures for the bottlenecks (2/2)

Several strategies to **reduce the dataset** / **compress the evaluation function**:

After training

- exact reduct.

  Downs et al. (2001) Exact simpl. of SV sol. *J. of M.L. Res.*

- approx. reduct.

  Engel et al. (2002) Sparse online greedy SV Regr. *ECML*

While training

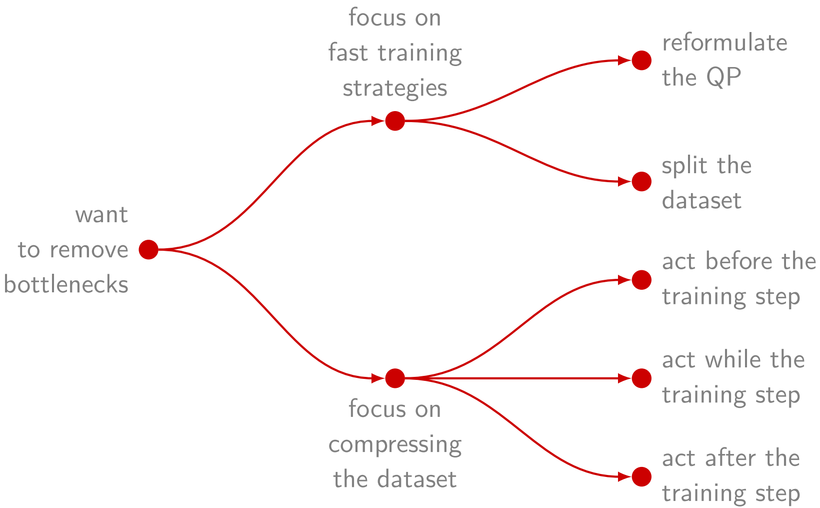
- reduced sets

  Burges Schölkopf (1997) Improv. acc. and speed of SV learn. mach. *NIPS*

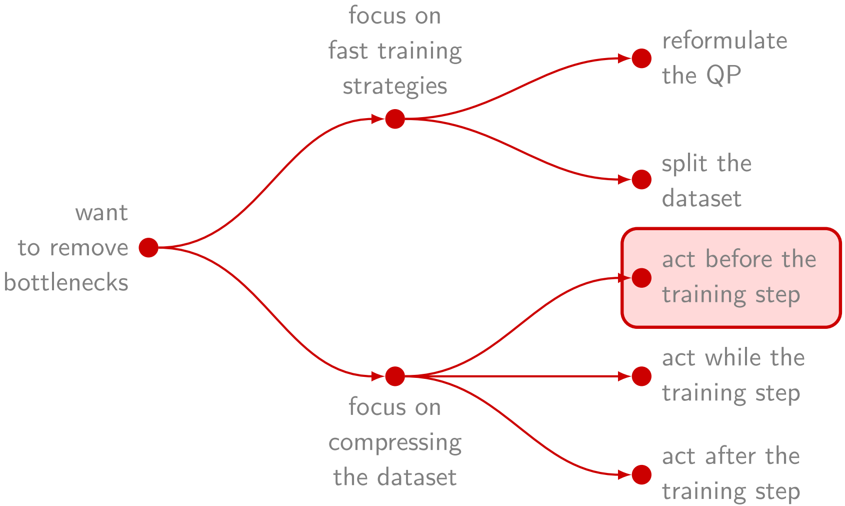
- huller

  Bordes Bottou (2005) The huller: a simple and efficient online SVM *ECML*

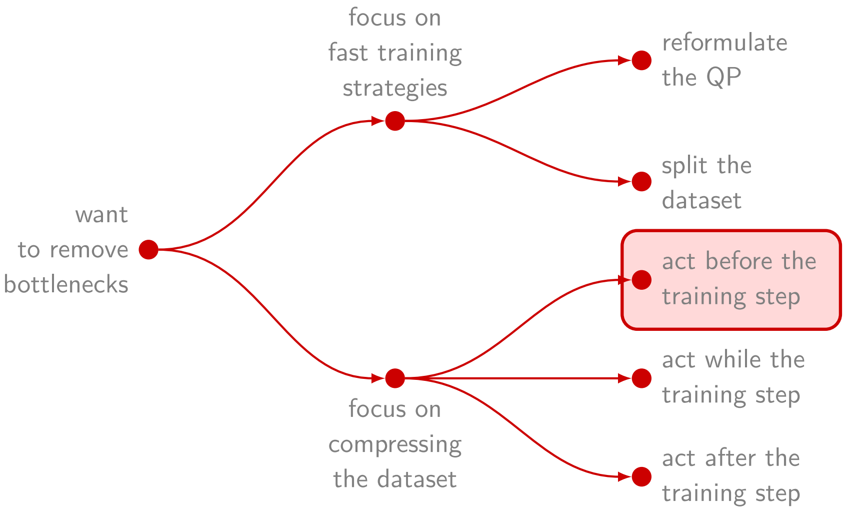
Our contributions w.r.t. the existing literature



Our contributions w.r.t. the existing literature



Our contributions w.r.t. the existing literature



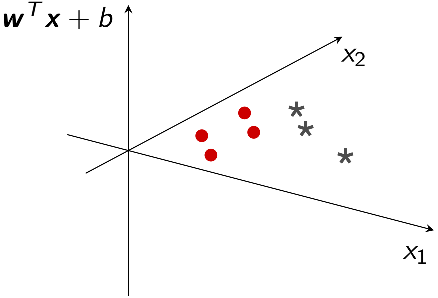
Motivation: distributed learning

Claim

in this talk we do not present the
results on non-separable datasets

Support Vector Classification: a brief overview

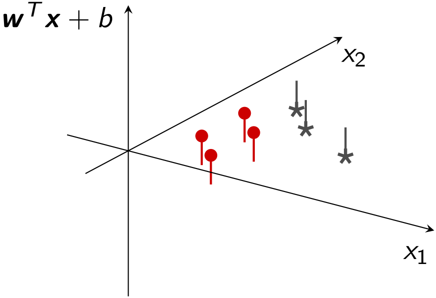
(for separable datasets)



$$\min_{\boldsymbol{w},b} \ \|\boldsymbol{w}\|_2$$
$$\text{s.t. } y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1$$

Support Vector Classification: a brief overview

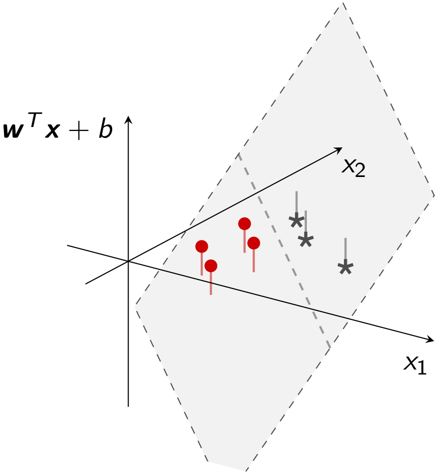
(for separable datasets)



$$\min_{\boldsymbol{w},b} \quad \|\boldsymbol{w}\|_2$$
$$\text{s.t. } y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1$$

Support Vector Classification: a brief overview

(for separable datasets)

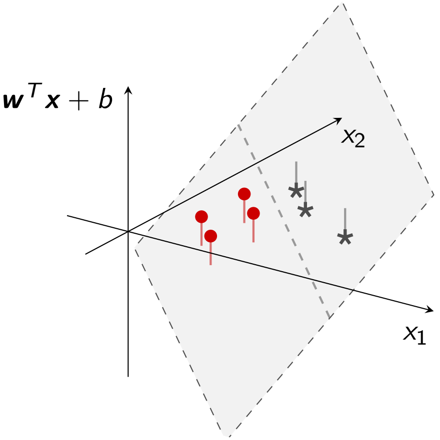


$$\min_{\boldsymbol{w},b} \quad \|\boldsymbol{w}\|_2$$

$$\text{s.t. } y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1$$

Support Vector Classification: a brief overview

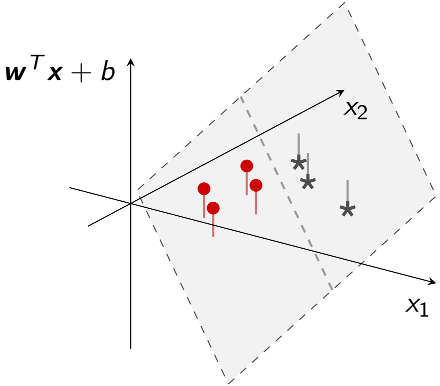
(for separable datasets)



$$\min_{\boldsymbol{w},b} \quad \|\boldsymbol{w}\|_2$$
$$\text{s.t. } y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1$$

Support Vector Classification: a brief overview

(for separable datasets)

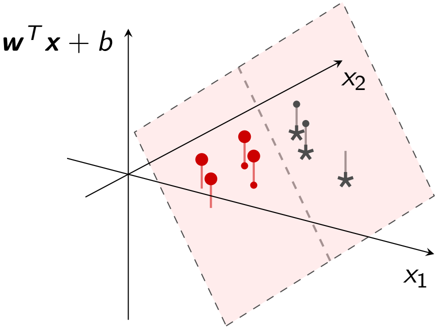


$$\min_{\boldsymbol{w},b} \quad \|\boldsymbol{w}\|_2$$

$$\text{s.t. } y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1$$

Support Vector Classification: a brief overview

(for separable datasets)



$$\min_{\boldsymbol{w},b} \quad \|\boldsymbol{w}\|_2$$

$$\text{s.t. } y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1$$

Potential Support Vectors and Discardable Vectors

Definition: Potential Support Vector

$(\boldsymbol{x}_i, y_i) =$ Potential SV for dataset $\mathcal{D}$

if

exists plausible **future data** s.t. $(\boldsymbol{x}_i, y_i)$ **will become** a SV

Potential Support Vectors and Discardable Vectors

Definition: Potential Support Vector

$(\boldsymbol{x}_i, y_i)$ = Potential SV for dataset $\mathcal{D}$

if

exists plausible **future data** s.t. $(\boldsymbol{x}_i, y_i)$ **will become** a SV

focus: keep information useful for future retrainings!

Potential Support Vectors and Discardable Vectors

Definition: Potential Support Vector

$(\boldsymbol{x}_i, y_i) =$ Potential SV for dataset $\mathcal{D}$

if

exists plausible **future data** s.t. $(\boldsymbol{x}_i, y_i)$ **will become** a SV

focus: keep information useful for future retrainings!

Definition: Discardable Vector

$(\boldsymbol{x}_i, y_i) =$ Discardable Vector for dataset $\mathcal{D}$

if

it is not a Potential SV

Potential Support Vectors and Discardable Vectors

Definition: Potential Support Vector

$(\boldsymbol{x}_i, y_i) =$ Potential SV for dataset $\mathcal{D}$

if

exists plausible **future data** s.t. $(\boldsymbol{x}_i, y_i)$ **will become** a SV

focus: keep information useful for future retrainings!

Definition: Discardable Vector

$(\boldsymbol{x}_i, y_i) =$ Discardable Vector for dataset $\mathcal{D}$

if

it is not a Potential SV

important: $(\boldsymbol{x}_i, y_i)$ is ***either*** Potential ***or*** Discardable

Towards the characterization of the Potential SVs and the Discardable Vectors

Definition: quasi separating hyperplane

$(\boldsymbol{w}, b)$ quasi separates a dataset $\mathcal{D}$ if $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 0$ for all i

Towards the characterization of the Potential SVs and the Discardable Vectors

Definition: quasi separating hyperplane

$(\boldsymbol{w}, b)$ quasi separates a dataset $\mathcal{D}$ if $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 0$ for all i

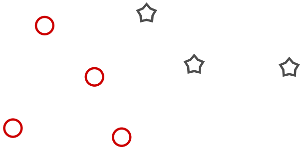
separating hyperplane $\Leftrightarrow y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$

Towards the characterization of the Potential SVs and the Discardable Vectors

Definition: quasi separating hyperplane

$(\boldsymbol{w}, b)$ quasi separates a dataset $\mathcal{D}$ if $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 0$ for all i

separating hyperplane $\Leftrightarrow y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$

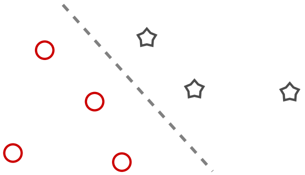


Towards the characterization of the Potential SVs and the Discardable Vectors

Definition: quasi separating hyperplane

$(\boldsymbol{w}, b)$ quasi separates a dataset $\mathcal{D}$ if $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 0$ for all i

separating hyperplane $\Leftrightarrow y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$

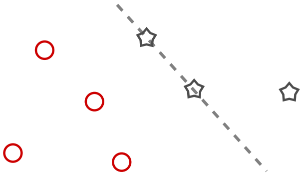


Towards the characterization of the Potential SVs and the Discardable Vectors

Definition: quasi separating hyperplane

$(\boldsymbol{w}, b)$ quasi separates a dataset $\mathcal{D}$ if $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 0$ for all i

separating hyperplane $\Leftrightarrow y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$

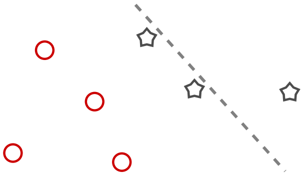


Towards the characterization of the Potential SVs and the Discardable Vectors

Definition: quasi separating hyperplane

$(\boldsymbol{w}, b)$ quasi separates a dataset $\mathcal{D}$ if $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 0$ for all i

separating hyperplane $\Leftrightarrow y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$

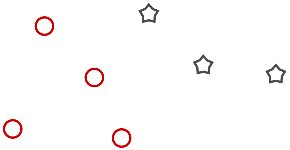


Full characterization of the Potential SVs

Proposition

$(\boldsymbol{x}_i, y_i) =$ Potential SV **if and only if** exists $(\boldsymbol{w}, b) \neq (\mathbf{0}, 0)$ that

1. pass through $(\boldsymbol{x}_i, 0)$
2. quasi separates $\mathcal{D}$
3. can pass through $(\boldsymbol{x}_j, 0)$ if $\boldsymbol{x}_j$ is of the same class of $\boldsymbol{x}_i$
4. cannot pass through $(\boldsymbol{x}_j, 0)$ if $\boldsymbol{x}_j$ is of the opposite class of $\boldsymbol{x}_i$



Full characterization of the Potential SVs

Proposition

$(\boldsymbol{x}_i, y_i)$ = Potential SV **if and only if** exists $(\boldsymbol{w}, b) \neq (\boldsymbol{0}, 0)$ that

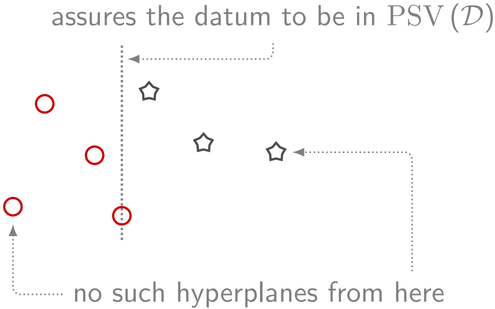
1. pass through $(\boldsymbol{x}_i, 0)$
2. quasi separates $\mathcal{D}$
3. can pass through $(\boldsymbol{x}_j, 0)$ if $\boldsymbol{x}_j$ is of the same class of $\boldsymbol{x}_i$
4. cannot pass through $(\boldsymbol{x}_j, 0)$ if $\boldsymbol{x}_j$ is of the opposite class of $\boldsymbol{x}_i$

Full characterization of the Potential SVs

Proposition

$(\boldsymbol{x}_i, y_i)$ = Potential SV **if and only if** exists $(\boldsymbol{w}, b) \neq (\boldsymbol{0}, 0)$ that

1. pass through $(\boldsymbol{x}_i, 0)$
2. quasi separates $\mathcal{D}$
3. can pass through $(\boldsymbol{x}_j, 0)$ if $\boldsymbol{x}_j$ is of the same class of $\boldsymbol{x}_i$
4. cannot pass through $(\boldsymbol{x}_j, 0)$ if $\boldsymbol{x}_j$ is of the opposite class of $\boldsymbol{x}_i$



Full characterization of the Potential SVs

Proposition

$(\boldsymbol{x}_i, y_i) =$ Potential SV **if and only if** exists $(\boldsymbol{w}, b) \neq (\boldsymbol{0}, 0)$ that

1. pass through $(\boldsymbol{x}_i, 0)$
2. quasi separates $\mathcal{D}$
3. can pass through $(\boldsymbol{x}_j, 0)$ if $\boldsymbol{x}_j$ is of the same class of $\boldsymbol{x}_i$
4. cannot pass through $(\boldsymbol{x}_j, 0)$ if $\boldsymbol{x}_j$ is of the opposite class of $\boldsymbol{x}_i$

assures the datum to be in $\mathrm{PSV}(\mathcal{D})$

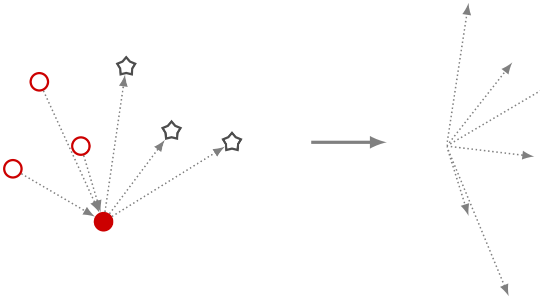
no such hyperplanes from here

Towards an alternative characterization

proposition not useful for algorithmic purposes
$\Rightarrow$ seek for alternative ones

Definition

Δ_j's of a given $(\boldsymbol{x}_i, y_i)$:



Alternative characterization of the Potential SVs

Proposition

$(\boldsymbol{x}_i, y_i)$ is Potential SV

if and only if

exists $\boldsymbol{w} \neq \mathbf{0}$ s.t.

$$\begin{cases} \Delta_n^T \boldsymbol{w} \leq 0 \\ \vdots \\ \Delta_m^T \boldsymbol{w} \leq 0 \end{cases} \qquad \begin{cases} \Delta_p^T \boldsymbol{w} < 0 \\ \vdots \\ \Delta_q^T \boldsymbol{w} < 0 \end{cases}$$

(data of the same class) *(data of the opposite class)*

Alternative characterization of the Potential SVs

Proposition

$(\boldsymbol{x}_i, y_i)$ is Potential SV

if and only if

exists $\boldsymbol{w} \neq \mathbf{0}$ s.t.

$$\begin{cases} \Delta_n^T \boldsymbol{w} \leq 0 \\ \vdots \\ \Delta_m^T \boldsymbol{w} \leq 0 \end{cases} \qquad \begin{cases} \Delta_p^T \boldsymbol{w} < 0 \\ \vdots \\ \Delta_q^T \boldsymbol{w} < 0 \end{cases}$$

(data of the same class) *(data of the opposite class)*

Corollary *(well known in literature)*

$(\boldsymbol{x}_i, y_i)$ discardable if $\boldsymbol{x}_i$ in the ***interior*** of the convex hull of the data of the same class

Towards a fast and implementable algorithm

$$\text{“exists } \boldsymbol{w} \neq \mathbf{0} \text{ s.t. } \begin{cases} \Delta_n^T \boldsymbol{w} \leq 0 \\ \vdots \\ \Delta_m^T \boldsymbol{w} \leq 0 \end{cases} \qquad \begin{cases} \Delta_p^T \boldsymbol{w} < 0 \\ \vdots \\ \Delta_q^T \boldsymbol{w} < 0 \end{cases} \text{”}$$

not fast to be checked numerically & not intuitive

Towards a fast and implementable algorithm

"exists $\boldsymbol{w} \neq \mathbf{0}$ s.t. $\begin{cases} \Delta_n^T \boldsymbol{w} \leq 0 \\ \vdots \\ \Delta_m^T \boldsymbol{w} \leq 0 \end{cases}$ $\begin{cases} \Delta_p^T \boldsymbol{w} < 0 \\ \vdots \\ \Delta_q^T \boldsymbol{w} < 0 \end{cases}$"

not fast to be checked numerically & not intuitive

more intuitive & faster to check

(we'll see why in 2 slides):

"exists $\boldsymbol{w} \neq \mathbf{0}$ s.t. $\begin{cases} \Delta_n^T \boldsymbol{w} \leq 0 \\ \vdots \\ \Delta_q^T \boldsymbol{w} \leq 0 \end{cases}$"

corresponds to check if $\operatorname{span}\langle\{\Delta_j\}\rangle = \mathbb{R}^d$

Towards a fast and implementable algorithm

"exists $\boldsymbol{w} \neq \mathbf{0}$ s.t. $\begin{cases} \Delta_n^T \boldsymbol{w} \leq 0 \\ \vdots \\ \Delta_m^T \boldsymbol{w} \leq 0 \end{cases}$ $\begin{cases} \Delta_p^T \boldsymbol{w} < 0 \\ \vdots \\ \Delta_q^T \boldsymbol{w} < 0 \end{cases}$"

not fast to be checked numerically & not intuitive

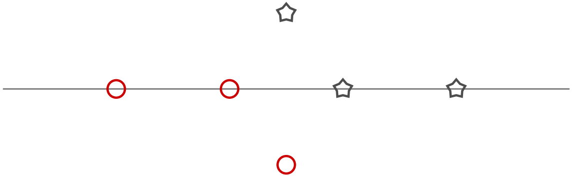
more intuitive & faster to check

(we'll see why in 2 slides):

"exists $\boldsymbol{w} \neq \mathbf{0}$ s.t. $\begin{cases} \Delta_n^T \boldsymbol{w} \leq 0 \\ \vdots \\ \Delta_q^T \boldsymbol{w} \leq 0 \end{cases}$"

is it wrong to use the latter?

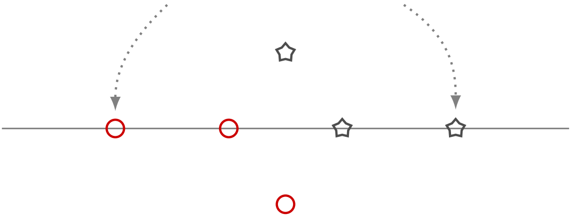
Differences between the two conditions



Differences between the two conditions

do *not* satisfy "<" condition

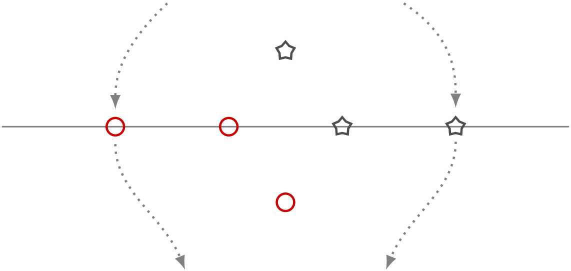
do satisfy "≤" one



Differences between the two conditions

do *not* satisfy "$<$" condition

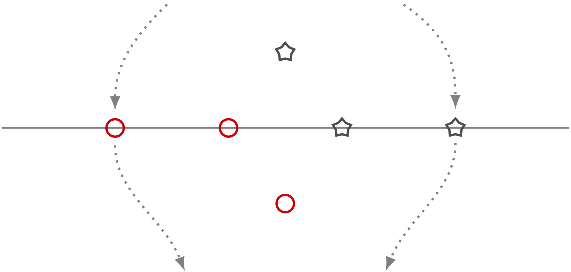
do satisfy "$\leq$" one



$\mathcal{D}$ separable $\Rightarrow$ they are not PSVs

Differences between the two conditions

do *not* satisfy "$<$" condition
do satisfy "$\leq$" one



$\mathcal{D}$ separable $\Rightarrow$ they are not PSVs

Proposition

The measure of the set of input locations that satisfy "$\leq$" condition but not "$<$" one is zero

The algorithm

1. consider a $(\boldsymbol{x}_i, y_i)$

The algorithm

1. consider a $(\boldsymbol{x}_i, y_i)$
2. compute the Δ_js

The algorithm

1. consider a $(\boldsymbol{x}_i, y_i)$
2. compute the Δ_js
3. consider the problem

$$\begin{aligned} \max. \quad & \omega_n + \ldots + \omega_q \\ \text{s.t.} \quad & \begin{cases} \Delta_j^T \boldsymbol{w} + \omega_j \leq 0 \\ \omega_j \geq 0 \end{cases} \qquad j = n, \ldots, q \end{aligned}$$

(feasibile if and only if "$\leq$" condition holds)

The algorithm

1. consider a $(\boldsymbol{x}_i, y_i)$
2. compute the Δ_js
3. consider the problem

$$\begin{aligned} \max . \quad & \omega_n + \ldots + \omega_q \\ \text{s.t.} \quad & \begin{cases} \Delta_j^T \boldsymbol{w} + \omega_j \leq 0 \\ \omega_j \geq 0 \end{cases} \qquad j = n, \ldots, q \end{aligned}$$

(feasibile if and only if "$\leq$" condition holds)

4. apply ***just one simplex step*** starting from $\boldsymbol{w} = \mathbf{0}$, $\omega_n = \ldots = \omega_p = 0$

(i.e. check if it is possible to move from the origin)

Some remarks

- the algorithm returns just the set of Potential SVs with probability one *(under mild assumptions)*

Some remarks

- the algorithm returns just the set of Potential SVs with probability one *(under mild assumptions)*
- the algorithm is ***optimal*** under information contents points of view:

no algorithms can return better answers

Some remarks

- the algorithm returns just the set of Potential SVs with probability one *(under mild assumptions)*
- the algorithm is ***optimal*** under information contents points of view:

no algorithms can return better answers

improvements possible only under
computational complexity points of view

Some remarks

- the algorithm returns just the set of Potential SVs with probability one *(under mild assumptions)*

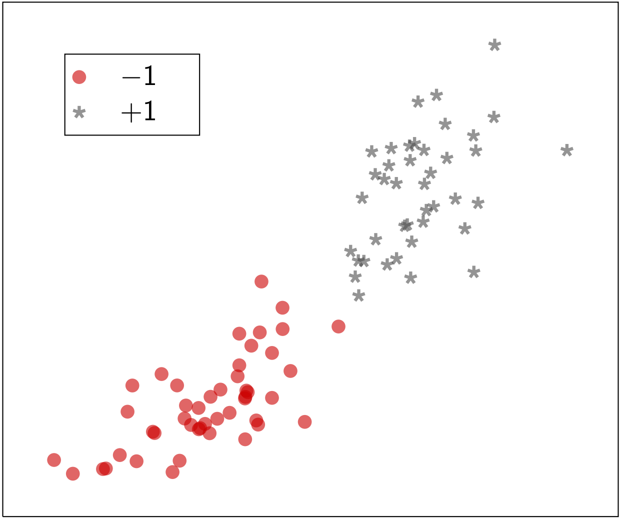
- the algorithm is ***optimal*** under information contents points of view:

no algorithms can return better answers

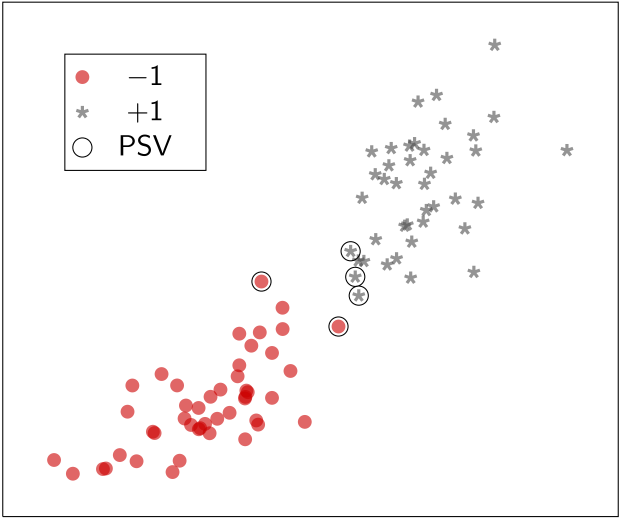
improvements possible only under
computational complexity points of view

- computational complexity $\propto$ complexity of simplex algorithm

A numerical example

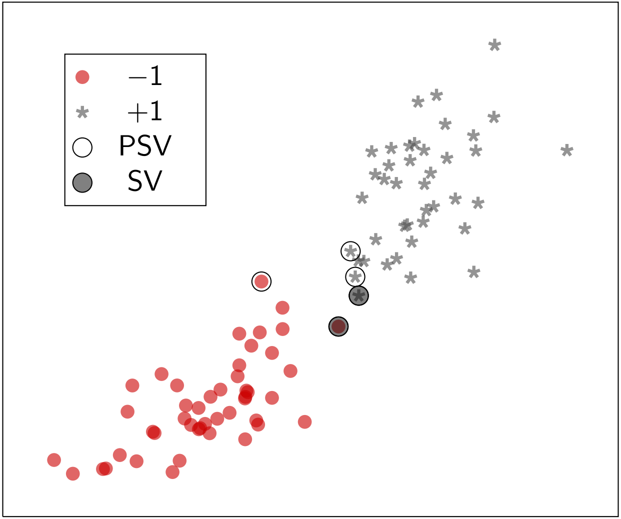


A numerical example



training not required to compute Potential SVs

A numerical example



future training can consider just Potential SVs

Summary

- considered separable datasets
- introduced the concept of ***Potential Support Vectors***
- saw that data that are not Potential SVs bring no information
- Potential SVs can be computed
  - ***before training steps***
  - ***iteratively***
  - ***exploiting just one simplex step per datum***

Future works

- extend results for non-separable datasets
- (analytically) check whether Potential SVs can speed-up training strategies
  (e.g., embed PSVs in SMO strategies)

On the discardability of data in Support Vector Classification problems

Simone Del Favero, **Damiano Varagnolo**, Francesco Dinuzzo,
Luca Schenato, Gianluigi Pillonetto

Department of Information Engineering – Padova, Italy
Max Planck Institute – Tübingen, Germany

December 13^{th}, 2011 – 50^{th} IEEE CDC

varagnolo@dei.unipd.it
www.dei.unipd.it/∼varagnolo/
google: damiano varagnolo