- Support Vector Machines (SVMs).
- Semi-Supervised Learning.
    - Semi-Supervised SVMs.

Maria-Florina Balcan

03/25/2015

Support Vector Machines (SVMs).

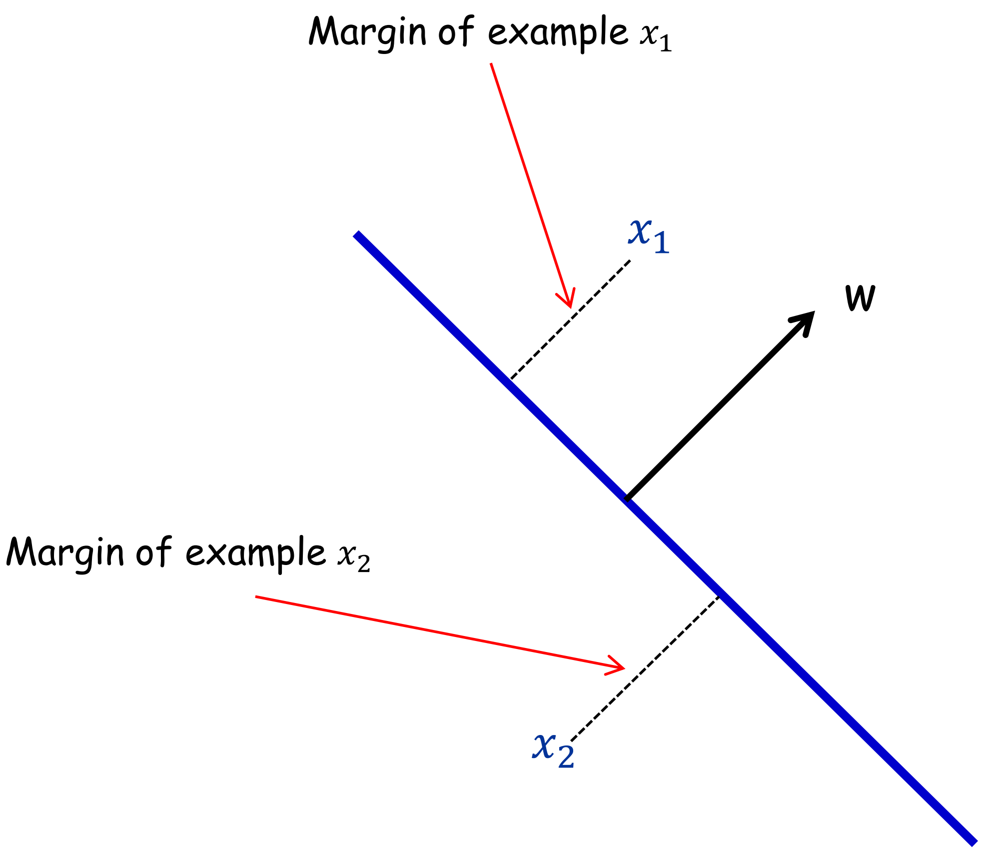
One of the most theoretically well motivated and practically most effective classification algorithms in machine learning.

Directly motivated by Margins and Kernels!

Geometric Margin

WLOG homogeneous linear separators $[w_0 = 0]$.

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$.



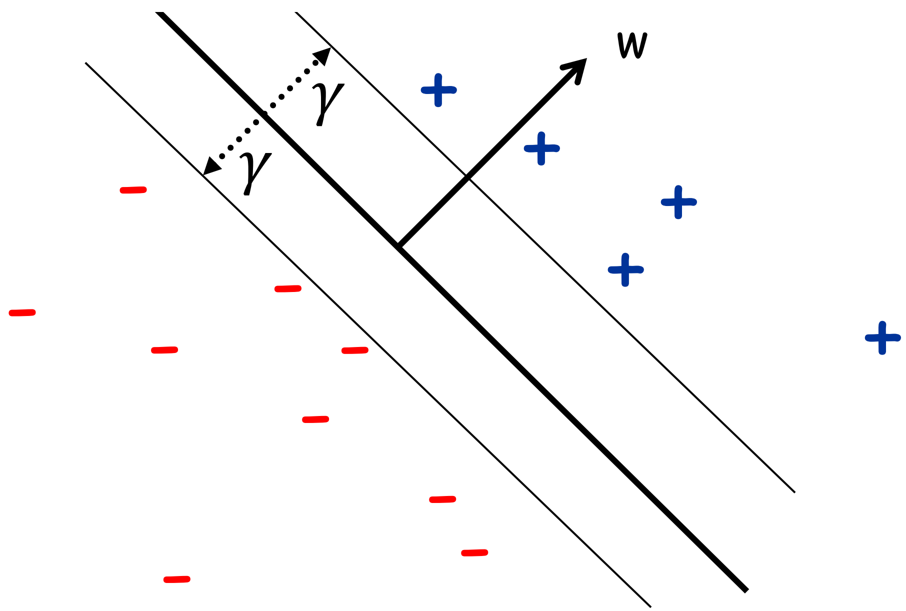
If $||w|| = 1$, margin of x w.r.t. w is $|x \cdot w|$.

Geometric Margin

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$.

Definition: The margin γ_w of a set of examples S wrt a linear separator w is the smallest margin over points $x \in S$.

Definition: The margin γ of a set of examples S is the maximum γ_w over all linear separators w.

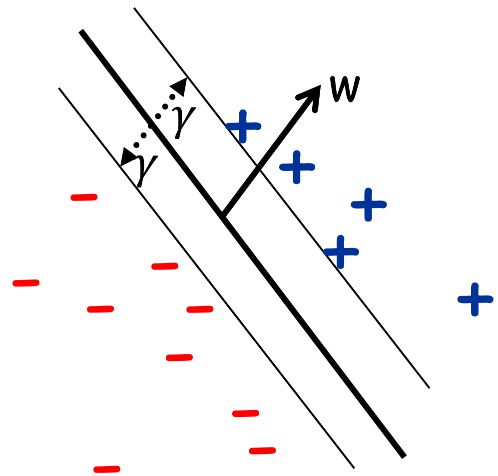


Margin Important Theme in ML

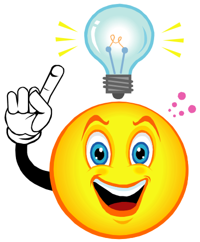
Both sample complexity and algorithmic implications.

Sample/Mistake Bound complexity:

- If large margin, # mistakes Peceptron makes is small (independent on the dim of the space)!
- If large margin γ and if alg. produces a large margin classifier, then amount of data needed depends only on R/γ [Bartlett & Shawe-Taylor '99].



Algorithmic Implications

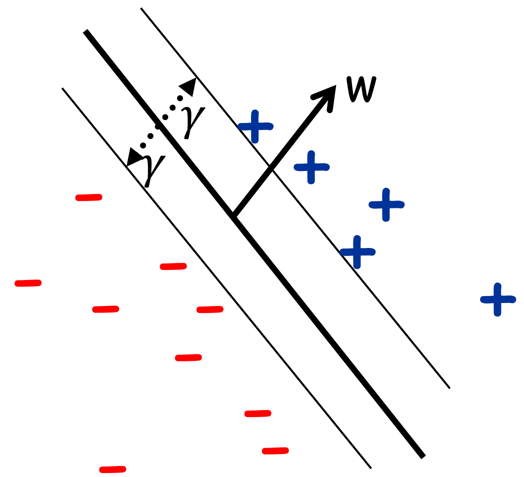


Suggests searching for a large margin classifier… SVMs

Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

First, assume we know a lower bound on the margin γ



Input: γ, $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

Find: some w where:

- $||w||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

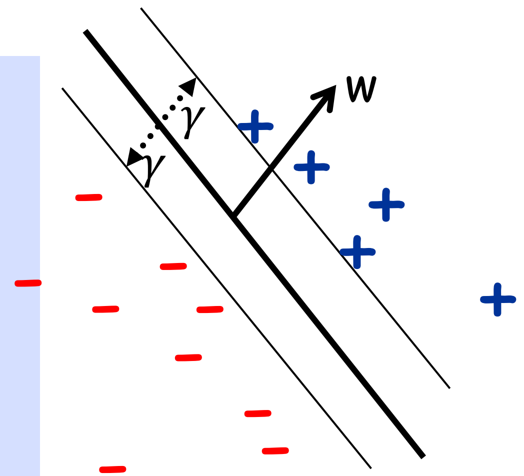
Output: w, a separator of margin γ over S

Realizable case, where the data is linearly separable by margin γ

Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

E.g., search for the best possible γ



Input: S={(x_1, y_1), …,$(\mathrm{x}_\mathrm{m}, y_\mathrm{m})$};

Find: some w and maximum γ where:

- $||\mathrm{w}||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

Output: maximum margin separator over S

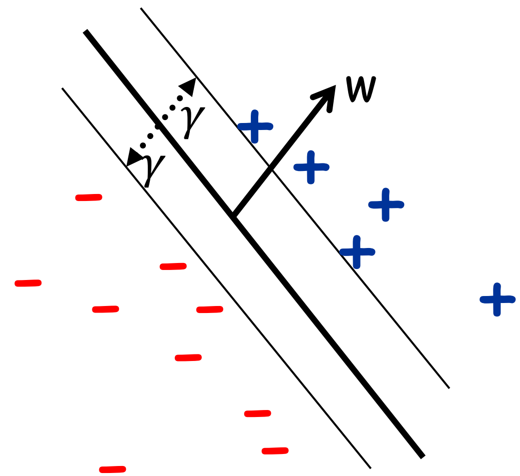
Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

Input: S={(x_1, y_1), …,(x_m, y_m)};

Maximize γ under the constraint:

- $||w||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$



Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

<u>Input</u>: S={(x_1, y_1), …,$(\mathrm{x}_\mathrm{m}, y_\mathrm{m})$};

Maximize γ under the constraint:

- $||\mathrm{w}||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

objective function

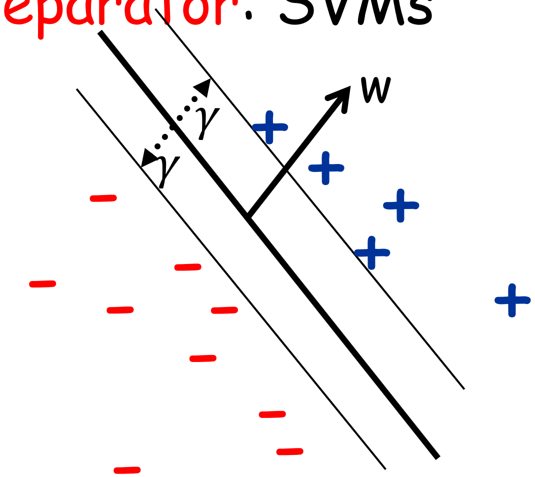
constraints

This is a **constrained optimization** problem.

- Famous example of constrained optimization: **linear programming**, where objective fn is linear, constraints are linear (in)equalities

Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs



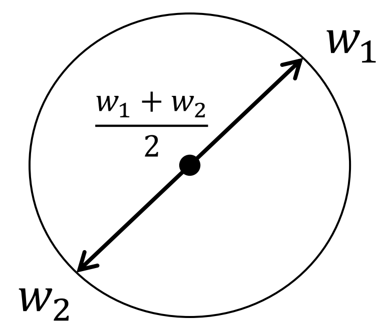
Input: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

Maximize γ under the constraint:

- $||w||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

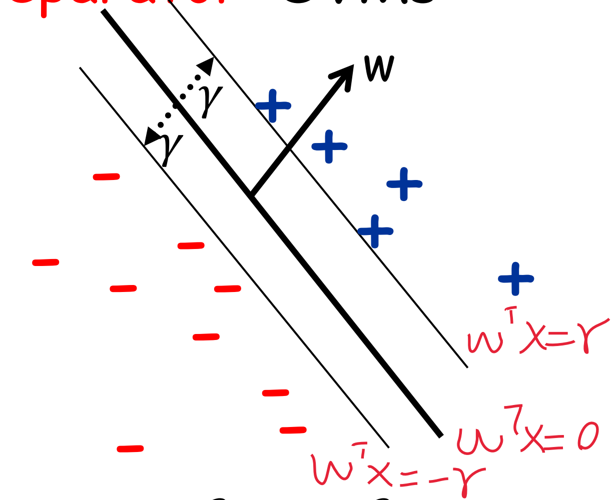
This constraint is non-linear.

In fact, it's even non-convex



Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

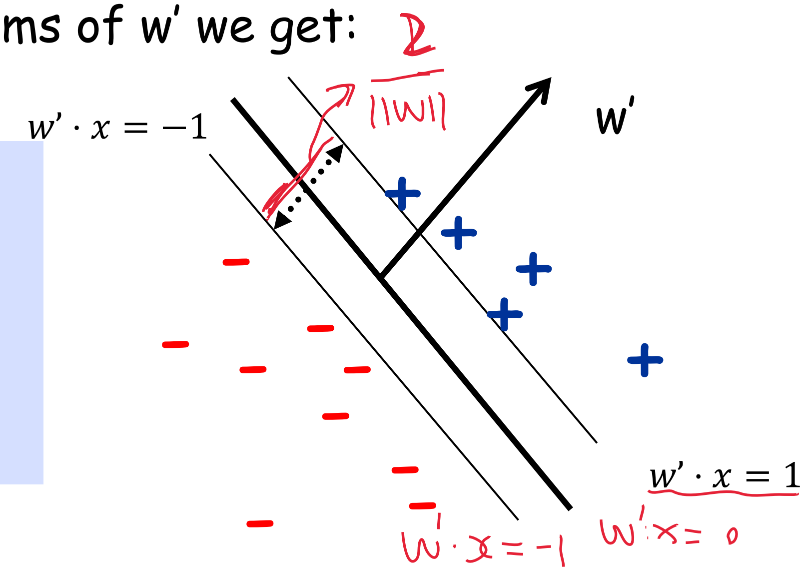


Input: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

Maximize γ under the constraint:

- $||w||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

$w' = w/\gamma$, then max γ is equiv. to minimizing $||w'||^2$ (since $||w'||^2 = 1/\gamma^2$).
So, dividing both sides by γ and writing in terms of w′ we get:



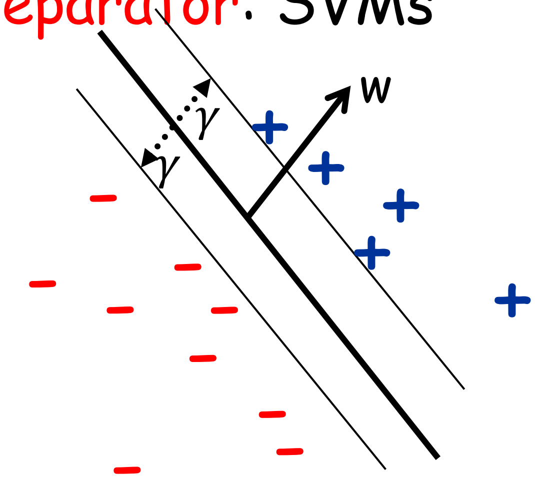
Input: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

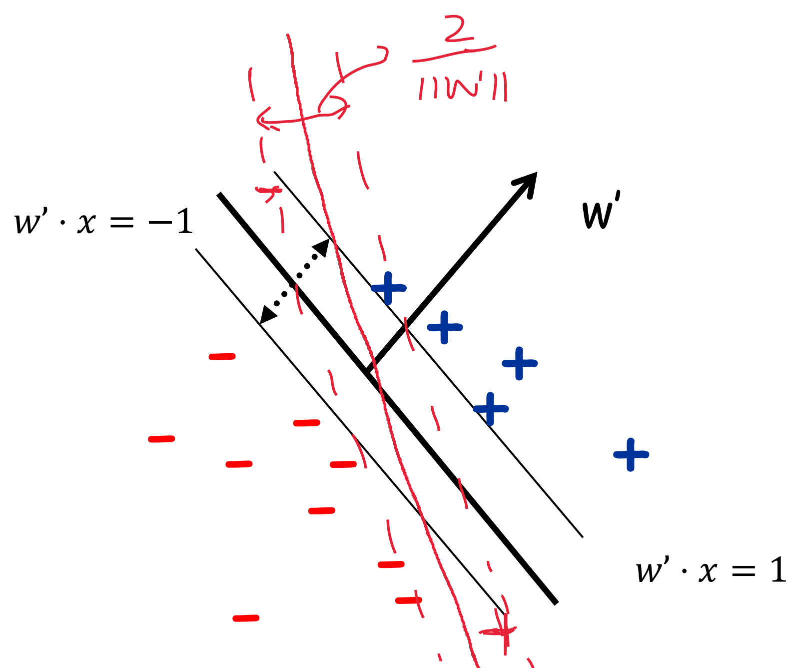
Minimize $||w'||^2$ under the constraint:

- For all i, $y_i w' \cdot x_i \geq 1$

Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs





Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

Input: $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$;

$\mathrm{argmin}_w \, ||w||^2$ s.t.:

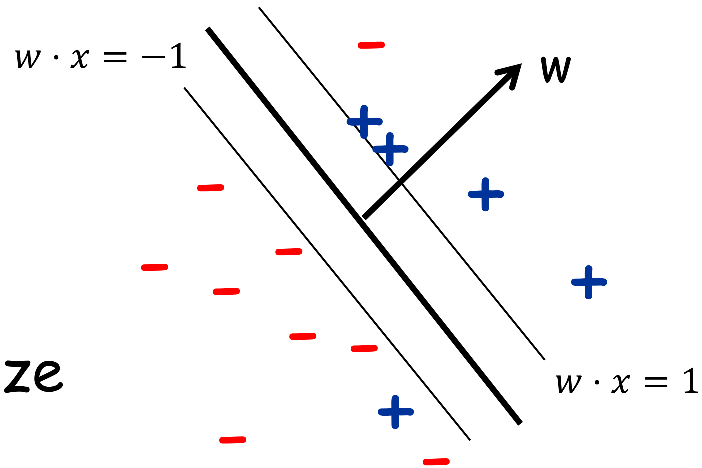
- For all i, $y_i w \cdot x_i \geq 1$

This is a **constrained optimization** problem.

- The objective is convex (quadratic)
- All constraints are linear
- Can solve efficiently (in poly time) using standard **quadratic programing** (QP) software

Support Vector Machines (SVMs)

Question: what if data isn't perfectly linearly separable?



Issue 1: now have two objectives

- maximize margin
- minimize # of misclassifications.

Ans 1: Let's optimize their sum: minimize

$\min_{w} \; ||w||^2 + C(\text{\# misclassifications})$

where C is some tradeoff constant.

Issue 2: This is computationally hard (NP-hard). 

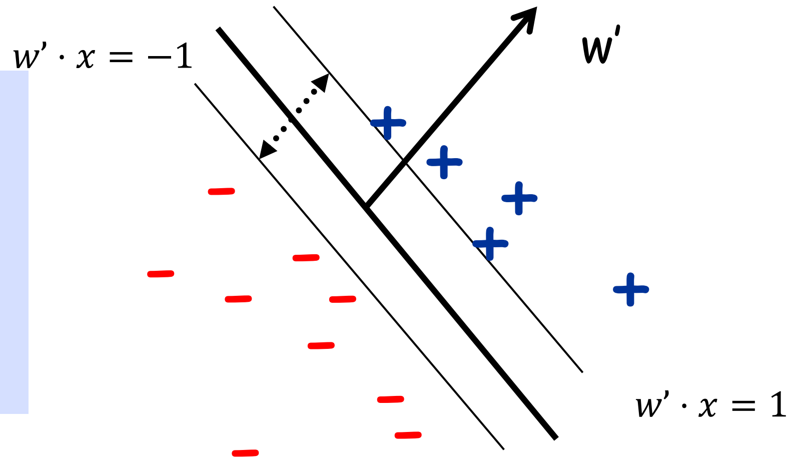
[even if didn't care about margin and minimized # mistakes]

NP-hard [Guruswami-Raghavendra'06]

Support Vector Machines (SVMs)

Question: what if data isn't perfectly linearly separable?

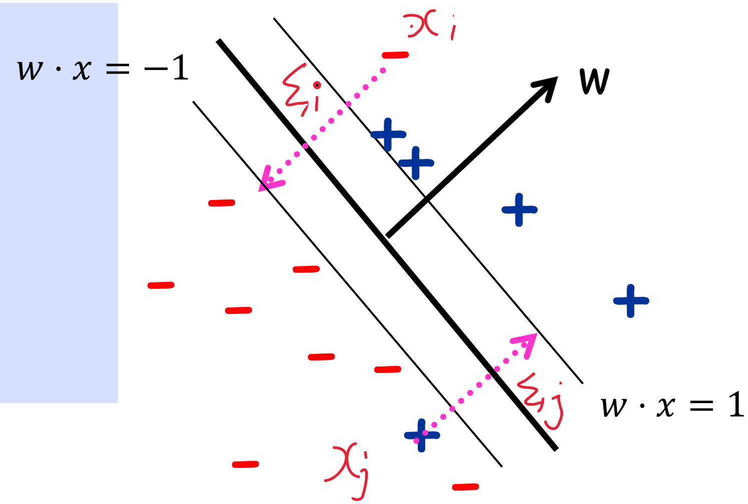
Replace "# mistakes" with upper bound called "hinge loss"



<u>Input</u>: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

Minimize $||w'||^2$ under the constraint:

- For all i, $y_i w' \cdot x_i \geq 1$



<u>Input</u>: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

<u>Find</u> $\text{argmin}_{w, \xi_1, \ldots, \xi_m} ||w||^2 + C \sum_i \xi_i$ s.t.:

- For all i, $y_i w \cdot x_i \geq 1 - \xi_i$

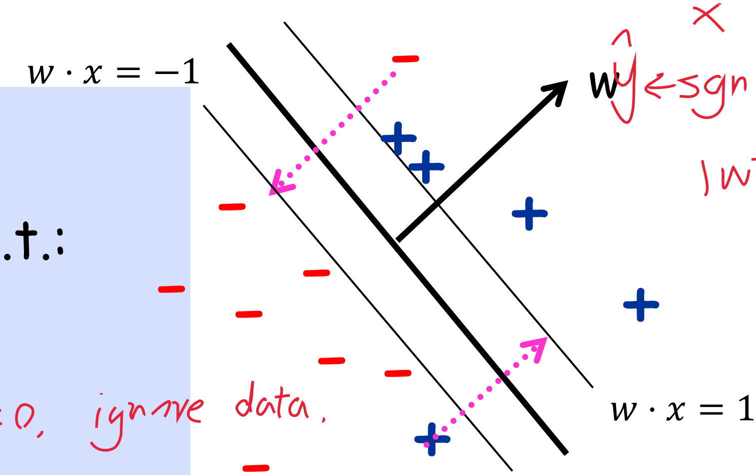
  $\xi_i \geq 0$

ξ_i are "slack variables"

Support Vector Machines (SVMs)

Question: what if data isn't perfectly linearly separable?

Replace "# mistakes" with upper bound called "hinge loss"



$\hat{y} \leftarrow \mathrm{sgn}(w^T\hat{x})$

$|w^T x|$

Input: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

Find $\mathrm{argmin}_{w,\xi_1,\ldots,\xi_m} ||w||^2 + C \sum_i \xi_i$ s.t.:

- For all i, $y_i w \cdot x_i \geq 1 - \xi_i$

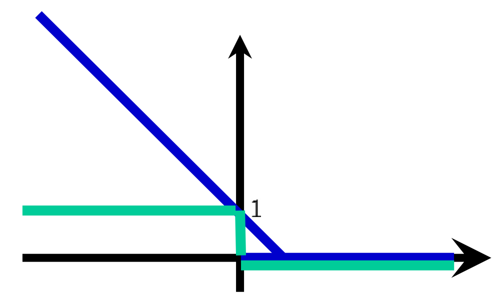
  $\xi_i \geq 0$

C=0, ignore data.

C=∞, have to separate data

ξ_i are "slack variables"

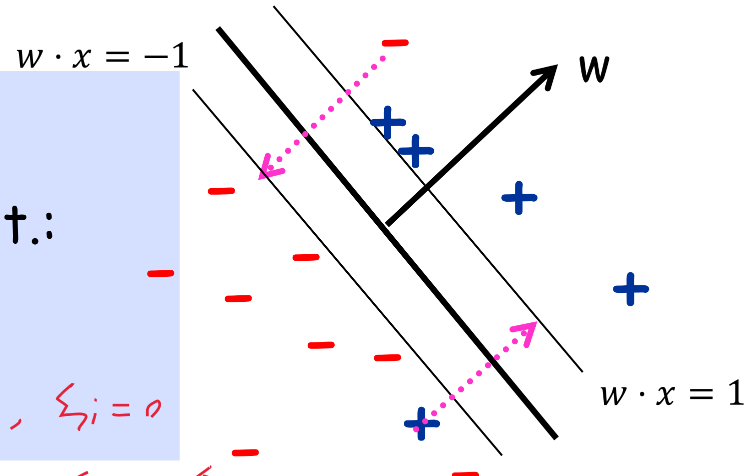
C controls the relative weighting between the twin goals of making the $||w||^2$ small (margin is large) and ensuring that most examples have functional margin ≥ 1.



$l(w, x, y) = \max(0, 1 - y\, w \cdot x)$

Support Vector Machines (SVMs)

Question: what if data isn't perfectly linearly separable?
Replace "# mistakes" with upper bound called "hinge loss"



Input: S={$(x_1, y_1), \ldots, (x_m, y_m)$};

Find $\text{argmin}_{w,\xi_1,\ldots,\xi_m} ||w||^2 + C\sum_i \xi_i$ s.t.:

- For all i, $y_i w \cdot x_i \geq 1 - \xi_i$

$\xi_i \geq 0$

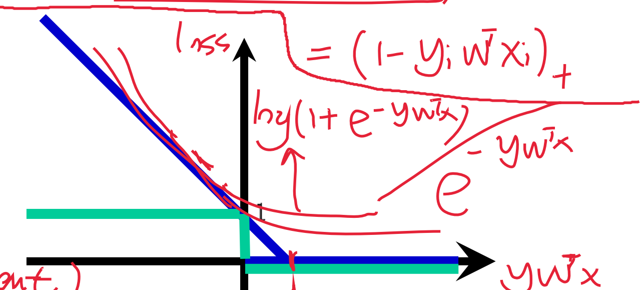
$y_i w^T x_i \geq 1, \ \xi_i = 0$

$y_i w^T x_i < 1, \ \xi_i > 0 \Leftrightarrow \xi_i = \max(0, 1 - y_i w^T x_i)$

$= (1 - y_i w^T x_i)_+$

Replace the number of mistakes with the hinge loss

$||w||^2 + C(\text{\# misclassifications})$



(subgradient)

$l(w, x, y) = \max(0, 1 - y\, w \cdot x)$

tight upper bound

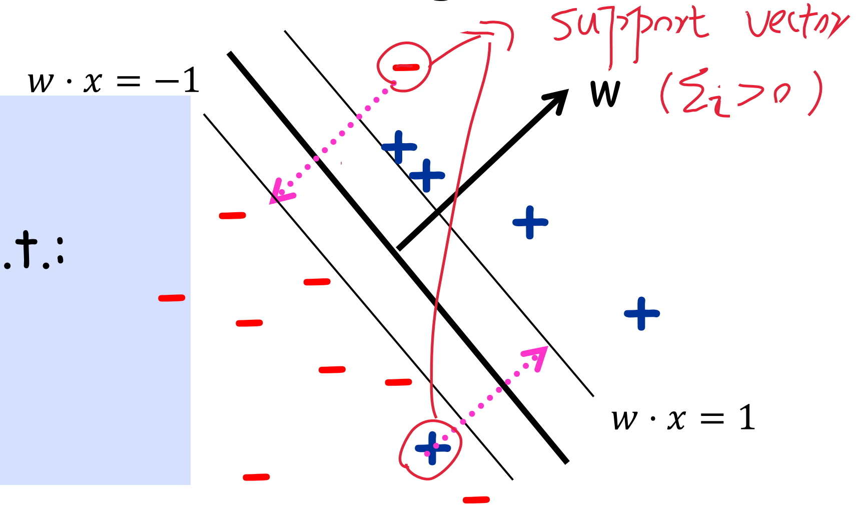
$\min_w \ ||w||^2 + C\sum_{i=1}^{n}(1 - y_i w^T x_i)_+$

$w \leftarrow w - \eta \nabla h(w)$

Support Vector Machines (SVMs)

Question: what if data isn't perfectly linearly separable?

Replace "# mistakes" with upper bound called "hinge loss"



<u>Input</u>: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

<u>Find</u> $\mathrm{argmin}_{w,\xi_1,\ldots,\xi_m} ||w||^2 + C \sum_i \xi_i$ s.t.:

- For all i, $y_i w \cdot x_i \geq 1 - \xi_i$

  $\xi_i \geq 0$

ξ_i are "slack variables"

Support Vector Machines (SVMs)

Question: what if data isn't perfectly linearly separable?

Replace "# mistakes" with upper bound called "hinge loss"

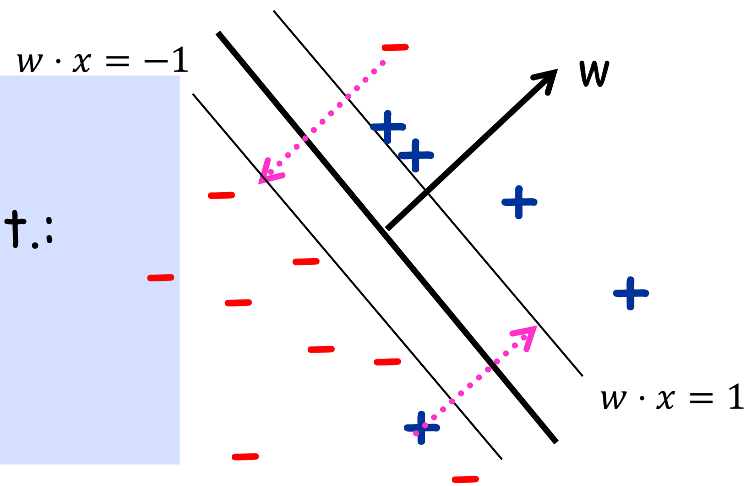
Primal Form

Input: S={(x_1, y_1), …,(x_m, y_m)};

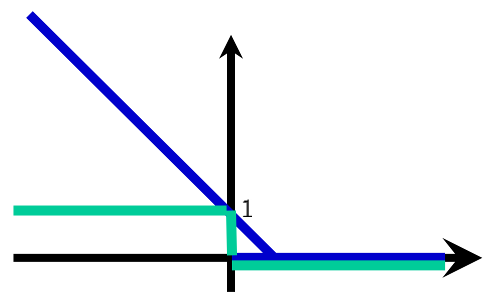
Find $\mathrm{argmin}_{w,\xi_1,\ldots,\xi_m} ||w||^2 + C\sum_i \xi_i$ s.t.:

- For all i, $y_i w \cdot x_i \geq 1 - \xi_i$

  $\xi_i \geq 0$



Total amount have to move the points to get them on the correct side of the lines $w \cdot x = +1/-1$, where the distance between the lines $w \cdot x = 0$ and $w \cdot x = 1$ counts as "1 unit".



$l(w, x, y) = \max(0, 1 - y\, w \cdot x)$

What if the data is far from being linearly separable?

$w \leftarrow$

Example:  vs 

No good linear separator in pixel representation.

$\underline{k(x_i, x_j)} = \langle \phi(x_i), \phi(x_j) \rangle$

$w^T x = 0$

$k(\langle x_i, x_j \rangle)$

$\phi: R^d \rightarrow R^p \quad (p \gg d)$

SVM philosophy: "Use a Kernel"

SVMs -- Primal Form and Dual Form

(perfectly linear separable)

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

s.t. $y_i w^T x_i \geq 1, \ \forall i \in [n] \leftarrow \alpha_i$

$y_i(w^T x_i + b) \geq 1$

Lagrangian.

$$L(w,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i w^T x_i - 1)$$

$$p^* = \min_{w} \max_{\alpha} L(w,\alpha) \quad \text{(Primal)}$$

$$d^* = \max_{\alpha} \min_{w} L(w,\alpha) \quad \text{(Dual)}$$

$$(d^* \leq p^*) \xrightarrow{KKT} d^* = p^*$$

KKT.

- Stationary. $\frac{\partial L}{\partial w} = 0, \ \frac{\partial L}{\partial \alpha} = 0$
- complementary: $\alpha_i (y_i w^T x_i - 1) = 0$
- primal: $y_i w^T x_i \geq 1, \ \forall i$
- dual: $\alpha_i \geq 0, \ \forall i$

$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$\hat{y} \leftarrow \mathrm{sgn}(w^T \hat{x})$

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle$$

$k(\langle x_i, x_j \rangle)$

s.t. $\alpha_i \geq 0, \ \forall i$

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

Dual

Support Vector Machines (SVMs)

Input: S={(x_1, y_1), …,(x_m, y_m)};

Find $\mathrm{argmin}_{w,\xi_1,\ldots,\xi_m} ||w||^2 + C\sum_i \xi_i$ s.t.:

- For all i, $y_i w \cdot x_i \geq 1 - \xi_i$

  $\xi_i \geq 0$

Primal form

Which is equivalent to:

Input: S={(x_1, y_1), …,(x_m, y_m)};

Find $\mathrm{argmin}_\alpha \frac{1}{2}\sum_i \sum_j y_i y_j \alpha_i \alpha_j x_i \cdot x_j - \sum_i \alpha_i$ s.t.:

- For all i, $\underline{0 \leq \alpha_i \leq \cancel{C_i}\ C}$

  $\sum_i y_i \alpha_i = 0$

Lagrangian Dual

Sample sparsity

$w = \sum_{i=1}^{n} \alpha_i y_i x_i$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \underline{C = \alpha_i + \lambda_i}$$

$$\Rightarrow \lambda_i = C - \alpha_i \geq 0$$

$$\Rightarrow \alpha_i \leq C$$

Primal. $\min_{w, \xi, b} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i$

s.t. $y_i(w^T x_i + b) \geq 1 - \xi_i, \ \forall i \leftarrow \alpha_i$

$\xi_i \geq 0, \ \forall i \longleftarrow \lambda_i$

$$L(w, b, \xi, \alpha, \lambda) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left(y_i(w^T x_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{n} \lambda_i \xi_i$$

KKT condition

$$\begin{cases} \frac{\partial L}{\partial w} = 0, \ \frac{\partial L}{\partial b} = 0, \ \frac{\partial L}{\partial \xi_i} = 0, \ \frac{\partial L}{\partial \alpha} = 0, \ \frac{\partial L}{\partial \lambda} = 0 \\ \alpha_i \left(y_i(w^T x_i + b) - 1 + \xi_i\right) = 0 \\ \lambda_i \xi_i = 0 \\ y_i(w^T x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0 \\ \alpha_i \geq 0, \ \lambda_i \geq 0 \end{cases}$$

$$\begin{cases} y_i(w^T x_i + b) > 1, \ \xi_i = 0. \Rightarrow \alpha_i = 0 \\ y_i(w^T x_i + b) < 1, \ \xi_i > 0. \Rightarrow \lambda_i = 0 \Rightarrow \alpha_i = C \\ y_i(w^T x_i + b) = 1, \ \xi_i = 0 \Rightarrow \alpha_i \geq 0, \ \alpha_i \leq C \end{cases}$$

$$\Rightarrow 0 \leq \alpha_i \leq C$$

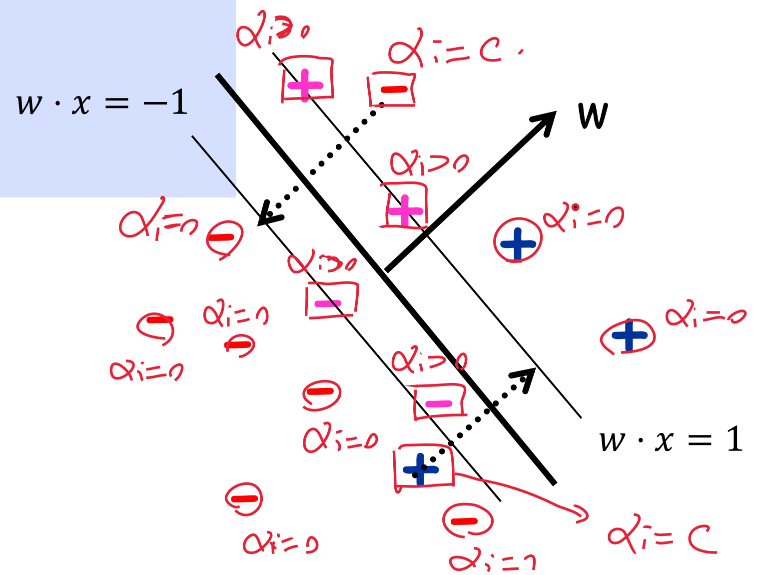
SVMs (Lagrangian Dual)

<u>Input</u>: $S=\{(x_1, y_1), \dots, (x_m, y_m)\}$;

<u>Find</u> $\text{argmin}_\alpha \frac{1}{2} \sum_i \sum_j y_i y_j \, \alpha_i \alpha_j x_i \cdot x_j - \sum_i \alpha_i$ s.t.:

- For all i, $0 \leq \alpha_i \leq C$

$$\sum_i y_i \alpha_i = 0$$



- Final classifier is: $w = \sum_i \alpha_i y_i x_i$
- The points x_i for which $\alpha_i \neq 0$ are called the "support vectors"

Kernelizing the Dual SVMs

Input: $S=\{(x_1, y_1), \dots, (x_m, y_m)\}$;

Find $\operatorname{argmin}_\alpha \frac{1}{2} \sum_i \sum_j y_i y_j \, \alpha_i \alpha_j x_i \cdot x_j - \sum_i \alpha_i$ s.t.:

- For all i, $0 \le \alpha_i \le C_i$

$$\sum_i y_i \alpha_i = 0$$

SMO

(Sequential Minimal Optimization)

Replace $x_i \cdot x_j$ with $K(x_i, x_j)$.

- Final classifier is: $w = \sum_i \alpha_i y_i x_i$
- The points x_i for which $\alpha_i \neq 0$ are called the "support vectors"
- With a kernel, classify x using $\sum_i \alpha_i y_i K(x, x_i)$

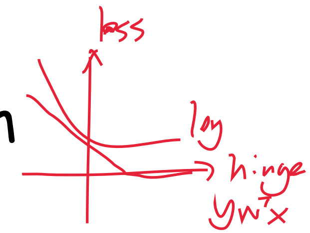
Support Vector Machines (SVMs).

One of the most theoretically well motivated and practically most effective classification algorithms in machine learning.

Directly motivated by Margins and Kernels!

What you should know

- The importance of margins in machine learning.
- The primal form of the SVM optimization problem



- The dual form of the SVM optimization problem.

RLR: $\min_{w} \sum_{i=1}^{n} \log(1+e^{-y_i w^T x_i}) + C\|w\|^2$

SVM. $\min_{w} \sum_{i=1}^{n} (1-y_i w^T x_i)_+ + C\|w\|^2$

↓ explicitly maximize margin by ignoring all sample but SVs $\left(\gamma = \frac{1}{\|w\|}\right)$

- Kernelizing SVM.
- Think about how it's related to Regularized Logistic Regression. (RLR)

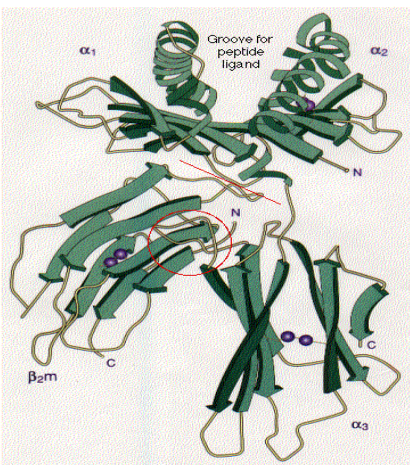
Modern (Partially) Supervised Machine Learning

- Using Unlabeled Data and Interaction for Learning

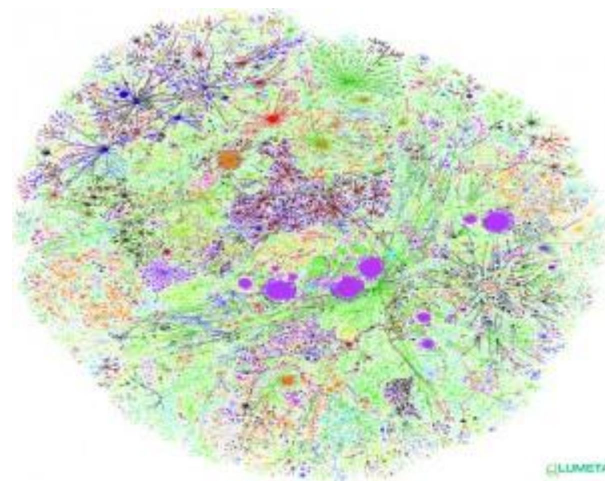
Classic Paradigm Insufficient Nowadays

Modern applications: massive amounts of raw data.

Only a tiny fraction can be annotated by human experts.



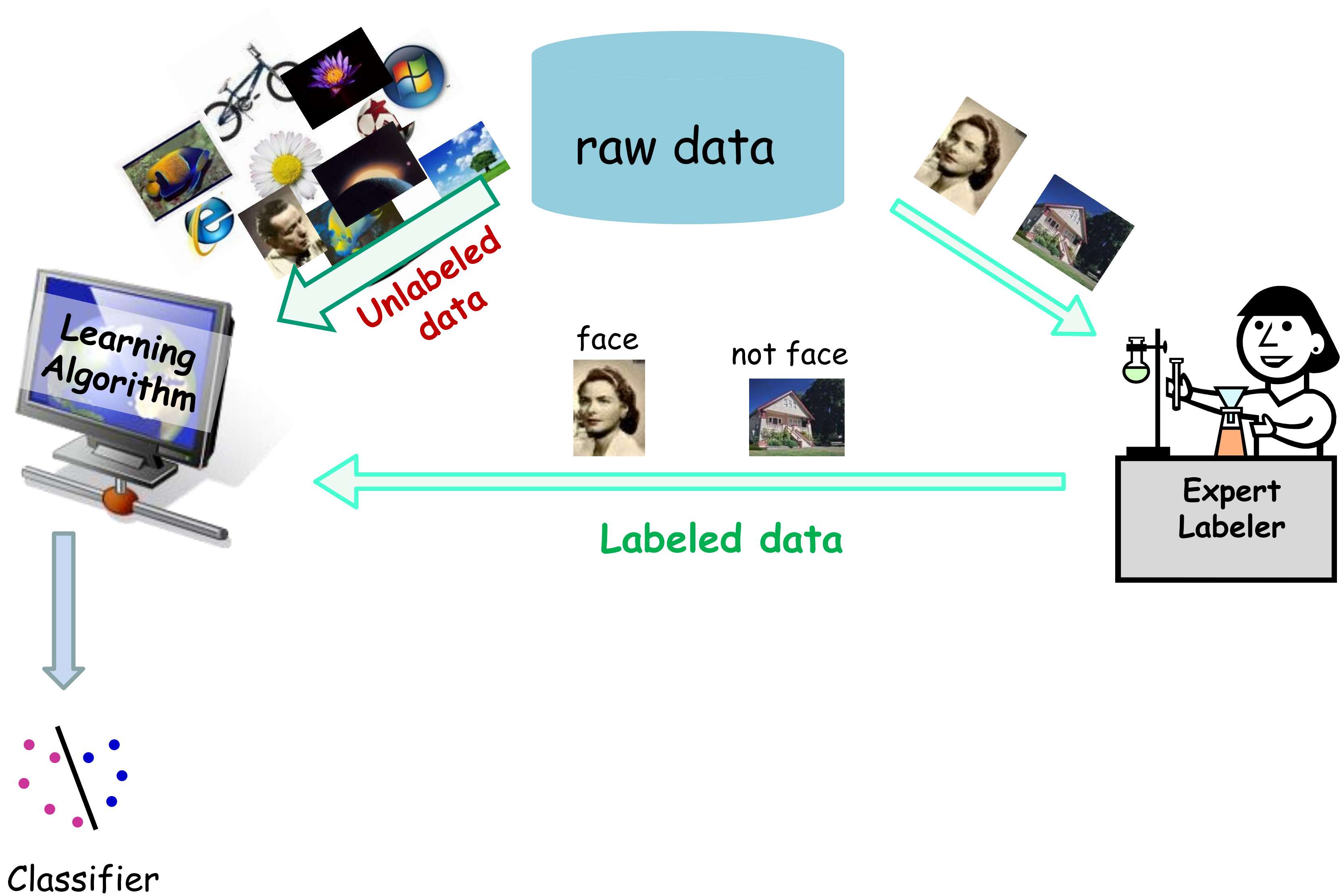
Protein sequences



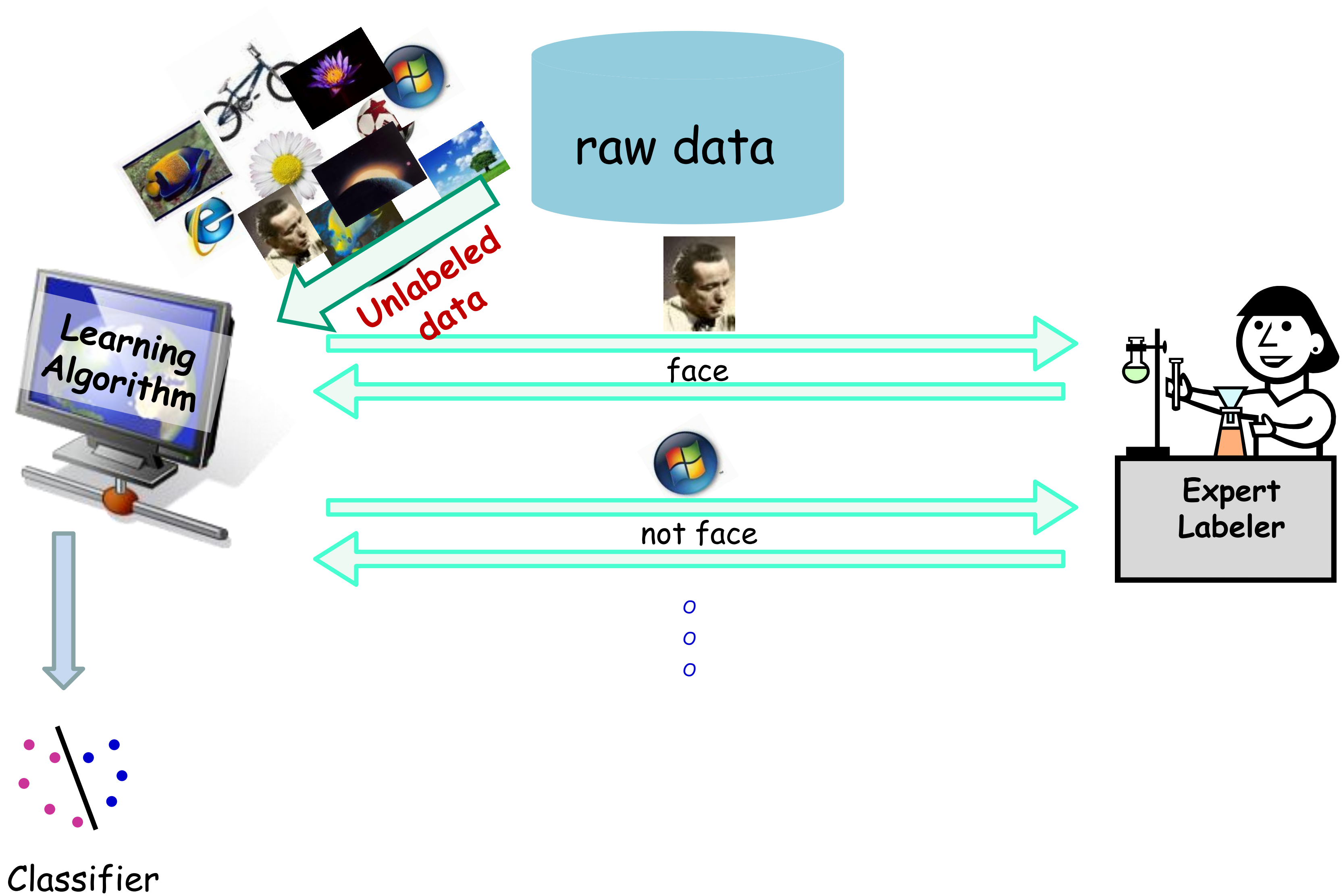
Billions of webpages

Images

Semi-Supervised Learning



Active Learning

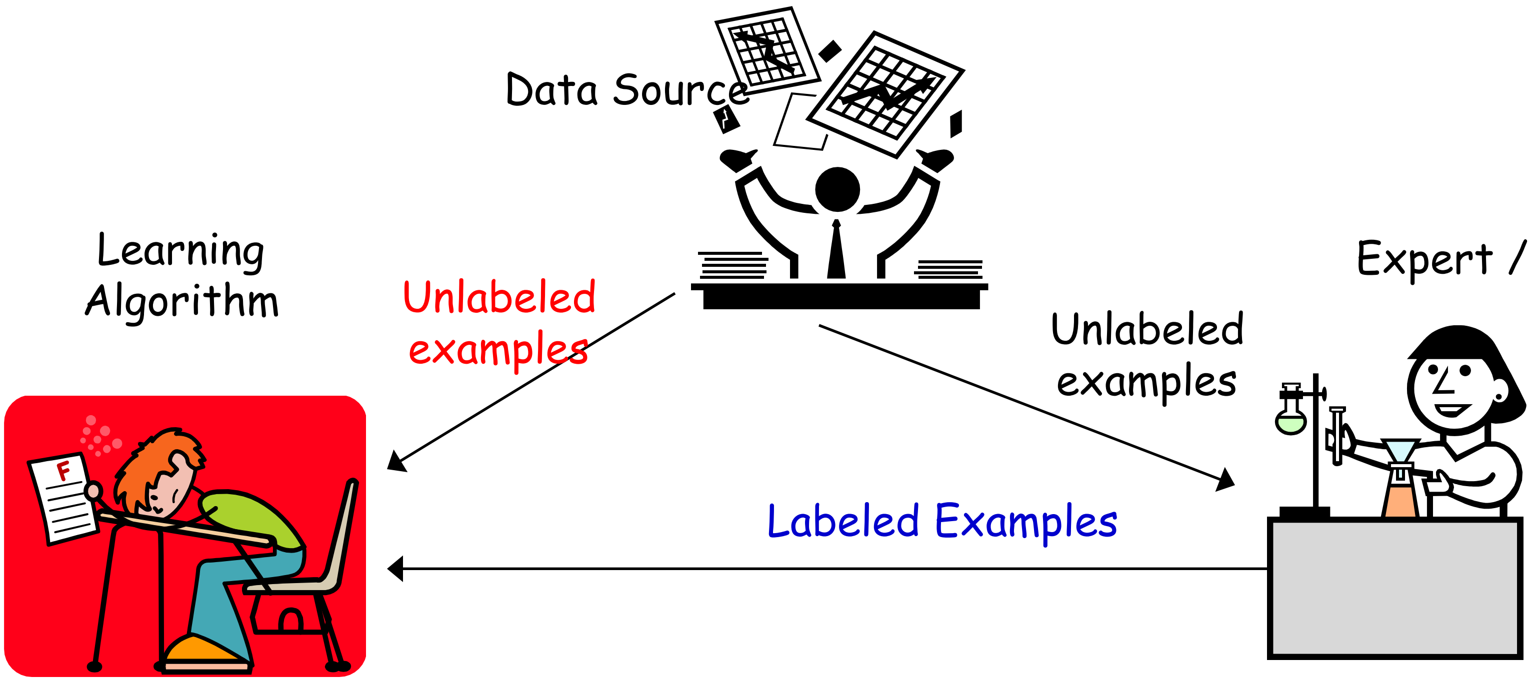


Semi-Supervised Learning

Prominent paradigm in past 15 years in Machine Learning.

- Most applications have lots of unlabeled data, but labeled data is rare or expensive:
  - Web page, document classification
  - Computer Vision
  - Computational Biology,
  - ….

Semi-Supervised Learning



Algorithm outputs a classifier

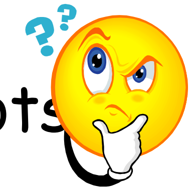
$S_l=\{(x_1,y_1), \ldots,(x_{m_l},y_{m_l})\}$

x_i drawn i.i.d from D, $y_i = c^*(x_i)$

$S_u=\{x_1, \ldots,x_{m_u}\}$ drawn i.i.d from D

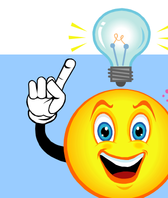
Goal: h has small error over D.

$$\mathrm{err}_D(h) = \Pr_{x\sim D}(h(x) \neq c^*(x))$$



Semi-supervised learning: no querying. Just have lots additional unlabeled data.

A bit puzzling since unclear what unlabeled data can do for you….



Key Insight

Unlabeled data useful if we have beliefs not only about the form of the target, but also about its relationship with the underlying distribution.

Combining Labeled and Unlabeled Data

- Several methods have been developed to try to use unlabeled data to improve performance, e.g.:
  - Transductive SVM [Joachims '99]
  - Co-training [Blum & Mitchell '98]
  - Graph-based methods [B&C01], [ZGL03]

Test of time awards at ICML!

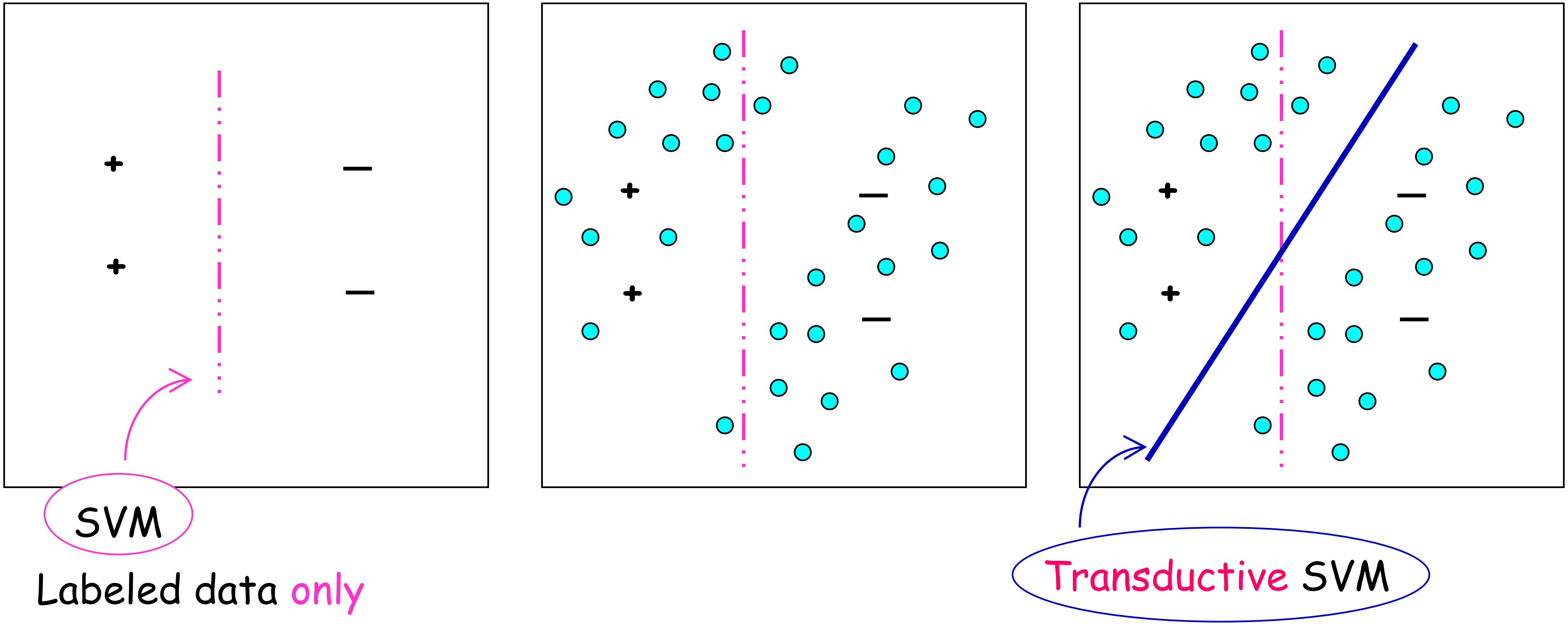
Workshops [ICML '03, ICML' 05, …]

Books:
- Semi-Supervised Learning, MIT 2006

  O. Chapelle, B. Scholkopf and A. Zien (eds)
- Introduction to Semi-Supervised Learning, Morgan & Claypool, 2009 Zhu & Goldberg

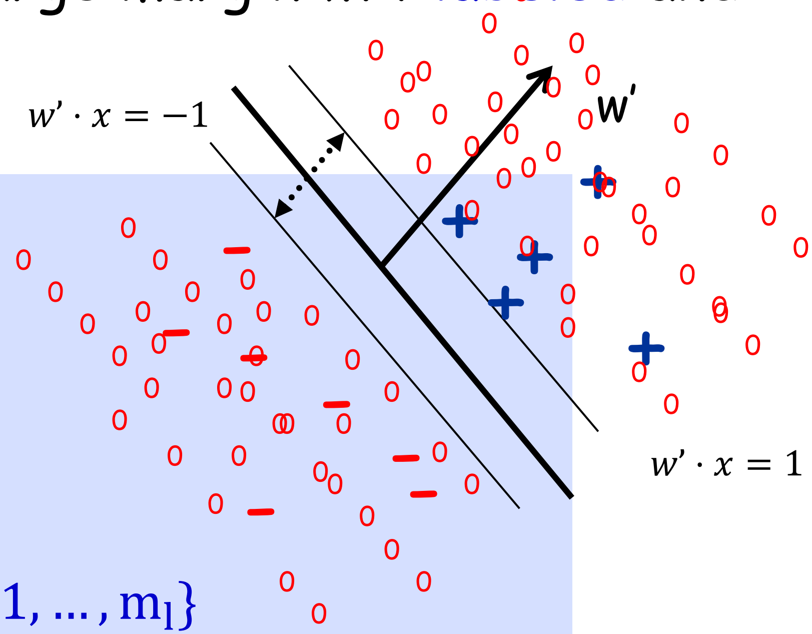
Example of "typical" assumption: Margins

- The separator goes through low density regions of the space/large margin.
  - assume we are looking for linear separator
  - belief: should exist one with large separation



Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]



Input: $S_l=\{(x_1, y_1), \dots, (x_{m_l}, y_{m_l})\}$

$S_u=\{x_1, \dots, x_{m_u}\}$

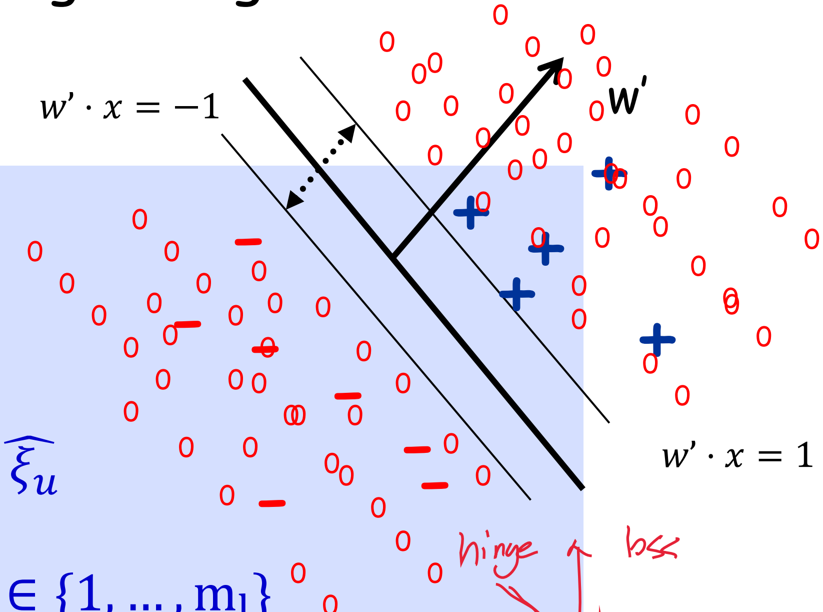
$\mathrm{argmin}_w \; ||w||^2$ s.t.:

- $y_i \, w \cdot x_i \geq 1$, for all $i \in \{1, \dots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1$, for all $u \in \{1, \dots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \dots, m_u\}$

Find a labeling of the unlabeled sample and w s.t. w separates both labeled and unlabeled data with maximum margin.

Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]

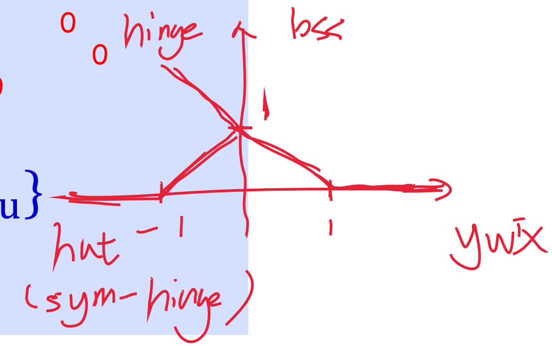


Input: $S_l=\{(x_1,y_1), \ldots,(x_{m_l},y_{m_l})\}$

$S_u=\{x_1, \ldots,x_{m_u}\}$

$$\mathrm{argmin}_w \; ||w||^2 + C \sum_i \xi_i + C \sum_u \widehat{\xi_u}$$

- $y_i \, w \cdot x_i \geq 1-\xi_i$, for all $i \in \{1, \ldots, m_l\}$
- $\widehat{y_i} w \cdot x_u \geq 1 - \widehat{\xi_u}$, for all $u \in \{1, \ldots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$



Find a labeling of the unlabeled sample and w s.t. w separates both labeled and unlabeled data with maximum margin.

Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data.

Input: $S_l=\{(x_1,y_1), \ldots,(x_{m_l},y_{m_l})\}$

$S_u=\{x_1, \ldots,x_{m_u}\}$

$$\mathrm{argmin}_w \, ||w||^2 + C\sum_i \xi_i + C\sum_u \widehat{\xi_u}$$

- $y_i \, w \cdot x_i \geq 1-\xi_i$, for all $i \in \{1, \ldots, m_l\}$
- $\widehat{y}_i w \cdot x_u \geq 1 - \widehat{\xi_u}$, for all $u \in \{1, \ldots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$

NP-hard..... Convex only after you guessed the labels… too many possible guesses…

Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data.

<u>Heuristic (Joachims) high level idea:</u>

- First maximize margin over the labeled points
- Use this to give initial labels to unlabeled points based on this separator.
- Try flipping labels of unlabeled points to see if doing so can increase margin

Keep going until no more improvements. Finds a locally-optimal solution.

Experiments [Joachims99]

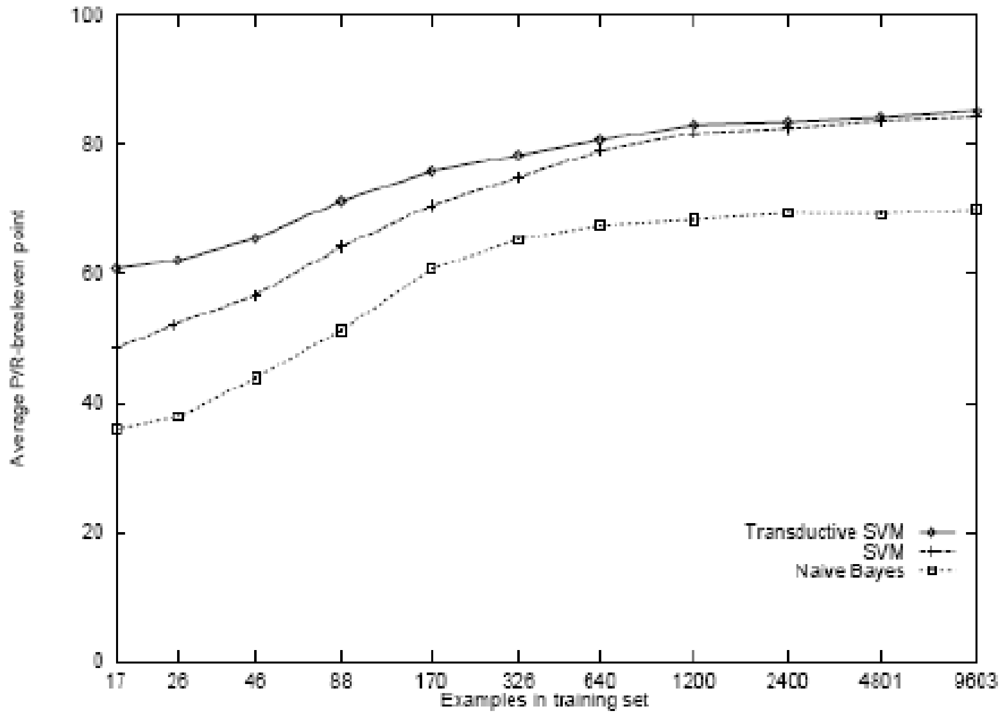
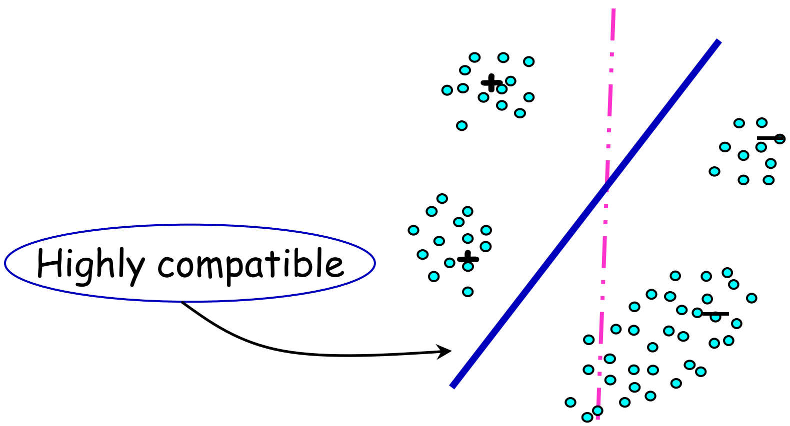


Figure 6: Average P/R-breakeven point on the Reuters dataset for different training set sizes and a test set size of 3,299.

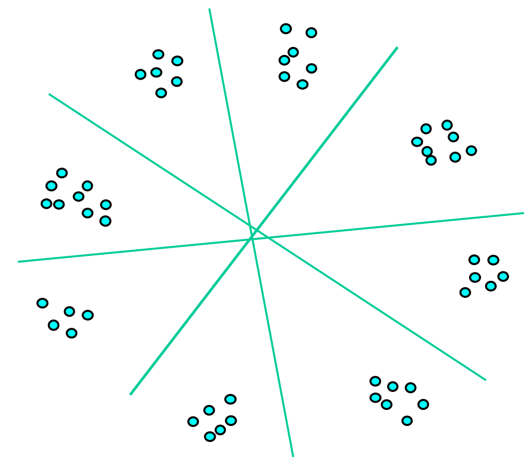
Transductive Support Vector Machines

Helpful distribution



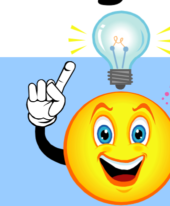
Non-helpful distributions

$1/\gamma^2$ clusters, all partitions separable by large margin



Semi-Supervised Learning

Prominent paradigm in past 15 years in Machine Learning.



Key Insight

Unlabeled data useful if we have beliefs not only about the form of the target, but also about its relationship with the underlying distribution.

Prominent techniques

- Transductive SVM [Joachims '99]
- Co-training [Blum & Mitchell '98]
- Graph-based methods [B&C01], [ZGL03]