

Now, minimize

$$C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\vec{w}\|^2, \text{ where } C > 0.$$

Since  $\xi_n > 1$  for every misclassified point,  $\sum_n \xi_n$  is an upper bound for the # of misclassified points.  $C$  controls the trade-off between the 2 terms above;  $C \rightarrow \infty$  is analogous to SVMs for separable data.

$$t_n y(\vec{x}_n) \geq 1 - \xi_n \quad \downarrow \quad \text{constraint}$$

Now, consider want to minimize

$$L(\vec{w}, b, \vec{a}) = \frac{1}{2} \|\vec{w}\|^2 + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N a_n [t_n y(\vec{x}_n) - 1 + \xi_n] - \sum_{n=1}^N \mu_n \xi_n$$

$\uparrow$   
 $\xi_n \geq 0$  constraint

$a_n \geq 0$  &  $\mu_n \geq 0$  are Lagrange multipliers.

KKT conditions:

$$\left\{ \begin{array}{l} a_n \geq 0, \\ t_n y(\vec{x}_n) - 1 + \xi_n \geq 0, \\ a_n (t_n y(\vec{x}_n) - 1 + \xi_n) = 0 \end{array} \right. \quad \left. \begin{array}{l} \mu_n \geq 0, \\ \xi_n \geq 0, \\ \mu_n \xi_n = 0 \end{array} \right\}$$

$n = 1, \dots, N$

Next,  $\left\{ \begin{array}{l} \frac{\partial L}{\partial \vec{w}} = 0 \Rightarrow \vec{w} = \sum_{n=1}^N a_n t_n \vec{g}(\vec{x}_n), \\ \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{n=1}^N a_n t_n = 0, \\ \frac{\partial L}{\partial \xi_n} = 0 \Rightarrow a_n = C - \mu_n. \end{array} \right.$

Then

$$L \rightarrow \underbrace{\tilde{L}(\vec{a})}_{\text{dual representation, need to maximize}} = \frac{1}{2} \sum_{n,m} a_n t_n a_m t_m \underbrace{\tilde{y}^T(\vec{x}_n) \cdot \tilde{y}(\vec{x}_m)}_{k(\vec{x}_n, \vec{x}_m)} +$$

$$\begin{aligned} & \oplus \sum_n a_n \xi_n + \sum_n a_n - \sum_n a_n \xi_n - \sum_{n,m} a_n t_n a_m t_m \tilde{y}^T(\vec{x}_n) \cdot \tilde{y}(\vec{x}_m) \\ & - b \sum_n a_n t_n = \sum_n a_n - \frac{1}{2} \sum_{n,m} a_n a_m t_n t_m k(\vec{x}_n, \vec{x}_m), \end{aligned}$$

subject to  $\sum_n a_n t_n = 0,$

$$a_n \geq 0 \quad \& \quad a_n = \underbrace{C}_{>0} - \underbrace{\mu_n}_{\geq 0} \geq C \Rightarrow \underbrace{0 \leq a_n \leq C}_{\text{box constraints}}$$

Predictions for new datapoints are made using  $y(\vec{x}) = \sum_{n=1}^N a_n t_n k(\vec{x}, \vec{x}_n) + b. \quad (*)$

For many datapoints,  $a_n = 0$  & those points do not affect (\*). Remaining points are support vectors:  $a_n > 0 \Rightarrow t_n y(\vec{x}_n) = 1 - \xi_n.$

If  $a_n < C \Rightarrow \mu_n > 0 \Rightarrow \xi_n = 0$  from  $\mu_n \xi_n = 0.$   
 $\uparrow$  these points lie on the margin

If  $a_n = C \Rightarrow \mu_n = 0 \Rightarrow \xi_n \geq 0$ , can lie inside the margin & will be correctly classified if  $\xi_n \leq 1$  & misclassified if  $\xi_n > 1.$

To determine  $b$ , use points with  $0 < a_n < C \Rightarrow t_n = 0 \Rightarrow t_n y(\vec{x}_n) = 1$ , or

$$t_n \left( \sum_{m \in S} a_m t_m k(\vec{x}_n, \vec{x}_m) + b \right) = 1.$$

$\uparrow$   
set of support vectors

Average over all points with  $0 < a_n < C$  for numerical stability:

$$b = \frac{1}{t_n} - \sum_{m \in S} a_m t_m k(\vec{x}_n, \vec{x}_m), \text{ or}$$

$\underbrace{t_n}_{t_n \in \{-1, 1\}}$

any  $n \in M$

$\underbrace{\text{set of indices of points with } 0 < a_n < C, \text{ of size } N_M}$

$$b = \frac{1}{N_M} \sum_{n \in M} \left[ t_n - \sum_{m \in S} a_m t_m k(\vec{x}_n, \vec{x}_m) \right].$$

Finally, what about finding  $a_n$ 's?

We need to maximize  $\bar{L}(\vec{a})$ , a quadratic f'n of  $a_n$ 's, subject to linear constraints.

This problem is known to produce a convex region with a unique (global) ~~maximum~~ maximum. A straightforward solution is often too costly  $\Rightarrow$  there're efficient algorithms that update Lagrange multipliers a subset at a time (rather than all at once).

## Relation to logistic regression

Recall that  $y_n t_n \geq 1$  (&  $\xi_n = 0$ ) on the correct side of the margin boundary, and for the remaining points  $\xi_n = 1 - y_n t_n$ .

Then  $C \sum_n \xi_n + \frac{1}{2} \|\vec{w}\|^2 \Rightarrow \sum_n [1 - y_n t_n]_+ + \lambda \|\vec{w}\|^2$ ,  
minimizing where  $\lambda = \frac{1}{2C}$  &  $[z]_+ = \begin{cases} z, & z \geq 0 \\ 0, & z < 0. \end{cases}$  (\*\*)

---

In logistic regression, we need to switch from  $t \in \{0, 1\}$  to  $t \in \{-1, 1\}$ .

Consider

$$p(t=1|y) = \sigma(y), \text{ then}$$

$$p(t=-1|y) = 1 - \sigma(y) = \sigma(-y), \text{ s.t.}$$

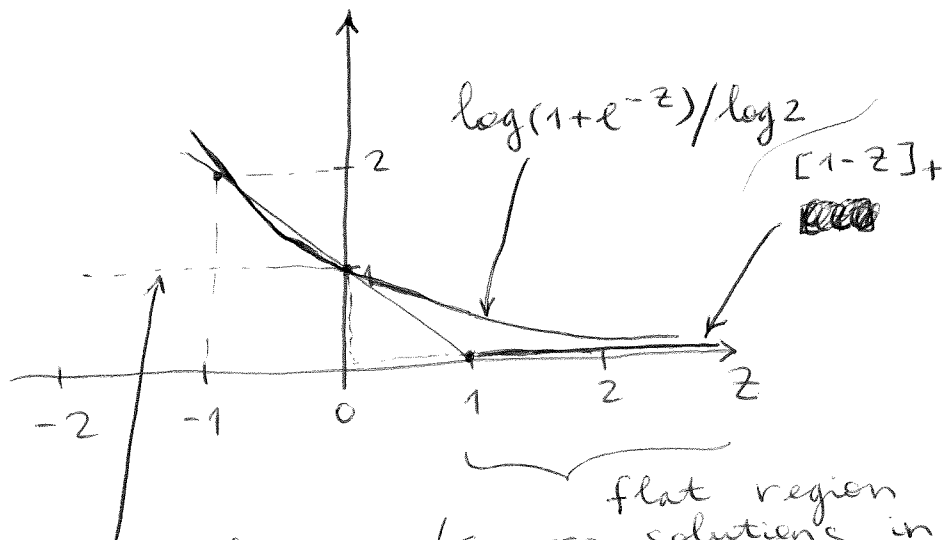
in general  $p(t|y) = \sigma(ty)$ .

Then  $\mathcal{L} = \prod_n \underbrace{p(t_n|y_n)}_{\sigma(t_n y_n)}$ , and

$$-\log \mathcal{L} = -\sum_n \log \sigma(t_n y_n) = \sum_n \log(1 + e^{-y_n t_n}).$$

With a quadratic regularizer, we need to minimize

$$\sum_n \log(1 + e^{-y_n t_n}) + \lambda \|\vec{w}\|^2, \text{ similar to (**)}$$



exact misclassification error:  $\begin{cases} 0 & \text{for } z \geq 0, \\ 1 & \text{for } z < 0. \end{cases}$  sparse solutions in SVMs

## SVMs for regression

Usually, we minimize

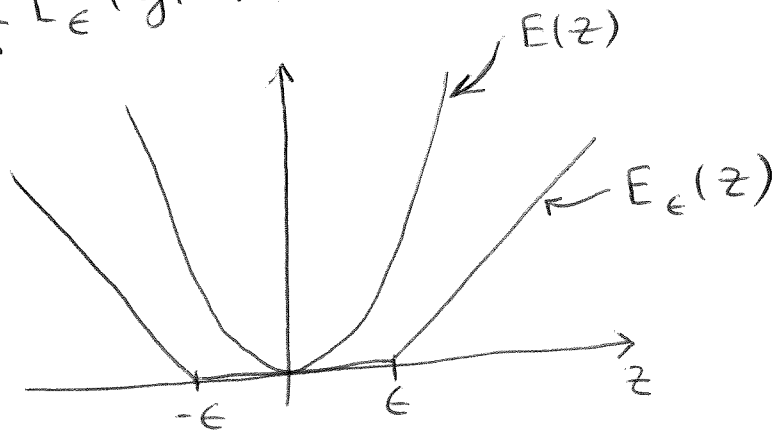
$$\frac{1}{2} \sum_n (y_n - t_n)^2 + \frac{\lambda}{2} \|\vec{w}\|^2$$

Idea: Replace the 1st term by an  $\epsilon$ -insensitive error  $f'$ 'n:

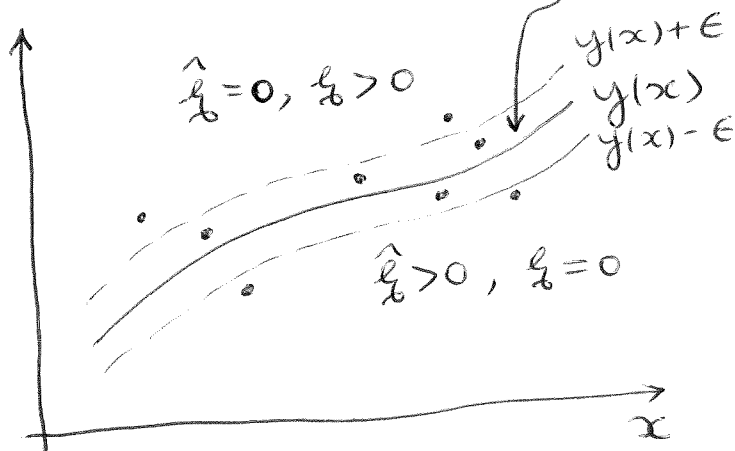
$$E_\epsilon(y(\vec{x}) - t) = \begin{cases} 0, & |y(\vec{x}) - t| < \epsilon, \\ |y(\vec{x}) - t| - \epsilon, & \text{otherwise.} \end{cases}$$

So, minimize  $\sum_n y_n$

$$C \sum_n E_\epsilon(y(\vec{x}_n) - t_n) + \frac{1}{2} \|\vec{w}\|^2$$



Introduce  $\xi_n \geq 0, \hat{\xi}_n \geq 0$   
 $\epsilon$ -tube:  $\xi = 0, \hat{\xi} = 0$



Inside the tube:  $y_n - \epsilon \leq t_n \leq y_n + \epsilon$ ,  
 above the tube:  $t_n \leq y_n + \epsilon + \xi_n$ , (\*\*\*)  
 Below the tube:  $t_n \geq y_n - \epsilon - \hat{\xi}_n$ .

Then, minimize  $C \sum_n (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|\vec{w}\|^2$   
 subject to (\*\*\*) above and  $\xi_n \geq 0, \hat{\xi}_n \geq 0$ .

Consider  

$$\mathcal{J} = C \sum_n (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|\vec{w}\|^2 - \sum_n (\mu_n \xi_n + \hat{\mu}_n \hat{\xi}_n) -$$

$$- \sum_n a_n (\underbrace{y_n - t_n + \epsilon + \xi_n}_{\geq 0 \text{ acc. to (***)}}) - \sum_n \hat{a}_n (\underbrace{t_n - y_n + \epsilon + \hat{\xi}_n}_{\geq 0 \text{ acc. to (***)}}),$$
 where  $a_n \geq 0, \hat{a}_n \geq 0, \mu_n \geq 0, \hat{\mu}_n \geq 0$  are

Lagrange multipliers.

Then

$$\begin{cases} \frac{\partial \mathcal{J}}{\partial \vec{w}} = 0 \Rightarrow \vec{w} = \sum_n (a_n - \hat{a}_n) \vec{\xi}(\vec{x}_n), \\ \frac{\partial \mathcal{J}}{\partial b} = 0 \Rightarrow \sum_n (a_n - \hat{a}_n) = 0, \\ \frac{\partial \mathcal{J}}{\partial \xi_n} = 0 \Rightarrow a_n + \mu_n = C, \\ \frac{\partial \mathcal{J}}{\partial \hat{\xi}_n} = 0 \Rightarrow \hat{a}_n + \hat{\mu}_n = C. \end{cases}$$

Now,  $\mathcal{J} \Rightarrow \underbrace{\tilde{\mathcal{J}}(\vec{a}, \hat{\vec{a}})}_{\text{to be maximized}} = C \sum_n (\xi_n + \hat{\xi}_n) +$   
 $+ \frac{1}{2} \sum_{n,m} (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\vec{x}_n, \vec{x}_m) -$   
 $- \sum_n [(C - a_n)\xi_n + (C - \hat{a}_n)\hat{\xi}_n] - \sum_n \underbrace{a_n(\epsilon + \xi_n)} -$   
 $- \sum_n \underbrace{\hat{a}_n(\epsilon + \hat{\xi}_n)} + \sum_n \underline{a_n t_n} - \sum_n \underline{\hat{a}_n t_n} - \sum_n (a_n - \hat{a}_n) y_n =$   
 $= \frac{1}{2} \sum_{n,m} (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\vec{x}_n, \vec{x}_m) - \epsilon \sum_n (a_n + \hat{a}_n) +$   
 $+ \sum_n (a_n - \hat{a}_n) t_n - \sum_{n,m} (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\vec{x}_n, \vec{x}_m) =$   
 $= -\frac{1}{2} \sum_{n,m} (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\vec{x}_n, \vec{x}_m) - \epsilon \sum_n (a_n + \hat{a}_n) +$   
 $+ \sum_n (a_n - \hat{a}_n) t_n$ , with the following constraints:

$$\underbrace{a_n \geq 0, \hat{a}_n \geq 0}_{\text{Lagrange multipliers}} ; \begin{cases} \mu_n \geq 0 \\ \hat{\mu}_n \geq 0 \end{cases} \Rightarrow \begin{cases} a_n \leq C \\ \hat{a}_n \leq C \end{cases}$$

So, we have box constraints:  $\begin{cases} 0 \leq a_n \leq C, \\ 0 \leq \hat{a}_n \leq C \end{cases}$   
 and  $\sum_n (a_n - \hat{a}_n) = 0$

Predictions for new inputs:

$$y(\vec{x}) = \sum_n (a_n - \hat{a}_n) k(\vec{x}, \vec{x}_n) + b \quad (*)$$

KKT conditions:  $\begin{cases} a_n(\epsilon + \xi_n + y_n - t_n) = 0, \\ \hat{a}_n(\epsilon + \hat{\xi}_n - y_n + t_n) = 0, \\ \mu_n \xi_n = (C - a_n) \xi_n = 0, \\ \hat{\mu}_n \hat{\xi}_n = (C - \hat{a}_n) \hat{\xi}_n = 0 \end{cases}$

each factor  $\Rightarrow$   
 in each product  $\Rightarrow$   
 $\leq \geq 0$

So, if  $a_n > 0 \Rightarrow \epsilon + \xi_n + y_n - t_n = 0 \Rightarrow$  data point is either on

the upper boundary of the  $\epsilon$ -tube:  $\begin{cases} \xi_n = 0, \\ t_n = y_n + \epsilon \end{cases}$

or above the upper boundary:  $\begin{cases} \xi_n > 0, \\ t_n = y_n + \epsilon + \underbrace{\xi_n}_{>0} \end{cases}$

Likewise, if  $\hat{a}_n > 0 \Rightarrow \epsilon + \hat{\xi}_n - y_n + t_n = 0$ , datapoint is either on ( $\hat{\xi}_n = 0$ ) or below ( $\hat{\xi}_n > 0$ ) the lower boundary of the  $\epsilon$ -tube.

Further,  $\begin{cases} \epsilon + \xi_n + y_n - t_n = 0, \\ \epsilon + \hat{\xi}_n - y_n + t_n = 0 \end{cases}$  are incompatible:  
 $\underbrace{2\epsilon}_{>0} + \underbrace{\xi_n + \hat{\xi}_n}_{>0} = 0$  does not work

So, for each  $\vec{x}_n$ ,  $a_n = 0$ , or  $\hat{a}_n = 0$ , or

$\underbrace{a_n = \hat{a}_n = 0}_{\text{inside the tube}}$

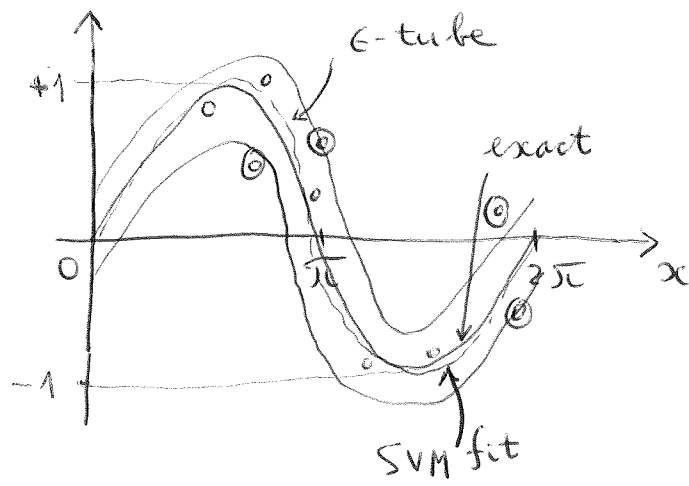
Only points outside of the tube contribute in (\*)  $\Rightarrow$  sparse solution.

We can find  $b$  by considering e.g. a point with  $0 < a_n < C \Rightarrow \xi_n = 0 \Rightarrow \epsilon + y_n - t_n = 0$  (upper tube boundary)

$$\begin{aligned} \text{Then } b &= t_n - \epsilon - \vec{w}^T \cdot \vec{g}(\vec{x}_n) = \\ &= t_n - \epsilon - \sum_m (a_m - \hat{a}_m) k(\vec{x}_n, \vec{x}_m). \end{aligned}$$

We can average over all points on the upper boundary; points on the lower boundary work just as well.  
 $\uparrow 0 < \hat{a}_n < C, a_n = 0$





⊙ are support vectors

Gaussian kernel