Support Vector Machines

Máster en Data Science y Big Data en Finanzas

José R. Dorronsoro

Revisiting the Classification Problem

- Basic problem: binary classification of a sample

$$S = \{(x^p, y^p), 1 \leq p \leq N\}$$

with d–dimensional x^p patterns and $y^p = \pm 1$

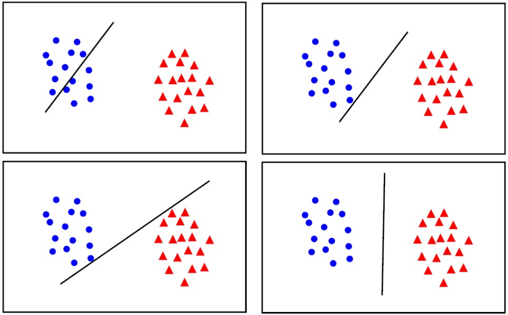
- We assume that S is linearly separable: for some w, b

$$\begin{aligned} w \cdot x^p + b &> 0 \quad \text{if } y^p = 1; \\ w \cdot x^p + b &< 0 \quad \text{if } y^p = -1 \end{aligned}$$

- More concisely, we want $y^p(w \cdot x^p + b) > 0$
- Q: How can we find a pair w, b so that the model generalizes well?

Which Hyperplane is Best?

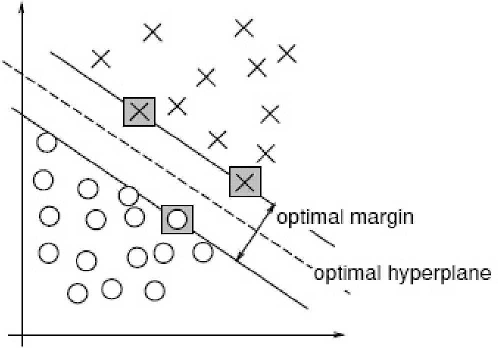
- Of the three separating hyperplanes, the lower right one is intuitively the best



From A. Zisserman, C19 Machine Learning, Oxford University

- Q: How can we characterize it?

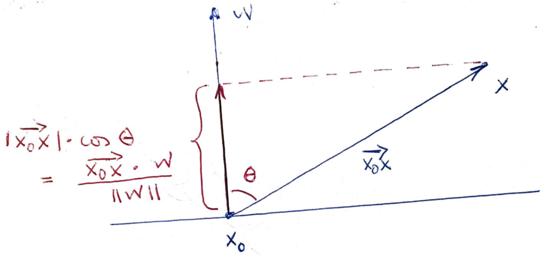
- A: Intuitively, we want (w, b) to have a large **margin**



- Q: How can we ensure a maximum margin?

Distance to a Line

- Recall basic analytic geometry



- This extends to the multidimensional case

Distance to a Hyperplane

- Recall that given the hyperplane $\pi : w \cdot x + b = 0$, w is orthogonal to the surface defined by π
- If $x_0 \in \pi$, we compute the distance $d(x, \pi)$ of a point x to π projecting on w the vector $\overrightarrow{x_0 x}$, i.e.

$$d(x, \pi) = \frac{|w \cdot \overrightarrow{x_0 x}|}{\|w\|} = \frac{|w \cdot x - w \cdot x_0|}{\|w\|} = \frac{|w \cdot x + b|}{\|w\|}$$

  for $w \cdot x_0 + b = 0$; i.e. $w \cdot x_0 = -b$
- The absolute values compensate for the orientation of w
- When the origin is in π (homogeneous π), the distance is

$$d(x, \pi) = \frac{|w \cdot x|}{\|w\|}$$

- If we assume w “points" to the positive patterns, we have $y^p(w \cdot x^p + b) = |w \cdot x^p + b|$
- The **margin** $\gamma = \gamma(w)$ is precisely the **minimum distance** between the sample S and π, i.e.,

$$\gamma = m(w, b, S) = \min_p d(x^p, \pi) = \min_p \frac{y^p(w \cdot x^p + b)}{\|w\|}$$

- Notice that $(\lambda w, \lambda b)$ give the same margin than (w, b); we can thus normalize (w, b) as we see fit
- For instance, taking $\|w\| = 1$ we have

$$\gamma(w) = \min_p \frac{y^p(w \cdot x^p + b)}{\|w\|} = \min_p y^p(w \cdot x^p + b)$$

- But we will work with the following normalization of w, b

$$\min_p y^p(w \cdot x^p + b) = 1$$

- Since S is finite, we will have $y^{p_0}(w \cdot x^{p_0} + b) = 1$ for some p_0
- For a pair w, b so normalized we then have

$$m(w,b) = \min_p \left\{ \frac{y^p(w \cdot x^p + b)}{\|w\|} \right\} = \frac{y^{p_0}(w \cdot x^{p_0} + b)}{\|w\|} = \frac{1}{\|w\|}$$

- Thus, we maximize the overall margin working with these w and maximizing $1/\|w\|$, i.e., **minimizing** $\|w\|$ or, simply, minimizing $\frac{1}{2}\|w\|^2$

The Primal Problem

- We therefore rewrite the problem of finding a maximum margin separating hyperplane as

$$\min_{w,b} f(w,b) = \frac{1}{2}\|w\|^2$$

s.t. $y^p(w \cdot x^p + b) \geq 1$

- This is the **SVM Primal Problem**: a quadratic programming problem with linear restrictions (actually affine)
- The function to minimize is very simple and also the constraints but there are too many of them for a direct attempt to minimization
- Solution within general theory of convex minimization

The Lagrangian

- For $\alpha_p \geq 0$, the Lagrangian of the primal problem is

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_p \alpha_p \left(y^p (w \cdot x^p + b) - 1 \right),$$

- Clearly, $L(w, b, \alpha) \leq f(w, b)$ and $L(w, b, 0) = f(w, b)$
- Thus, for feasible w, b, α,

$$\min_{w,b \text{ feasible}} f(w, b) = \min_{w,b \text{ feasible}} \max_{\alpha \text{ feasible}} L(w, b, \alpha)$$

- Q: perhaps it holds that

$$\min_{w,b \text{ feasible}} \max_{\alpha \text{ feasible}} L(w, b, \alpha) = \max_{\alpha \text{ feasible}} \min_{w,b \text{ feasible}} L(w, b, \alpha)$$

- To explore this we will define the **dual** function $\Theta(\alpha) = \min_{w,b} L(w, b, \alpha)$
  - Notice that we drop the requirement that w, b be feasible

- The **dual problem** D is now

$$\max \Theta(\alpha) \text{ s. t. } \alpha_p \geq 0$$

- Now we have for any feasible w, b, α

$$\Theta(\alpha) = \min_{w', b'} L(w', b', \alpha) \leq L(w, b, \alpha) \leq f(w, b)$$

- **Weak duality**: for primal optimal w^*, b^*, dual optimal α^* and any feasible w, b, α,

$$\Theta(\alpha) \leq \Theta(\alpha^*) \leq L(w^*, b^*, \alpha^*) \leq f(w^*, b^*) \leq f(w, b)$$

- **Dual gap** at feasible w, b, α: $f(w, b) - \Theta(\alpha) \geq 0$

Strong Duality

- We achieve **strong duality** if the dual gap at optima w^*, b^*, α^* is 0, that is,

$$f(w^*, b^*) = \Theta(\alpha^*)$$

- Moreover $\Theta(\alpha^*) = L(w^*, b^*, \alpha^*) = f(w^*, b^*)$
- **Theorem**: *The SVM problem has strong duality*
- Thus, to solve the SVM problem, we can try the following:
  - Write an explicit dual problem with easier constraints
  - Solve the dual problem
  - Get the optimal primals w^*, b^* from the optimal dual α^*

Computing the Dual Function

- We follow the previous program and try first to write down $\Theta(\alpha) = \min_{w,b} L(w, b, \alpha)$
- We first reorganize the (convex) Lagrangian as

$$L(w, b, \alpha) = w \cdot \left(\frac{1}{2} w - \sum_p \alpha_p y^p x^p \right) - b \sum_p \alpha_p y^p + \sum_p \alpha_p$$

- To minimize $L(w, b, \alpha)$ w.r. w and b, we just solve $\nabla_w L = 0$, $\frac{\partial L}{\partial b} = 0$
- From $\nabla_w L = 0$ we derive $w = \sum_p \alpha_p y^p x^p$
- From $\frac{\partial L}{\partial b} = 0$ we derive $\sum_p \alpha_p y^p = 0$

Computing the Dual Function II

- Substituting both into L we arrive at

$$\begin{aligned}\Theta(\alpha) &= \sum_p \alpha_p - \frac{1}{2} w \cdot \sum_p \alpha_p y^p x^p \\ &= \sum_p \alpha_p - \frac{1}{2} \sum_{p,q} \alpha_p \alpha_q y^p y^q \ x^p \cdot x^q = \sum_p \alpha_p - \frac{1}{2} \alpha^\tau Q \alpha\end{aligned}$$

  with $Q_{p,q} = y^p y^q \ \ x^p \cdot x^q$

- The dual problem becomes

$$\max_\alpha \ \Theta(\alpha) = \max_\alpha \left\{ \sum_p \alpha_p - \frac{1}{2} \alpha^\tau Q \alpha \right\}$$

  subject to the constraints $\alpha_p \geq 0$, $\sum_p \alpha_p y^p = 0$

- As usual, we will minimize $-\Theta(\alpha)$ (and drop the $-$ from the notation)

Solving the Dual Problem

- We arrive again at a quadratic programming problem but with much simpler restrictions that we can try to simplify further
- The more difficult constraint $\sum_p \alpha_p y^p = 0$ comes from $\frac{\partial L}{\partial b} = 0$ and we could avoid it dropping b
- Thus, we try first to solve the **homogeneous** primal problem

$$\min \frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y^p \; w \cdot x^p \geq 1$$

and its dual one

$$\min \frac{1}{2}\alpha^\tau Q\alpha - \sum_p \alpha_p \quad \text{s.t.} \quad \alpha_p \geq 0$$

Projected Gradient Descent

- We can solve the homogeneous dual by **projected gradient descent**
- The gradient of Θ is just

$$\nabla\Theta = Q\alpha - \mathbf{1}$$

  with **1** the all ones vector and we can solve it by **projected gradient descent**
- Projected (i.e., clipped) descent:
  - At step t update first α^t to α' as $\alpha'_p = \alpha^t_p - \rho\left((Q\alpha^t)_p - 1\right)$ for an appropriate step ρ
  - And then clip α' as $\alpha^{t+1}_p = \max\{\alpha'_p, 0\}$
- Nice and fine, but notice that dim$(\alpha) = N$:
  - Computations have a cost of $O(N^2)$ per iteration
  - We need to keep Q in memory, which has dimension $N \times N$
  - Both too costly for large N

- Usually homogeneous SVMs give poorer results
- The simplest way to handle the equality constraint is
  - Start with an α^0 that verifies it
  - Update α^t to $\alpha^{t+1} = \alpha^t + \rho_t d^t$ with a direction d^t that also verifies it
  - Then $\sum_p \alpha_p^{t+1} y^p = \sum_p \alpha_p^t y^p + \rho_t \sum_p d_p^t y^p = 0$
- Simplest choice: select L_t, U_t so that $d^t = y^{L_t} e_{L_t} - y^{U_t} e_{U_t}$ is a maximal **descent direction**
- Since $\nabla_\alpha \Theta(\alpha^t) \cdot d^t = y^{L_t} \nabla\Theta(\alpha^t)_{L_t} - y^{U_t} \nabla\Theta(\alpha^t)_{U_t}$, the straightforward choice is

$$L_t = \arg\min_p y^p \nabla\Theta(\alpha^t)_p, \quad U_t = \arg\min_q y^q \nabla\Theta(\alpha^t)_q$$

- This is the basis of the **Sequential Minimal Optimization** (SMO) algorithm

- Since L is convex in w, b and we have

$$\Theta(\alpha^*) = \min_{w,b} L(w, b, \alpha^*)$$

stationarity is necessary:

$$\nabla_w L(w^*, b^*, \alpha^*) = 0, \; \frac{\partial L}{\partial b}(w^*, b^*, \alpha^*) = 0$$

- By strong duality, $L(w^*, b^*, \alpha^*) = f(w^*, b^*)$ and, for all p, **complementary slackness** follows

$$\alpha_p^* \left(y^p (w^* \cdot x^p + b^*) - 1 \right) = 0$$

- These two plus feasibility are together known as the **Karush–Kuhn–Tucker (KKT)** conditions, that are necessary and sufficient for w^*, b^*, α^* to be optimal

From Dual Solutions to Primal Solutions I

- We will use some of the KKT conditions to derive the optimal primal w^*, b^* after we obtain a dual optimal α^*
- Obvioulsy $w^* = \sum_p \alpha_p^* y^p x^p = \sum_{\alpha_p^* > 0} \alpha_p^* y^p x^p$
- What about b^*? Recall that the optimal α^*, w^*, b^* must satisfy the KKT conditions, that now are

$$\alpha_p^* \left(y^p (w^* \cdot x^p + b^*) - 1 \right) = 0$$

- Thus, if $\alpha_p^* > 0$, then $w^* \cdot x^p + b^* = y^p$ and, hence

$$b^* = y^p - w^* \cdot x^p$$

From Dual Solutions to Primal Solutions II

- In practice is better to average this formula over all $\alpha_p^* > 0$:

  $$b^* = \frac{1}{N_S} \sum_{\{\alpha_q^* > 0\}} (y^q - w^* \cdot x^q)$$

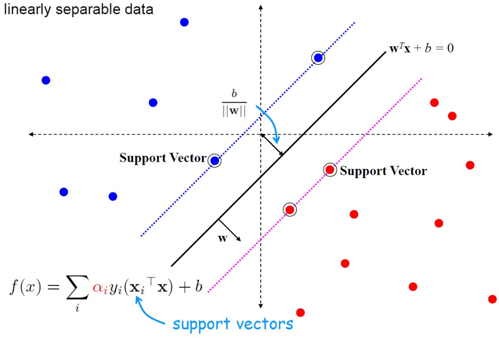
  with $N_S = |\{q : \alpha_q^* > 0\}|$
- We have now completely solved the **linear SVM problem** for classification
- But there are more insights to be gained from the convex optimization perspective
- In particular, the KKT conditions have more information

- Again, if $\alpha_p^* > 0$, then $y^p(w^* \cdot x^p + b^*) = 1$
  - Thus if $\alpha_p^* > 0$, x^p lies in one of the two **support hyperplanes** $w^* \cdot x^p + b^* = \pm 1$
- Vectors for which $\alpha_p^* > 0$ are thus called **support vectors** and the optimal w^* is a **linear combination** of them

$$w^* = \sum_{\{x^p \ SV\}} \alpha_p^* y^p x^p$$

- On the other hand, if x^p is not in a support hyperplane, then $y^p(w^* \cdot x^p + b^*) > 1$ and the KKT conditions imply $\alpha_p^* = 0$
- Notice that there may be x^p in the support hyperplanes that do not contribute to w^*

- In fact, while the optimal w^* is unique, the optimal α^* may be not
- In any case, the support vectors completely determine the SVM classifier



From A. Zisserman, C19 Machine Learning, Oxford University

- Maximum margins (MM) improve the generalization of linear classifiers
- To get a MM classifier we solve the primal problem

$$\min_{w,b} \frac{1}{2}\|w\|^2 \text{ s.t. } y^p(w \cdot x^p + b) \geq 1, 1 \leq p \leq N$$

- This is a convex quadratic programming problem whose Lagrangian for $\alpha_p \geq 0$ is

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_p \alpha_p \left(y^p(w \cdot x^p + b) - 1\right)$$

- If $\mathcal{C} = \{\alpha : \alpha_p \geq 0, \sum \alpha_p y^p = 0\}$, the dual problem is

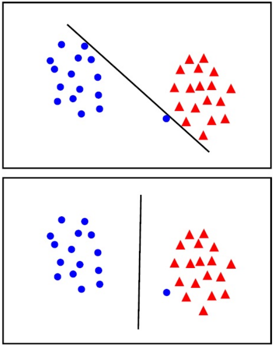
$$\min_{\alpha_p \in \mathcal{C}} \Theta(\alpha) = \frac{1}{2}\alpha^\tau Q \alpha - \sum_p \alpha_p$$

Takeaways on Linear SVMs II

- The dual gap $f(w^*, b^*) - \Theta(\alpha^*)$ at optima is 0 and so we can
  - Obtain the optimal dual α^* and then
  - Derive from α^* the optimal primal w^*, b^*
- We solve the dual problem using the **SMO algorithm**, with a cost at least $\Omega(N^2)$
- The KKT conditions are used to obtain w^* and b^*
- For the optimal w^* we have $w^* = \sum_{SV} \alpha_p^* y^p x^p$
- For the optimal b^* we have $b^* = y^p - w^* \cdot x^p$ if $\alpha^* > 0$
- If $\alpha^* > 0$, $w^* \cdot x^p + b^* = y^p$, i.e., x^p is in one of the **support hyperplanes** $w^* \cdot x + b^* = \pm 1$

Linear Is Not Always Best

- Going for linear is not always the best option



From A. Zisserman, C19 Machine Learning, Oxford University

- Besides, linear problems are not very frequent

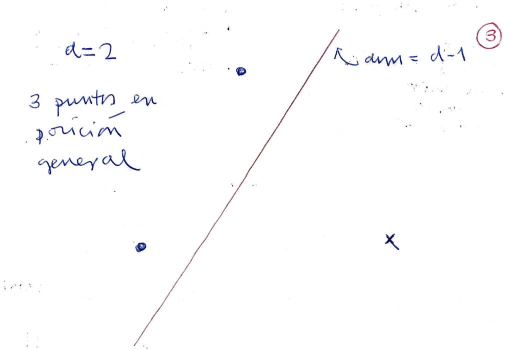
Cover's Theorem

- SVMs are simple and elegant, but also linear
- Q: Will linear SVM classifiers powerful enough?
- Alternative Q: Are linearly solvable classification problems frequent enough?
- A: No, because of **Cover's Theorem**
- The patterns in a size N sample S with dimension d are said to be in **general position** if no $d+1$ points are in a $(d-1)$–dimensional hyperplane
- Then, *if* $N \leq d+1$*, all 2–class problems on* S *are linearly separable and if* $N > d+1$*, the number of linearly separable problems is*

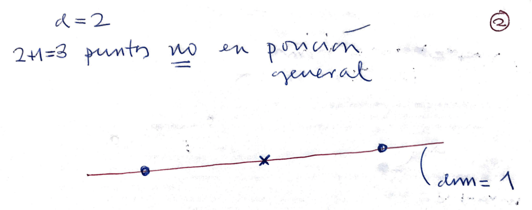
$$2\sum_{i=0}^{d}\binom{N-1}{i}$$

Points in General Position

- Consider $d = 2$, $3 = d + 1$ points and a $1 = d - 1$-dimensional hyperplane



- Consider now $d = 2$ and $3 = d + 1$ points **not** on a $1 = d - 1$-dimensional hyperplane (i.e., a line)



Are Linearly Separable Problems Frequent?

- Our current SVM classifiers will be useful if linearly separable 2–class problems are frequent enough
- It is relatively easy to show that for $N \gg d+1$

$$2\sum_{i=0}^{d}\binom{N-1}{i} \leq 2(d+1)\binom{N-1}{d} \leq 2\frac{d+1}{d!}N^d \lesssim N^d$$

- On the other hand, the **total number of two–class problems** over a sample of size N is 2^N
- And $\frac{N^d}{2^N} \to 0$ very fast when $N \to \infty$
- Since in many practical problems we will have $N \gg d$, **essentially all such 2–class problems won't be linearly separable**
- And our current SVMs will be useless on them

- Q: What can we do?
- First step: make room for non linearly separable problems
- We no longer require perfect classification but **allow for error (slacks) in some patterns**
- We relax the previous requirement $y^p(w \cdot x^p + b) \geq 1$ to

$$y^p(w \cdot x^p + b) \geq 1 - \xi_p$$

where we impose a new constraint $\xi_p \geq 0$
- Notice that if $\xi_p \geq 1$, x^p will not be correctly classified
- Thus, we allow for defective classification but we also **penalize** it

L_k Penalty SVMs

- New primal problem: for $K \geq 1$ consider the cost function

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + \frac{C}{K} \sum \xi_p^K$$

now subject to $y^p(w \cdot x^p + b) \geq 1 - \xi_p$, $\xi_p \geq 0$

- Simplest choice $K = 2$: L_2 (i.e., square penalty) SVMs
  - It can be seen to reduce to the previous set up
- Usual (and best) choice $K = 1$
  - We will concentrate on it
- Notice that if $C \to \infty$ we recover the previous slack-free approach

- Primal problem

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum \xi_p$$

subject to $y^p(w \cdot x^p + b) \geq 1 - \xi_p$, $\xi_p \geq 0$

- The L_1 Lagrangian is then

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + C\sum \xi_p - \\ \sum \alpha_p \left[y^p(w \cdot x^p + b) - 1 + \xi_p\right] - \sum \beta_p \xi_p$$

with $\alpha_p, \beta_p \geq 0$

L_1 SVM Lagrangian

- Again we reorganize the L_1 Lagrangian as

$$
\begin{aligned}
L(w, b, \xi, \alpha, \beta) \quad = \quad & w \cdot \left(\frac{1}{2} w - \sum \alpha_p y^p \, x^p \right) + \\
& \sum \xi_p (C - \alpha_p - \beta_p) - b \sum \alpha_p y^p + \\
& \sum \alpha_p
\end{aligned}
$$

- The w and b partials yield as before $w = \sum \alpha_p y^p x^p$, $\sum \alpha_p y^p = 0$

- From $\frac{\partial L}{\partial \xi_p} = C - \alpha_p - \beta_p = 0$ we see that

$$C = \alpha_p + \beta_p,$$

- Substituting things back into the Lagrangian we arrive at the L_1 dual function

$$\begin{aligned} \Theta(\alpha, \beta) &= \sum_p \alpha_p - \frac{1}{2} w \cdot \sum \alpha_p y^p x^p \\ &= \sum_p \alpha_p - \frac{1}{2} \alpha^\tau Q \alpha \end{aligned}$$

subject to $\sum_p \alpha_p y^p = 0, \alpha_p + \beta_p = C$, plus $\alpha_p \geq 0, \beta_p \geq 0$

The L_1 SVM Dual II

- In fact, we can drop β
  - Notice that we already have that $\Theta(\alpha, \beta) = \Theta(\alpha)$
  - It is also clear that the constraints on α, β can be reduced to $0 \leq \alpha_p \leq C$
- Thus, we get essentially the same dual problem as before

$$\min_{\alpha} \ \frac{1}{2}\alpha^{\tau} Q \alpha - \sum_p \alpha_p$$

subject to $\sum \alpha_p y^p = 0$, $0 \leq \alpha^p \leq C$, $1 \leq p \leq N$

  - Notice again that if $C \to \infty$ we recover the penalty free SVM
- We can solve it by SMO
- And here also $w^* = \sum \alpha_p^* y^p x^p$ for the optimal w^*

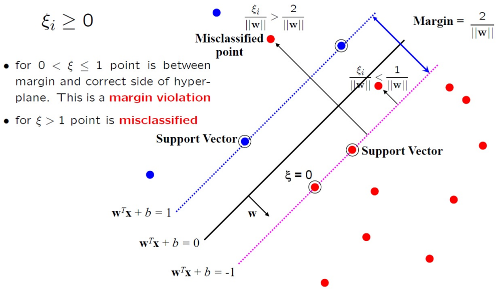
- The complementary slackness conditions are now

$$\begin{aligned} \alpha_p^* \left[y^p (w^* \cdot x^p + b^*) - 1 + \xi_p^* \right] &= 0 \\ \beta_p^* \xi_p^* &= 0 \end{aligned}$$

- Now, if $\xi_p^* > 0$, then $\beta_p^* = 0$ and, therefore, $\alpha_p^* = C$
  - We say that such an x^p is **at bound**
- Also, if $0 < \alpha_p^* < C$, then $\beta_p^* > 0$ and $\xi_p^* = 0$
  - Thus, if $0 < \alpha_p^* < C$, $y^p (w^* \cdot x^p + b^*) = 1$ and x^p lies in one of the support hyperplanes
  - We can obtain $b^* = y^p - w^* \cdot x^p$ just as before
  - If needed, we can then derive $\xi_p^* > 0$, since $\alpha_p^* = C$ and

$$\xi_p^* = 1 - y^p (w^* \cdot x^p + b^*)$$

- Slacks determine whether or not a pattern will be correctly classified



From A. Zisserman, C19 Machine Learning, Oxford University

SMO And Its Cost

- SMO can also be applied to L_1 SVMs straightforwardly; recall that
  - We start with $\alpha^0 = 0$ for which trivially $\sum y^p \alpha_p^0 = 0$
  - At step t select
    $L_t = \arg\min_p y^p \nabla\Theta(\alpha^t)_p, \quad U_t = \arg\min_q y^q \nabla\Theta(\alpha^t)_q$
  - Update $\alpha^{t+1} = \alpha^t + \rho_t d^t$ with $d^t = y^{L_t} e_{L_t} - y^{U_t} e_{U_t}$ and clip it if needed to have $0 \leq \alpha_{L_t}^{t+1}, \alpha_{U_t}^{t+1} \leq C$
  - And iterate until a KKT–related stopping condition is met
- Notice that two α values are updated on each iteration

- Thus, if the number of SVs is m, SMO requires at least $m/2$ iterations
- Since in general the number of SVs is $\Theta(N)$, the number of iterations is at least $\Theta(N)$
- And since each iteration has a $\Theta(N)$ cost, the cost of SMO is at least $\Theta(N^2)$
  - Very high!!! The NN cost was $\Theta(N)$
- Moreover, the cost of predicting on a new pattern is $m = \Theta(N)$
  - Also very high!!! The NN cost was $O(1)$
- SVMs give excellent models, usually hard to beat
- But cannot be used when sample size is above, say, 10^5 or when streaming exploitation is needed

Good Option, But ...

- L_1 SVMs are (relatively) **sparse**, i.e., the number of non–zero multipliers is $\ll N$
- The bound $\alpha_p^* = C$ for $\xi_p^* > 0$ limits the effect of not correctly classified patterns
- And usually L_1 SVMs are much better than, say, L_2 SVMs
- But still **they are linear ...**
- We must thus somehow **introduce some kind of non–linear processing for SVMs to be truly effective**

- Recall that the number $L(N, d)$ of linearly separable dichotomies is

$$L(N, d) = \begin{cases} 2^N & \text{if } N \leq d+1 \\ 2\sum_{i=0}^{d} \binom{N-1}{i} & \text{if } N \geq d+1 \end{cases}$$

- Recall that for d fixed, $\frac{L(N,d)}{2^N} \to 0$ as $N \to 0$
- In practice $N \gg d$ and the fraction of separable dichotomies will be very small
- But if we transform the initial patterns into new ones with dimension $D \gg N$, **all dichotomies will be linearly separable**

- Idea: (non linearly) augment pattern dimension going from $x \in \mathbf{R}^d$ to $\Phi(x) \in \mathbf{R}^D$ with $D \gg d$
- First option: do it explicitly as, for instance, in $\Phi(x) = (x_1, \ldots, x_i, \ldots, x_i x_j, \ldots, x_i x_j x_k, \ldots)$
- Too cumbersome, so try to do it **implicitly**
- Observation: in SVMs we only need to compute dot products $x \cdot x'$
  - And the same is true for the SMO algorithm
- Thus, we can work **implicitly** with extensions $\Phi(x)$ provided it is easy to compute $\Phi(x) \cdot \Phi(x')$
- Simplest case: $\Phi(x) \cdot \Phi(x') = k(x, x')$ for an appropriate **kernel** k

Polynomial Kernels

- A simple option is to work with **polynomial** kernels $k(x, x') = (1 + x \cdot x')^m$
- Assume $m = 2$, $x = (x_1, x_2)$, $x' = (x'_1, x'_2)$; then

$$\begin{aligned} k(x, x') &= (1 + x_1 x'_1 + x_2 x'_2)^2 \\ &= 1 + 2x_1 x'_1 + 2x_2 x'_2 + x_1^2 (x'_1)^2 + \\ &\quad x_2^2 (x'_2)^2 + 2x_1 x_2 x'_1 x'_2 \\ &= \Phi(x) \cdot \Phi(x') \end{aligned}$$

with

$$\Phi(x) = \Phi(x_1, x_2) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$$

- In fact, if the kernel is **positive definite** we can diagonalize it as

$$k(x, x') = \sum_{0}^{\infty} \lambda_k \varphi_k(x) \varphi_k(x')$$

with $\lambda_k > 0$ and the (possibly infinitely many) $\{\varphi_k(x)\}$ orthonormal

- Defining then

$$\Phi(x) = (\sqrt{\lambda_0}\varphi_0(x), \sqrt{\lambda_1}\varphi_1(x), \ldots ...)$$

we have $k(x, x') = \Phi(x) \cdot \Phi(x')$

- The dot product matrix Q is now the **kernel matrix** $Q_{p,q} = k(x^p, x^q)$

The Gaussian Kernel

- If we use the Gaussian kernel $k(x, x') = e^{-\gamma \|x - x'\|^2}$, $\Phi(x)$ has (theoretically) an infinite dimension
  - So Cover's theorem ensures that all samples will be linearly separable
  - And practical SVMs are (almost) always built using Gaussian kernels
  - Thus, overfitting is guaranteed unless we **renounce to perfect separability**
- Thus, we have to build effective SVMs using a powerful kernel but, also, avoiding overfit, by
  - Adequately adjusting the penalty constant C
  - And also the Gaussian kernel's width γ

Selecting the C Hyperparameter for SVMs

- C is actually a **regularization** parameter as it limits where we can find the optimal α
- Notice also that we can write the primal cost function as

$$\frac{1}{N}\sum \xi_p + \frac{1}{2}\frac{1}{CN}\|w\|^2$$

  - Thus $\frac{1}{CN}$ behaves similarly to α in Ridge or Logistic Regression
- From another point of view,
  - Small C allow large slacks and a possible underfit
  - But large C imply very small slacks and possible overfit
- One usually explores values $10^k, \quad -K_L \leq k \leq K_R$
  - Typical values are $K_L = -1, 0$, i.e., $C_L = 0.1$ or 1, and $K_R = 3$ or 4, i.e., $C_R = 1,000$ or $10,000$

Selecting the γ Hyperparameters for Gaussian SVMs

- When working with Gaussian kernels, the features x_i are usually scaled to a $[0, 1]$ range
- Then $|x_i - x'_i| \leq 1$ and if d is pattern dimension

$$\|x - x'\|^2 = \sum_1^d (x_i - x'_i)^2 \lesssim d \;\; \Rightarrow \;\; \frac{\|x - x'\|^2}{d} \lesssim 1$$

  - Then $e^{-\frac{\|x-x'\|^2}{d}}$ behaves approximately as $e^{-|z|}$
- This suggests to explore γ values of the form, for instance,

$$\frac{4^k}{d}, \;\; -K \leq k \leq K, \text{ i.e., } e^{-4^k|z|} = \left(e^{-|z|^2}\right)^{2^k}$$

- Large k values result in very sharp Gaussians
  - We may end up with a Gaussian for each sample x^p and, hence, overfit
- Small k values result in flat, nearly constant Gaussians
  - No x^p is relevant and, hence, underfit is quite likely

Linear Kernels?

- Recall that we use kernels to enlarge pattern dimension
  - We get better models but costlier training
  - And working with large datasets may become impractical
- We may try to avoid them if pattern dimension is already large and just use linear SVMs
- This is the approach followed by the LIBLINEAR package, which offers
  - Dual-based solvers using coordinate descent methods
  - Primal-based solvers using Newton–type methods
- The constant term b is usually not considered, so data should be centered before training
- Only C has to be hyperparameterized

Other Things

- SVMs do not have an underlying probability model
  - Label prediction is the primary output
- The LIBSVM and its Scikit–learn wrapper can give probability predictions using an ad–hoc model
- ν–SVMs (available in LIBSVM) can also be used for classification (and regression) usually with very similar results
- SVM classification is intrinsically two-class
- Multiclass problems are usually reduced to a number of 2-class problems under two strategies
  - One versus one (OVO)
  - One versus the rest (OVR)

Multiclass SVMs

- Assume the number of classes is C
- In OVO $\frac{C(C-1)}{2}$ 2-class problems are solved, one for each pair of classes
  - The sample sizes become $2\frac{N}{C}$
- The $\frac{C(C-1)}{2}$ models are applied on a new pattern and it is assigned to the class with more votes
- In OVR C 2-class problems are solved where each class is pitted against all others
  - The sample sizes remain N
- The C models are are applied on a new pattern and it is assigned to the class with the highest score

- The L_1 primal problem is

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum \xi_p$$

s.t. $y^p(w \cdot x^p + b) \geq 1 - \xi_p$, $\xi_p \geq 0$, $1 \leq p \leq N$

- For $\alpha_p, \beta_p \geq 0$ the new Lagrangian is

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + C\sum \xi_p - \sum_p \alpha_p \left(y^p(w \cdot x^p + b) - 1 + \xi_p\right) - \sum \beta_p \xi_p$$

- And for $\mathcal{C} = \{\alpha : 0 \leq \alpha_p \leq C, \sum \alpha_p y^p = 0\}$, the L_1 dual problem is

$$\max_{\alpha_p \in \mathcal{C}} \quad \Theta(\alpha) = \sum_p \alpha_p - \frac{1}{2}\alpha^\tau Q \alpha$$

Takeaways on Non Linear SVMs II

- The new dual coincides essentially with the linear dual and can also be solved by the SMO algorithm, with a cost $\Omega(N^2)$
- The KKT conditions are again used to obtain w^* and b^*
- For the optimal w^* we have $w^* = \sum_{x^p \in SV} \alpha_p^* y^p x^p$
- If $0 < \alpha_p^* < C$ we have $b^* = y^p - w^* \cdot x^p$
- And if $\xi_p^* > 0$, $\alpha_p^* = C$
- All the dot products can be replaced by kernel operations $k(x^p, x^q)$
- Two hyperparameters appear: the penalty C and (if used) the Gaussian kernel width γ

- The classification slack ξ of a pattern x, y can be written as

$$\xi = \max\{0, 1 - y(w \cdot x + b)\} = h(y(w \cdot x + b) - 1)$$

where $h(z) = \max\{0, -z\}$ is the **hinge loss**

- We can write the linear SVC primal problem as

$$\text{arg min}_{w,b} \quad \frac{1}{2}\|w\|^2 + C \sum_p h(y(w \cdot x + b) - 1) =$$

$$\text{arg min}_{w,b} \quad \frac{1}{N} \sum_p h(y(w \cdot x + b) - 1) + \frac{1}{2C\ N}\|w\|^2 \quad (1)$$

- The hinge loss is not differentiable only at $z = 0$
  - This is also the case of the ReLUs
- We need the dual problem to be able to use the kernel trick
  - But we could put the primal hinge loss at the end of a DNN

- In SV regression (SVR) we could try to solve another regularized problem

$$\min_{w,b} f(w,b) = \frac{1}{2}\|w\|^2 + C\sum_p \left[y^p - (w\cdot x^p + b)\right]_\epsilon$$

or, equivalently,

$$\min_{w,b} \frac{1}{N}\sum_p \left[y^p - (w\cdot x^p + b)\right]_\epsilon + \frac{1}{2}\frac{\lambda}{N}\|w\|^2$$

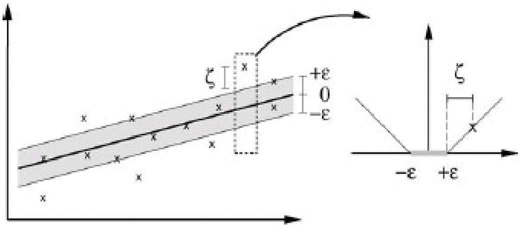
using the ϵ–**insensitive** loss

$$[z]_\epsilon = \max(0, |z| - \epsilon)$$

- Notice we penalize an error $|y^p - f(x^p, w, b)|$ only if it is $> \epsilon$

The ϵ Error Tube

- Therefore, we **do not penalize errors of predictions that fall inside an ϵ–wide tube around the true function**



SVR as a Constrained Problem

- We have $f(w,b) = \ell_\epsilon(w,b) + \frac{\lambda}{2}\|w\|^2$
  - f is convex but $\ell_\epsilon = \sum_p [y^p - (w \cdot x^p + b)]_\epsilon$ is not smooth
  - Direct minimization of $f(w,b)$ seems difficult
  - Thus, we recast the unconstrained SVR problem as a constrained one
- If $C = 1/\lambda$, we rewrite f as

$$f(w,b,\xi,\eta) = \frac{1}{2}\|w\|^2 + C\sum_p(\xi_p + \eta_p)$$

with the following constraints on the errors $w \cdot x^p + b - y^p$:

$$\begin{aligned} -\xi_p - \epsilon &\le w \cdot x^p + b - y^p, \quad \text{(model below target)} \\ \eta_p + \epsilon &\ge w \cdot x^p + b - y^p, \quad \text{(model above target)} \\ \xi_p, \eta_p &\ge 0 \end{aligned}$$

- This leads to the Lagrangian

$$L(w, b, \xi, \eta, \alpha, \beta, \gamma, \delta) = \frac{1}{2}\|w\|^2 + C\sum_p (\xi_p + \eta_p)$$

$$- \sum_p \alpha_p (w \cdot x^p + b - y^p + \xi_p + \epsilon)$$

$$+ \sum_q \beta_q (w \cdot x^q + b - y^q - \eta_q - \epsilon) - \sum_p \gamma_p \xi_p - \sum_q \delta_q \eta_q$$

with $\alpha_p, \beta_q, \gamma_r, \delta_s$ all ≥ 0

- Setting $\Theta(\alpha, \beta, \gamma, \delta) = \min_{w,b,\xi,\eta} L(w, b, \xi, \eta, \alpha, \beta, \gamma, \delta)$, we have by construction

$$\Theta(\alpha, \beta, \gamma, \delta) \leq L(w, b, \xi, \eta, \alpha, \beta, \gamma, \delta) \leq f(w, b, \xi, \eta)$$

- We derive the dual function solving the equations

$$\frac{\partial L}{\partial w_i} = 0,\ \frac{\partial L}{\partial b} = 0,\ \frac{\partial L}{\partial \xi_p} = 0,\ \frac{\partial L}{\partial \eta_p} = 0$$

  - From $\nabla_w L = 0$ we get

$$w = \sum_p \alpha_p x_p - \sum_q \beta_q x_q$$

  - From $\frac{\partial L}{\partial b} = 0$ we obtain

$$\sum \alpha_p = \sum \beta_q$$

  - And from $\frac{\partial L}{\partial \xi_p} = 0$, $\frac{\partial L}{\partial \eta_q} = 0$ we get

$$C = \alpha_p + \gamma_p,\quad C = \beta_q + \delta_q$$

- And we next plug these back in L

Simplifying the Lagrangian

- As done in SV classification, we rewrite the Lagrangian to exploit these equalities to simplify it

$$
\begin{aligned}
L(w,b,\xi,\eta,\alpha,\beta,\gamma,\delta) \quad = \quad & \sum_p \xi_p (C - \alpha_p - \gamma_p) + \\
& \sum_q \eta_q (C - \beta_q - \delta_q) - \\
& \frac{1}{2} w \cdot w - w \cdot \left(\sum_p \alpha_p x^p - \sum_q \beta_q x^q \right) + \\
& b \left(\sum \alpha_p - \sum \beta_q \right) - \\
& \epsilon \sum_p (\alpha_p + \beta_p) + \sum_p y^p (\alpha_p - \beta_p)
\end{aligned}
$$

SVR's Dual Problem

- Working things out, the minus dual function that we write as Θ, becomes

$$\Theta(\alpha, \beta, \gamma, \delta) = \frac{1}{2}\sum_{p,q}(\alpha_p - \beta_p)(\alpha_q - \beta_q)x^p \cdot x^q + \epsilon\sum_p(\alpha_p + \beta_p) - \sum_p y^p(\alpha_p - \beta_p)$$

  - γ and δ drop out of Θ
- Since $\xi_p \geq 0, \eta_q \geq 0$, the previous C constraints become

$$0 \leq \alpha_p \leq C, \;\; 0 \leq \beta_q \leq C$$

- Thus, the dual problem becomes

$$\min_{\alpha,\beta} \Theta(\alpha, \beta) \text{ subject to } 0 \leq \alpha_p, \beta_q \leq C, \sum \alpha_p = \sum \beta_q$$

Solving the SVR Dual Problem

- It can be shown that **the dual gap is 0**, i.e., if $(w^*, b^*, \xi^*, \eta^*)$ and (α^*, β^*) are primal and dual optima respectively, then $f(w^*, b^*, \xi^*, \eta^*) = \Theta(\alpha^*, \beta^*)$
- Things are a little bit easier if we remove the (trickier) constraint $\sum \alpha_p = \sum \beta_q$ by dropping b, i.e., assuming a homogeneous model $w \cdot x$
  - Then we only have box constraints and we can simply apply projected gradient descent
  - But risk ending in a worse model (unless we center everything)
- But the dual problem is also easy to solve, for which a simple variant of the SMO algorithm is used

KKT Conditions

- From $f(w^*, b^*, \xi^*, \eta^*) = L(w^*, b^*, \xi^*, \eta^*, \alpha^*, \beta^*, \gamma^*, \delta^*)$ we deduce the complementary slackness KKT conditions

$$
\begin{aligned}
0 &= \alpha_p^*(w^* \cdot x^p + b^* - y^p + \xi_p^* + \epsilon);\\
0 &= \beta_q^*(w^* \cdot x^q + b^* - y^q - \eta_q^* - \epsilon);\\
0 &= \gamma_p^* \xi_p^* = (C - \alpha_p^*)\xi_p^*;\\
0 &= \delta_q^* \eta_q^* = (C - \beta_q^*)\eta_q^*
\end{aligned}
$$

- Thus, if $0 < \alpha_p^* < C$, we have $\xi_p^* = 0$ and $w^* \cdot x^p + b^* - y^p = -\epsilon$
- Similarly, if $0 < \beta_q^* < C$, we have $\eta_q^* = 0$ and $w^* \cdot x^q + b^* - y^q = \epsilon$
- Either one can be used to derive b^* once w^* is known

- The corresponding x^p, x^q are called **support vectors**
  - Now they define the ϵ–tube envelope around the true model
- Also $\xi_p^* > 0$ implies $\alpha_p^* = C$ and $\eta_q^* > 0$ implies $\beta_q^* = C$
- The optimal w^* is $w^* = \sum(\alpha_p^* - \beta_p^*)x^p$, with

$$\alpha_p^* \beta_p^* = 0$$

for notice that a given x^p can only verify one of the conditions

$$\begin{aligned} w^* \cdot x^q + b^* - y^q + \xi_p^* &= \epsilon, \\ w^* \cdot x^q + b^* - y^q - \eta_q^* &= -\epsilon \end{aligned}$$

The Kernel Trick for SVR I

- Again, stating and solving the the dual problem only requires computing dot products
- Also, the model applied to a new x is

$$f(x) = b^* + \sum (\alpha_p^* - \beta_p^*) x^p \cdot x$$

- Thus, the kernel trick can be used again to project the original patterns x into larger dimensional patterns $\Phi(x)$

The Kernel Trick for SVR II

- Again, we do not deal with the $\Phi(x)$ but just work with $\Phi(x) \cdot \Phi(x') = k(x, x')$
- The model is applied as

$$
\begin{aligned}
b^* + w^* \cdot \Phi(x) &= b^* + \sum (\alpha_p^* - \beta_p^*) \Phi(x^p) \cdot \Phi(x) \\
&= b^* + \sum (\alpha_p^* - \beta_p^*) k(x^p, x)
\end{aligned}
$$

- If we use a Gaussian kernel, the model becomes

$$
f(x; w^*, b^*) = b^* + \sum (\alpha_p^* - \beta_p^*) \;\; e^{-\gamma \|x^p - x\|^2}
$$

i.e., a sum of Gaussians centered at the x^p

Hyperparameterizing C, γ and ϵ

- C and γ are explored as in SV classification
- In a reasonable model ϵ shouldn't be larger than σ_y
  - We can try ϵ values of the form

$$2^k \sigma_y, \quad -K \leq k \leq -1$$

  - $\sigma_y = 1$ if we use a `TransformedTargetRegressor` with `StandardScaler()`
- But we have to explore three hyperparameters which is going to be quite costly
- The stopping tolerance is also somewhat tricky as it depends on gradient properties
  - The default 10^{-3} should be OK on medium size problems if we use a `TransformedTargetRegressor`
- Some guidelines can be found on LIBSVM home pages

Overfitting and Underfitting

- As in SVC, large C and γ will result in overfit unless ϵ is large
- A large C forces slacks to be near 0 and thus perfect training fit
  - This is parallel to what happened in Ridge regression, since $\frac{1}{CN}$ behaves as α
- Large γ result in sharp Gaussians
- But models with small C and γ will likely underfit
- Large ϵ models will usually underfit
  - At the extreme there will be no slacks and we are likely to end in a near constant model
  - On the other hand, a very small ϵ will force 0 slacks and possible overfit
- But the joint effects of C, γ and ϵ may change the preceding observations

Takeaways on SVR I

- The primal SVR problem can be written as a regularized loss function

$$\min_{w,b} f(w,b) = \sum_p [y^p - (w \cdot x^p + b)]_\epsilon + \frac{\lambda}{2}\|w\|^2$$

- If $\mathcal{C} = \{\alpha, \beta : 0 \leq \alpha_p, \beta_p \leq C, \sum \alpha_p = \sum \beta_p\}$, the dual problem is now

$$\begin{aligned}\min_{\mathcal{C}} \quad \Theta(\alpha,\beta) \quad &= \quad \frac{1}{2}\sum_{p,q}(\alpha_p - \beta_p)(\alpha_q - \beta_q)x^p \cdot x^q + \\ &\quad \epsilon \sum_p (\alpha_p + \beta_p) - \sum_p y^p(\alpha_p - \beta_p)\end{aligned}$$

Takeaways on SVR II

- A variant of SMO can again be used, with a cost $\Omega(N^2)$
- KKT conditions are again used to obtain w^* and b^* from α^*, β^*
- And again SVs, i.e., vectors x^p for which $\alpha_p^* > 0$ or $\beta_p^* > 0$ define the SVR model
- Using a Gaussian kernel we arrive at a final model

$$f(x; w^*, b^*) = b^* + \sum (\alpha_p^* - \beta_p^*) \ \ e^{-\gamma \|x^p - x\|^2}$$

- Two hyperparameters appear: the penalty C and the ϵ tube width
- Plus the width γ if we use a Gaussian kernel