Linear models

Oriol Pujol

Introduction to Machine Learning

2013

Acknowledgements

This work is inspired in the courses of T. Jaakkola, M. Collins, L. Kaelbling, and T. Poggio at MIT, Andrew Ng at Stanford, Y. Abu-Mostafa at CalTech, E. Xing at CMU, and all my mentors and people who made me realize Machine Learning is one of my passions.

Outline

- The perceptron
- Perceptron as an Instance Based Learning
- An optimization approach to linear classification
- Support Vector Machines
- A whirlwind tour into convex optimization
- Support Vector Machines in the Dual

The supervised learning problem

The components of the learning model (what we have to define):

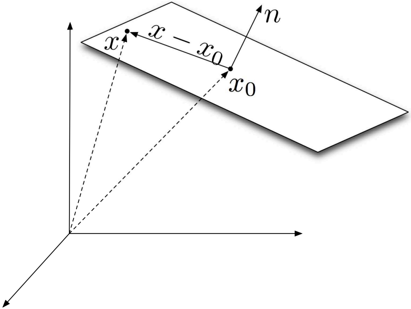
1. The hypotheses space $\mathcal{H}$, e.g. linear models.
2. The real, plausible, or friendly error measure, e.g. sum of squared differences.
3. The optimization/ search algorithm, e.g. gradient descent.

Review of linear model basic geometry.

Given $x \in \mathbf{R^N}$ we define a hyperplane as the set of points that fulfills the following relation

$$\{x | a^T x = b\}$$

It can be decomposed as $\{x | n^T(x - x_0) = 0\}$ where x_0 is the *base point* and n is the *normal* vector to the hyperplane.



Our first classifier

For input $\mathbf{x} = (x_1, \ldots, x_d)$ attributes/features.

- $\mathbf{x}$ belongs to class '5' if:

$$\sum_{i=1}^{d} w_i x_i > \text{threshold}$$

- $\mathbf{x}$ belongs to class '1' if:

$$\sum_{i=1}^{d} w_i x_i \leqslant \text{threshold}$$

This can be written as $h \in \mathcal{H}$

$$h(\mathbf{x}) = \ \text{sign}\Big(\sum_{i=1}^{d} w_i x_i - \text{threshod} \Big)$$

Our first classifier

$$h(\mathbf{x}) = \text{sign}\Big(\sum_{i=1}^{d} w_i x_i - w_0\Big)$$

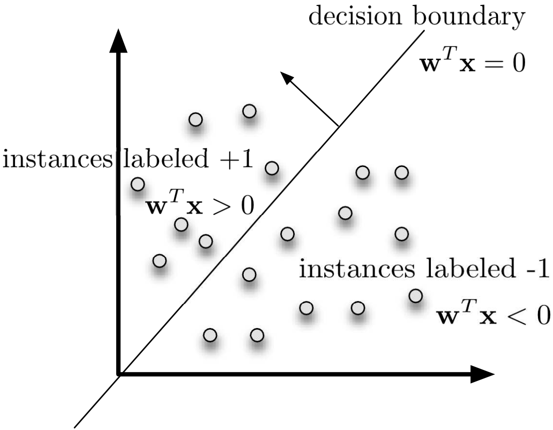
We can introduce an artificial coordinate $x_0 = 1$ and then

$$h(\mathbf{x}) = \text{sign}\Big(\sum_{i=0}^{d} w_i x_i\Big)$$

that in vector form it is

$$h(x) = \text{sign}(\mathbf{w}^T \mathbf{x})$$

Our first classifier


decision boundary
$\mathbf{w}^T\mathbf{x} = 0$
instances labeled +1
$\mathbf{w}^T\mathbf{x} > 0$
instances labeled -1
$\mathbf{w}^T\mathbf{x} < 0$

Our first classifier

The model

- The hypotheses space: Linear classifier – linear combination of the input features.
- Measure of error/fitness: Training error (*zero-one loss*) $e(h(\mathbf{x}), y)) = [\![h(\mathbf{x}) \neq y)]\!]$
- The learning algorithm: Perceptron Learning Algorithm

Our first classifier

The model

- The hypotheses space: Linear classifier – linear combination of the input features.
- Measure of error/fitness: Training error (*zero-one loss*) $e(h(\mathbf{x}), y)) = [\![h(\mathbf{x}) \neq y)]\!]$
- The learning algorithm: Perceptron Learning Algorithm

Perceptron Learning Algorithm

1. Pick a misclassified example x_i such that

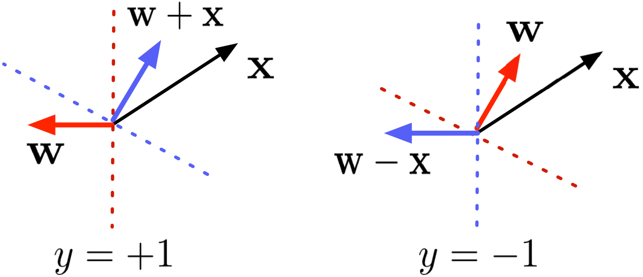
$$\text{sign}(\mathbf{w}^T \mathbf{x}_i) \neq y_i$$

2. Update the weight vector:

$$w' \leftarrow w + y_i \mathbf{x}_i$$

Our first classifier

The basic idea of PLA.



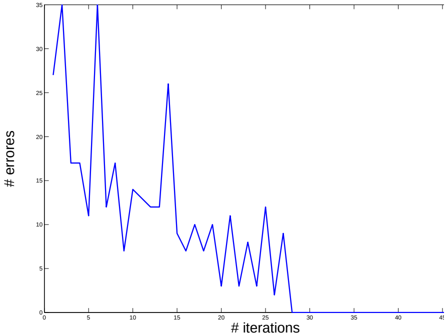
Consider a mistake on $\mathbf{x}_i$ i.e. $y_i \mathbf{w}^T \mathbf{x}_i < 0$. Consider what happens if we re-classify after the update $w' \leftarrow w + y_i \mathbf{x}_i$,

$$y_i \mathbf{w}'^T \mathbf{x}_i = y_i(\mathbf{w} + y_i \mathbf{x_i})^T \mathbf{x}_i = y_i \mathbf{w}^T \mathbf{x}_i + y_i^2 \mathbf{x}_i^T \mathbf{x}_i = y_i \mathbf{w}^T \mathbf{x}_i + \|x_i\|^2$$

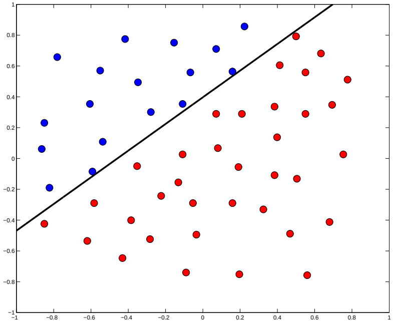
Our first classifier

Consider the following separable problem

Error convergence



Classification result

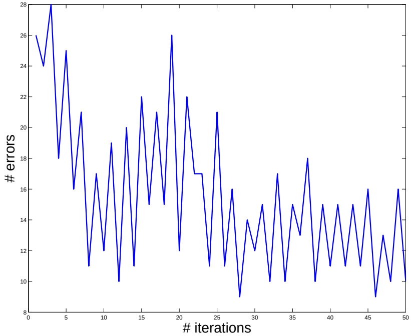


(It can be shown that perceptron converges in a finite number of iterations. To be proved if we have time.)

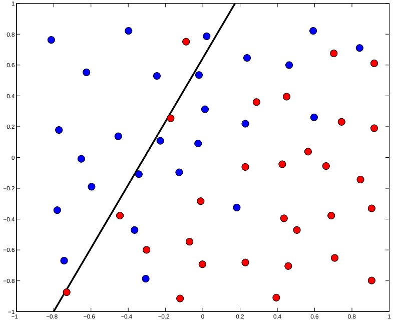
Our first classifier

Consider the following non separable problem

Error convergence



Classification result

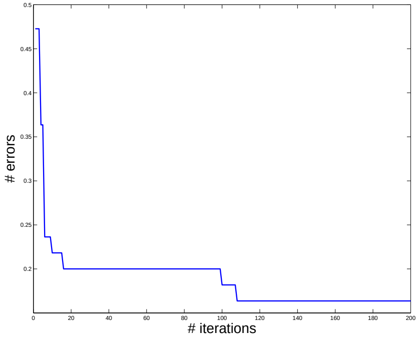


See any problem? Do you expect the algorithm to converge?

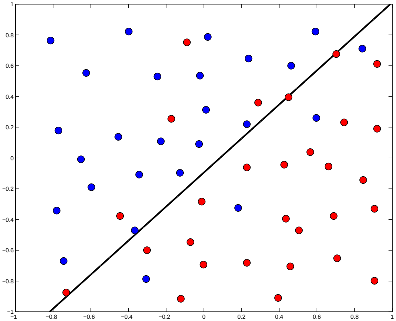
Our first classifier

Can we make it better?. What if we keep the best up to date parameters? This is called pocket preceptron.

Error convergence



Classification result



Not so bad, right? ... but still not perfect.

Relationship of the perceptron algorithm with IBL

We can connect the perceptron with Instance Based Learning. Consider the perceptron decision rule:

$$f(x_q) = \text{sign}\Big(\sum_j w_j x_{qj}\Big).$$

Relationship of the perceptron algorithm with IBL

We can connect the perceptron with Instance Based Learning. Consider the perceptron decision rule:

$$f(x_q) = \text{sign}\Big(\sum_j w_j x_{qj}\Big).$$

Observe that

$$w_j = \sum_i \alpha_i y_i x_{ij}.$$

Relationship of the perceptron algorithm with IBL

We can connect the perceptron with Instance Based Learning. Consider the perceptron decision rule:

$$f(x_q) = \text{sign}\Big(\sum_j w_j x_{qj}\Big).$$

Observe that

$$w_j = \sum_i \alpha_i y_i x_{ij}.$$

So

$$f(x_q) = \text{sign}\Big(\sum_j \Big(\sum_i \alpha_i y_i x_{ij}\Big) x_{qj}\Big) = \text{sign}\Big(\sum_i \alpha_i y_i \mathbf{x}_q^T \mathbf{x}_i\Big)$$

The perceptron is a special case of weighted kNN when the similarity function is the **dot product**.

Relationship of the perceptron algorithm with IBL (II)

So in general we can generalize the equation as,

$$f(x_q) = \text{sign}\Big(\sum_i \alpha_i y_i \mathcal{K}(\mathbf{x}_q, \mathbf{x}_i)\Big),$$

where $\mathcal{K}(a, b)$ is the similarity between a and b.

A convex optimization perspective to linear classification

Consider an optimization problem in the standard form:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array}$$

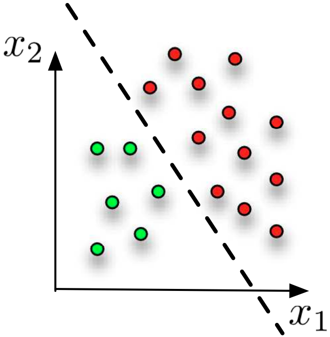
with variable $x \in \mathbf{R}^n$. We assume its domain $\mathcal{D} = \bigcap_{i=0}^{m} \textbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \textbf{dom}\, h_i$ is nonempty, and the optimal value of the objective is given by $p^* = f_0(x^*)$.

A convex optimization perspective to linear classification.

Linear discrimination

Separate two sets of points $\{r_1, \ldots, r_N\}$ and $\{s_1, \ldots, s_M\}$ by a linear model:

$$a^T r_i + b < 0, \ \forall i = 1, \ldots, N \quad a^T s_i + b > 0, \ \forall i = 1, \ldots, M$$



A convex optimization perspective to linear classification.

First, we should observe that because $a^T r_i + b < 0$ and $a^T s_i + b > 0$ are *homogeneous* with respect to a, b, the problem is equivalent to

$$a^T r_i + b \leqslant -1,\ \forall i = 1, \dots, N \quad a^T s_i + b \geqslant 1\ \forall i = 1, \dots, M$$

Consider the following:

1. Because of homogeneity with respect (a, b), $a^T r_i + b < 0 \equiv \xi(a^T r_i + b) < 0$
2. The solution $(a, b)^*$ satisfies $\xi(a^T r_i + b) < 0$. We can evaluate $\xi({a^*}^T r_i + b^*) = -\zeta$. Thus, we can replace the strict inequality $\xi(a^T r_i + b) < 0$ by the nonstrict inequality $\xi(a^T r_i + b) \leqslant -\zeta$.
3. Since ξ can take any value, we can use $\xi = \zeta$ obtaining

$$a^T r_i + b < 0 \equiv a^T r_i + b \leqslant -1$$

A convex optimization perspective to linear classification.

We can write this problem in standard form as a feasibility problem:

$$\begin{array}{ll} \text{minimize} & 1 \\ \text{subject to} & a^T r_i + b \leqslant -1,\ \forall i = 1, \ldots, N \\ & a^T s_i + b \geqslant 1\ \forall i = 1, \ldots, M \end{array}$$

or alternatively as

$$\begin{array}{ll} \text{minimize} & \max(d((a^T, b), \mathcal{C}_1^{(i)}), d((a^T, b), \mathcal{C}_2^{(j)})) \\ \text{where} & \mathcal{C}_1 \equiv a^T r_i + b \leqslant -1,\ \forall i = 1, \ldots, N \\ & \mathcal{C}_2 \equiv a^T s_j + b \geqslant 1\ \forall j = 1, \ldots, M \end{array}$$

Alternating projections.

Goal:

Find a weight vector that fulfills all the constraints (is feasible).

Alternating projections:

Project the current weight onto the feasible set defined by the most violated constraint. Repeat until convergence.

Consider the simplified problem:

$$\begin{array}{ll} \text{minimize} & 1 \\ \text{subject to} & \text{"}y_i w^T x_i \geqslant 0\text{"},\ \forall i = 1, \dots, N \end{array}$$

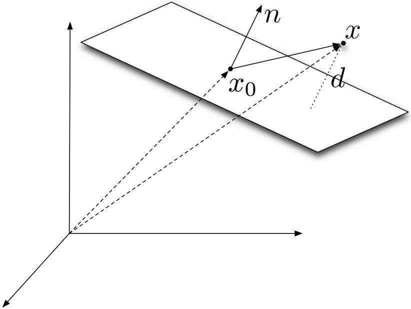
Algorithm:

$$w^{(k+1)} = P_{\mathcal{C}_j}(w^{(k)})$$

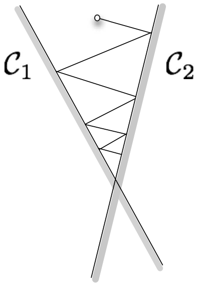
Remember linear model basic geometry.

Observe that if $x \notin \pi(x)$ then;

- $d(x, \pi) = |n^T(x - x_0)| / \|n\|_2$ is the distance from x to the hyperplane.
- and the sign$(n^T(x - x_0))$ identifies the half-space where the point lies.



Alternating projections.



Consider the projection of a weight point w_0 onto a linear feasible set $\mathcal{C}_j \equiv y_j w^T x_j \geqslant 0$,

$$d(w_0, \mathcal{C}_j) = \frac{y_j w_0^T x_j}{\|x_j\|_2},$$

$$P_{\mathcal{C}_j}(w_0) = \begin{cases} w_0 & w_0 \in \mathcal{C}_j \\ w_0 + \left(\frac{y_j w_0^T x_j}{\|x_j\|_2}\right) y_j x_j & w_0 \notin \mathcal{C} \end{cases}$$

Alternating projections.

Rearranging terms, the algorithm becomes:

1. Given the current weight vector $w^{(k)}$, find the most violated constraint of the set,

$$\arg\min_j (y_j w^{(k)T} x_j) \text{ subject to } y_j w^{(k)T} x_j \leqslant 0$$

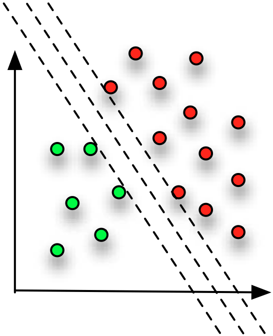
2. Update weights.

$$w^{(k+1)} = w^{(k)} + \underbrace{\eta}_{\left(\frac{y_j w^{(k)T} x_j}{\|x_j\|_2}\right)} y_j x_j$$

Meet the mighty Perceptron Algorithm, ... again!

A convex optimization perspective to linear classification.

Given any possible hyperplane we can select the one that maximally separates both point sets. It can be seen as putting the thickest slab between both classes.



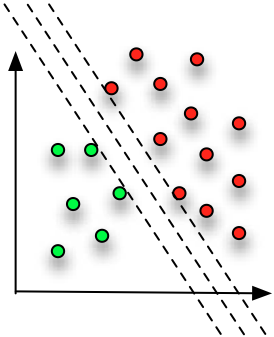
A convex optimization perspective to linear classification.

The Euclidean distance between hyperplanes

$\pi_1 \equiv \{z | a^T z + b = 1\}$

$\pi_2 \equiv \{z | a^T z + b = -1\}$

is $d(\pi_1, \pi_2) = 2/\|a\|$. This distance is also called margin.



To separate two sets of points with maximum margin

$$\begin{array}{ll} \text{minimize} & (1/2)\|a\|_2 \\ \text{subject to} & a^T r_i + b \leqslant -1, \quad \forall i = 1, \dots, N \\ & a^T s_i + b \geqslant 1 \quad\quad \forall i = 1, \dots, M, \end{array}$$

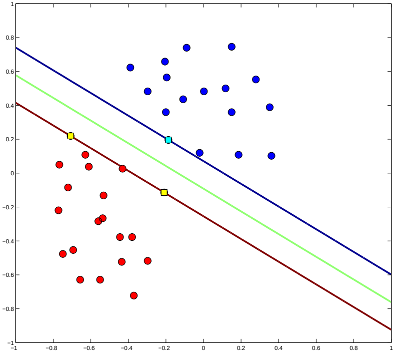
after squaring the objective it becomes a Quadratic Programming on a, b.

A convex optimization perspective to linear classification.

This formulation is called *Support Vector Machine*

Property

The solution depends on a small set of data points, namely *support vectors*. Those points that fulfill the conditions $a^T s_i + b = 1$ or $a^T r_i + b = -1$.



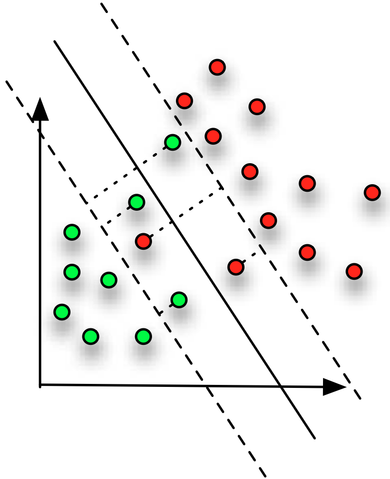
A convex optimization perspective to linear classification.

Properties

1. Unique solution.
2. Robustness to noisy examples.
3. leave-one-out CV error $\leqslant \frac{\#\text{ of support vectors}}{N}$

A convex optimization perspective to linear classification.

Consider now the case in which the sets of points are not linearly separable.



A convex optimization perspective to linear classification.

In that case, we want to minimize the number of misclassifications. The problem is that this is a combinatorial search and it is presumably 'NP' hard.
We may find a convex relaxation of the problem by minimizing an L1-norm. L1-norm is sparse and thus is a good heuristic for minimizing the number of non-zero elements.

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{d} |x_i|$$

A convex optimization perspective to linear classification.

$$\begin{array}{lll} \text{minimize} & u^T\mathbf{1} + v^T\mathbf{1} & \\ \text{subject to} & a^T r_i + b \leqslant -1 + v_i, & \forall i = 1, \ldots, N \\ & a^T s_i + b \geqslant 1 - u_i & \forall i = 1, \ldots, M, \\ & u \succeq 0, \quad v \succeq 0 & \end{array}$$

- an LP in a, b, u, v
- at optimum, $u_i = \max\{0, 1 - a^T r_i - b\}, v_i = \max\{0, 1 + a^T s_i + b\}$
- can be interpreted as a heuristic for minimizing #misclassified points

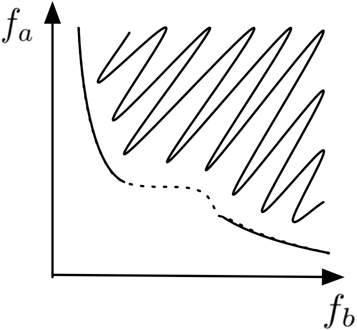
A convex optimization perspective to linear classification.

What about maximizing the margin $2/\|a\|_2$ and minimize the number of misclassified points $u^T\mathbf{1} + v^T\mathbf{1}$?

This problem is a clear example of multi-objective optimization $F_0 = (\|a\|_2, u^T\mathbf{1} + v^T\mathbf{1})$. The optimum in this problem is called Pareto's optimal surface.

A convex optimization perspective to linear classification.

Given the multi-criterion objective $F_0 = (f_a, f_b)$. The pareto optimal surface or regularization path consists on the set of all **minimal** (not minimum) points.



Scalarization $f_0 = f_a + \lambda f_b$ allows to travel along the pareto optimal surface. Observe that in the case of a convex set, it allows to travel the complete surface.

A convex optimization perspective to linear classification.

A simple way of solving this trade-off problem is by means of scalarization: $f_0 = \|a\|_2 + \lambda(u^T\mathbf{1} + v^T\mathbf{1})$

$$\begin{array}{ll} \text{minimize} & \|a\|_2 + \lambda(u^T\mathbf{1} + v^T\mathbf{1}) \\ \text{subject to} & a^T r_i + b \leqslant -1 + v_i, \qquad \forall i = 1, \dots, N \\ & a^T s_i + b \geqslant 1 - u_i \qquad \forall i = 1, \dots, M, \\ & u \succeq 0, \quad v \succeq 0 \end{array}$$

This is the famous Support Vector Machine!

A convex optimization perspective to linear classification.

Let us change a bit the formulation so that it is consistent with our notation.

$$\begin{array}{ll} \text{minimize} & \|a\|_2 + \lambda u^T \mathbf{1} \\ \text{subject to} & y_i(a^T x_i + b) \geqslant 1 - u_i \qquad \forall i = 1, \ldots, N \\ & u \succeq 0 \end{array}$$

A convex optimization perspective to linear classification.

Let us change a bit the formulation so that it is consistent with our notation.

$$\begin{array}{ll} \text{minimize} & \|a\|_2 + \lambda u^T \mathbf{1} \\ \text{subject to} & y_i(a^T x_i + b) \geqslant 1 - u_i \qquad \forall i = 1, \dots, N \\ & u \succeq 0 \end{array}$$

We can rewrite the constraints as

$$\begin{array}{ll} \text{minimize} & \|a\|_2 + \lambda u^T \mathbf{1} \\ \text{subject to} & u_i \geqslant 1 - y_i(a^T x_i + b) \qquad \forall i = 1, \dots, N \\ & u \succeq 0 \end{array}$$

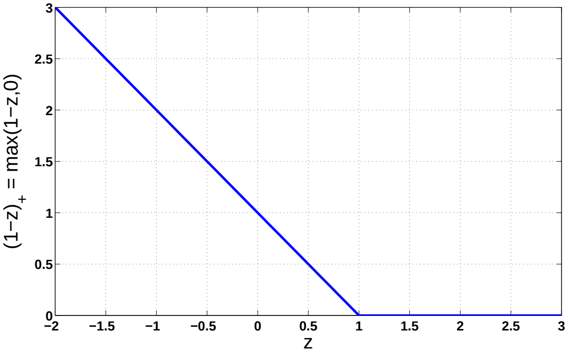
and observe that the solution of minimizing $u^T \mathbf{1}$ subject to these constraints is achieved when either u_i is either strictly zero or $y_i(a^T x_i + b)$. Note that given the optimal a^* any value above those is not optimal. Thus we only care about these two equality cases and rewrite the complete formulation as an unconstrained optimization problem

$$\text{minimize} \quad \|a\|_2 + \lambda \sum \max(0, 1 - y_i(a^T x_i + b))$$

Support vector machines.

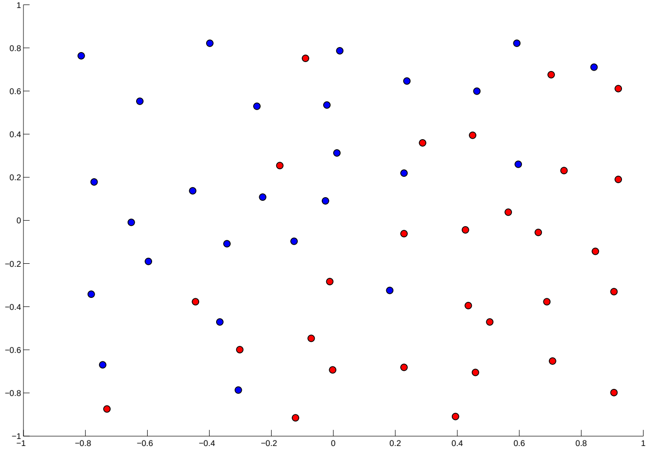
Let us look at the individual equality values for u_i. The first constraint is a line with negative slope. The second constraint tells us that we only care about positive values.

Hinge loss: $(1 - z)_+ = max(1 - z, 0)$



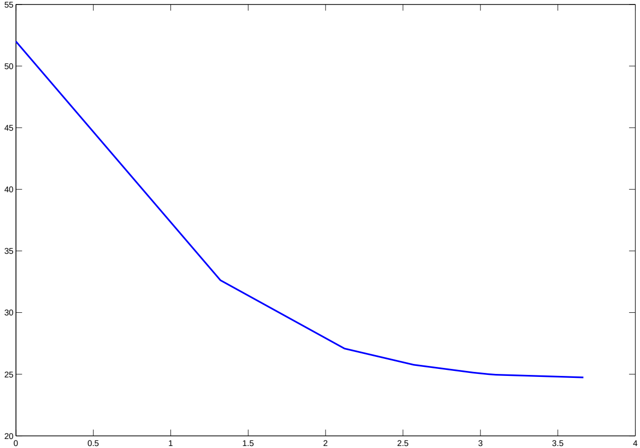
An example.

Consider the following problem.

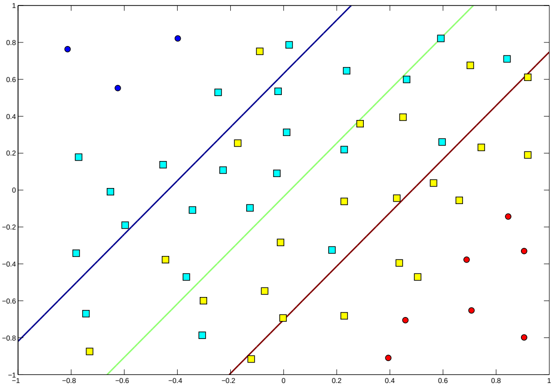


An example.

By varying λ and solving we can evaluate the multi-criterion objective functions, and plot the Pareto optimal surface. (Abcissa: $\|\mathbf{a}\|_2$; ordinate: $\mathbf{u}^T\mathbf{1}$)

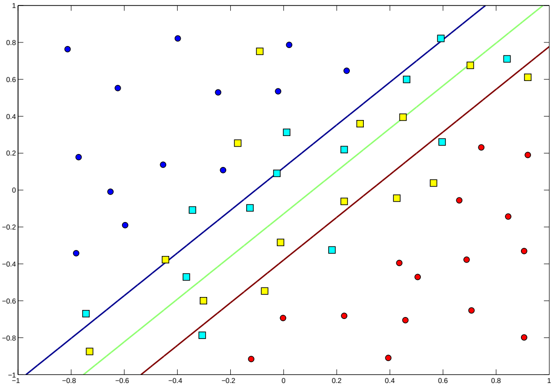


An example.



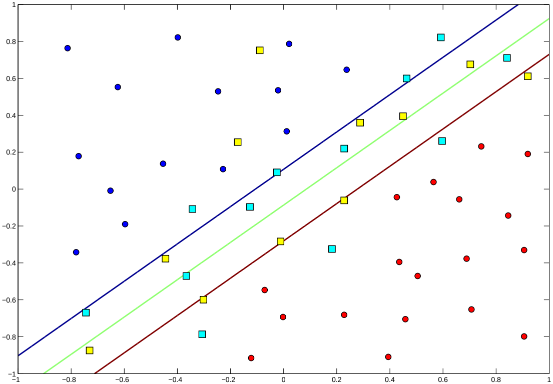
$$(\lambda = 0.1)$$

An example.



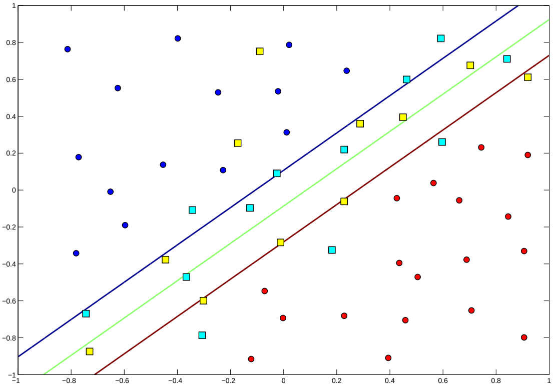
$(\lambda = 1)$

An example.



$$(\lambda = 10)$$

An example.



$(\lambda = 1000)$

A convex optimization perspective to linear classification.

A word of caution with MOP

Multi-objective optimization problems trade off two objective values. In classification we are clearly concerned with misclassification errors. Reporting just misclassification errors is reporting half of the problem.

Convex optimization in 10'

Consider an optimization problem in the standard form:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p, \end{array}$$

with variable $x \in \mathbf{R}^n$. We assume its domain $\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i$ is nonempty, and the optimal value of the objective is given by $p^* = f_0(x^*)$.

Convex optimization in 10'

Basic idea: Lagrangian duality is to take the constraints into account by augmenting the objective function with a weighted sum of the constraint functions.

We define the *Lagrangian* $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \rightarrow \mathbf{R}$ associated with the former problem as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x),$$

with $\mathbf{dom} L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p$. The vectors λ and ν are called *dual variables* or *Lagrange multiplier verctors* associated.

Convex optimization in 10'

We define the *Lagrange dual function* $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ as the minimum value of the Lagrangian over x: for $\lambda \in \mathbf{R}^m$, $\nu \in \mathbf{R^p}$,

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} (f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x))$$

Observation:

This is concave, even when the primal problem is not convex. It yields a lower bound on the optimal value p^*. For any $\lambda \succeq 0$ and ν we have

$$g(\lambda, \nu) \leqslant p^*$$

This is only true if $\lambda \succeq 0$ because $g(\nu, \lambda) = \inf_x L(x, \lambda, \nu) \leqslant L(x, \lambda, \nu) \leqslant f_0(x)$. Observe that if x is a feasible point then $h_i(x) = 0$ and $f_i(x) \leqslant 0$, thus if λ is positive then $L(x, \lambda, \nu) \leqslant f_0(x)$.

Convex optimization in 10'

For each pair (λ, ν) with $\lambda \succeq 0$, the Lagrange dual function gives a lower bound on the optimal value. The "best" lower bound defined by

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

is called the *Lagrange dual problem* and the original problem *primal*.

Convex optimization in 10'

Let us define d^* as the optimal value of the Lagrange dual function. We have seen that for any problem (even non-convex ones)

$$d^* \leqslant p^*$$

When $d^* = p^*$ the weak duality inequality becomes tight then it is called *strong duality*.
Usually, if the primal problem is convex in all $f_i(x)$ then strong duality holds.

Convex optimization in 10'

Suppose strong duality holds for a given problem, then

$$\begin{aligned} f_0(x^*) &= g(\lambda^*, \nu^*) \\ &= \inf_x (f_0(x) + \sum_i \lambda_i^* f_i(x) + \sum_i \nu_i^* h_i(x)) \\ &\leqslant f_0(x^*) + \sum_i \lambda_i^* f_i(x^*) + \sum_i \nu_i^* h_i(x^*) \\ &\leqslant f_0(x^*) \end{aligned}$$

For strong duality this is only possible if $\sum_i \lambda_i^* f_i(x^*) = 0 \implies$

$$\lambda_i^* f_i(x^*) = 0, \quad i = 1, \ldots, m$$

This condition is known as *complementary slackness*.

Convex optimization in 10'

Assume that $f_0, \ldots, f_m, h_1, \ldots, h_p$ are differentiable (no convexity assumed) and x^* and (λ^*, ν^*) the primal and dual optimal points with zero duality gap. Since x^* minimizes $L(x, \lambda^*, \nu^*)$. It follows that its gradient must vanish at x^*, i.e.,

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} \nu_i^* \nabla h_i(x^*) = 0$$

Karush-Kuhn-Tucker optimality conditions

Summarizing the results we obtain the KKT conditions:

1. *Primal constraints:* It states that x has to be feasible,
   $f_i(x) \preceq 0, \quad i = 1, \ldots, m$
   $h_i(x) = 0, \quad i = 1, \ldots, p$
2. *Dual constraints:* It states the dual feasibility.
   $\lambda \succeq 0$
3. *Complementary slackness:*
   $\lambda_i f_i(x) = 0, \quad i = 1, \ldots, m$
4. *The gradient of the Lagrangian function with respect to x vanishes:*
   $\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0$

If **strong duality** holds and (x, λ, ν) are optimal and they must satisfy KKT conditions. For a **convex** problem – $f_i(x)$ convex and $h_i(x)$ affine – if KKT conditions hold for (x, λ, ν) then they are optimal and KKT becomes a sufficient condition for optimality.

SVM Dual Problem

Recall the *primal problem*:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}a^Ta + \lambda u^T\mathbf{1} \\ \text{subject to} & y_i(a^Tx_i + b) \geqslant 1 - u_i \qquad \forall i = 1, \ldots, N \\ & u \succeq 0 \end{array}$$

SVM Dual Problem

Recall the *primal problem*:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}a^Ta + \lambda u^T\mathbf{1} \\ \text{subject to} & y_i(a^Tx_i + b) \geqslant 1 - u_i \qquad \forall i = 1, \dots, N \\ & u \succeq 0 \end{array}$$

The Lagrangian is

$$L(a, b, u, \nu, \eta) = \frac{1}{2}a^Ta + \lambda u^T\mathbf{1} - \sum_{i=1}^{N} \nu_i(y_i(a^Tx_i + b) - 1 + u_i) - \sum_{k=1}^{N} \eta_k u_k$$

and the *dual problem* is

$$\arg\max_{\nu,\eta\geqslant 0} \inf_{c,u} L(a, b, u, \nu, \eta)$$

SVM Dual Problem

Minimize L w.r.t. (c, u):

- (1) $\frac{\partial L}{\partial a} = 0 \implies a = \sum_{i=1}^{N} \nu_i y_i x_i$
- (2) $\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^{N} \nu_i y_i = 0$
- (3) $\frac{\partial L}{\partial u_i} = 0 \implies \lambda - \nu_i - \eta_i = 0 \implies 0 \leqslant \nu_i \leqslant \lambda$

From (2), plugging $\eta_i = \lambda - \nu_i$ in the Lagrangian we get

$$L(a, b, u, \nu, \eta) = \frac{1}{2} a^T a - \sum_{i=1}^{N} \nu_i (y_i (a^T x_i + b) - 1)$$

$$L(a, b, u, \nu, \eta) = \frac{1}{2} a^T a - \sum_{i=1}^{N} \nu_i y_i (a^T x_i + b) + \sum_{i=1}^{N} \nu_i$$

SVM Dual Problem

From (1), plugging $a = \sum_{i=1}^{N} \nu_i y_i x_i$ in the Lagrangian we get

$$\arg\max_{\nu \geqslant 0} L(\nu) = \frac{1}{2} \sum_{i=1}^{N} \nu_i y_i x_i^T \sum_{j=1}^{N} \nu_j y_j x_j - \sum_{i=1}^{N} \nu_i y_i (\sum_{j=1}^{N} \nu_j y_j x_j^T) x_i$$

$$- \sum_{i=1}^{N} \nu_i y_i b + \sum_{i=1}^{N} \nu_i$$

From (2), the term $b \sum_{i=1}^{N} \nu_i y_i = 0$, leading to

$$\arg\max_{\nu \geqslant 0} L(\nu) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \nu_i \nu_j y_i y_j x_i^T x_j + \sum_{i=1}^{N} \nu_i$$

SVM Dual Problem

and we formulate the dual problem w.r.t. ν as

$$\begin{array}{ll} \text{maximize} & \nu^T\mathbf{1} - \frac{1}{2}\nu^T Q\nu \\ \text{subject to} & 0 \leqslant \nu_i \leqslant \lambda \qquad\qquad \forall i = 1,\dots,N \\ & \nu^T y = 0 \end{array}$$

where $Q = \nu^T\mathbf{diag}(\mathbf{y})X^T X\mathbf{diag}(\mathbf{y})\nu$. Although the primal an the dual are QP, and can be solved in roughly $\mathcal{O}(N^3)$, the dual problem is easier to solve – it has simple box constraints.
And the final classifier is

$$h(x) = \underbrace{\sum_{i=1}^{N} \nu_i y_i x_i^T x}_{\text{Instance Based Learning}} + b$$

Interpreting the optimal values of the dual variables.

Complementary slackness

- $\nu_i(y_i(a^T x_i - b) - 1 + u_i) = 0$
- $\eta_i u_i = 0$

and the constraint $\lambda - \nu_i - \eta_i = 0$ allows us to analyze the solution achieved:

(1) $\nu_i = 0$. **Non-SVs.**

(Set of examples that are not part of the solution.)

- From $\lambda - \nu_i - \eta_i = 0$ we obtain $\eta_i = \lambda$, hence using second complementary condition $u_i = 0$.
- From $\nu_i = 0$ we have $(y_i(a^T x_i - b) - 1 + u_i) > 0$. And using the former derivation $y_i(a^T x_i - b) > 1$.

Interpreting the optimal values of the dual variables.

(2) $\nu_i = \lambda$. **SVs violating the hard constraints.**

- From $\lambda - \nu_i - \eta_i = 0$ we obtain $\eta_i = 0$, hence $u_i > 0$.
- From $\nu_i > 0$ we have $(y_i(a^T x_i - b) - 1 + u_i) = 0$. And using the former derivation $y_i(a^T x_i - b) = 1 - u_i$.

(3) $0 < \nu_i < \lambda$. **SVs.**

- From $\lambda - \nu_i - \eta_i = 0$ we obtain $\eta_i > 0$, hence using second complementary condition $u_i = 0$.
- From $\nu_i > 0$ we have $(y_i(a^T x_i - b) - 1 + u_i) = 0$. Hence $y_i(a^T x_i - b) = 1$.

Computing the offset value: b

Note that b can be found using any SV from the set **(3)** by a simple substitution on $y_i(a^T x_i - b) = 1$. It is usual to check on all SVs and average the resulting b offsets (numerical stability).