Pattern Classification, and Quadratic Problems

(Robert M. Freund)

March 30, 2004

1 Overview

- Pattern Classification, Linear Classifiers, and Quadratic Optimization
- Constructing the Dual of CQP
- The Karush-Kuhn-Tucker Conditions for CQP
- Insights from Duality and the KKT Conditions
- Pattern Classification without strict Linear Separation

2 Pattern Classification, Linear Classifiers, and Quadratic Optimization

2.1 The Pattern Classification Problem

We are given:

- points $a^1, \ldots, a^k \in \Re^n$ that have property "P"
- points $b^1, \ldots, b^m \in \Re^n$ that do not have property "P"

We would like to use these $k+m$ points to develop a linear rule that can be used to predict whether or not other points x might or might not have property P. In particular, we seek a vector v and a scalar β for which:

- $v^T a^i > \beta$ for all $i = 1, \ldots, k$
- $v^T b^i < \beta$ for all $i = 1, \ldots, m$

We will then use v, β to predict whether or not other points c have property P or not, using the rule:

- If $v^T c > \beta$, then we declare that c has property P.
- If $v^T c < \beta$, then we declare that c does not have property P.

We therefore seek v, β that defines the hyperplane

$$H_{v,\beta} := \{x | v^T x = \beta\}$$

for which:

- $v^T a^i > \beta$ for all $i = 1, \ldots, k$
- $v^T b^i < \beta$ for all $i = 1, \ldots, m$

This is illustrated in Figure 1.

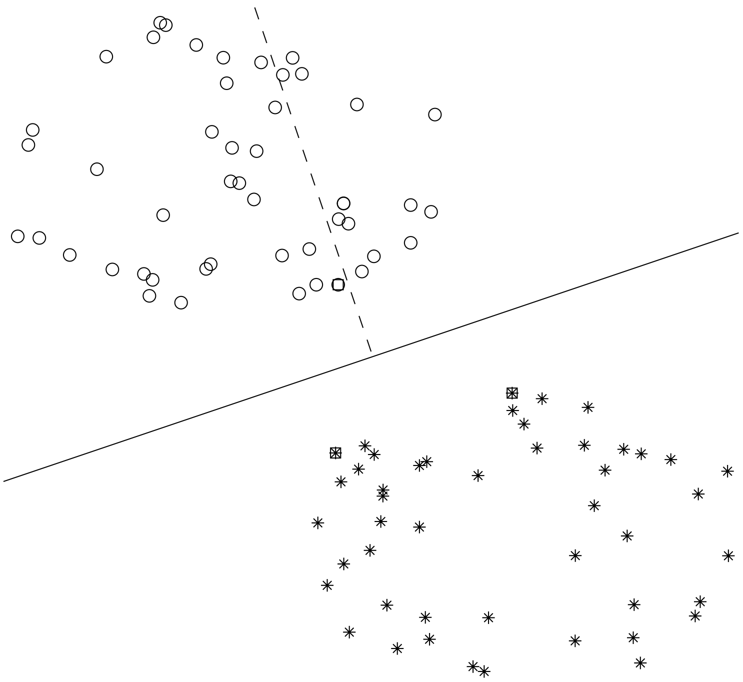

Figure 1: Illustration of the pattern classification problem.

2.2 The Maximal Separation Model

We seek v, β that defines the hyperplane

$$H_{v,\beta} := \{x | v^T x = \beta\}$$

for which:

- $v^T a^i > \beta$ for all $i = 1, \ldots, k$
- $v^T b^i < \beta$ for all $i = 1, \ldots, m$

We would like the hyperplane $H_{v,\beta}$ not only to separate the points with different properties, but to be as far away from the points $a^1, \ldots, a^k, b^1, \ldots, b^m$ as possible. It is easy to derive via elementary analysis that the distance from the hyperplane $H_{v,\beta}$ to any point a^i is equal to

$$\frac{v^T a^i - \beta}{\|v\|} \ .$$

Similarly, the distance from the hyperplane $H_{v,\beta}$ to any point b^i is equal to

$$\frac{\beta - v^T b^i}{\|v\|} \ .$$

If we normalize the vector v so that

$$\|v\| = 1 \ ,$$

then the minimum distance from the hyperplane $H_{v,\beta}$ to any of the points $a^1, \ldots, a^k, b^1, \ldots, b^m$ is then:

$$\min \left\{ v^T a^1 - \beta, \ldots, v^T a^k - \beta, \beta - v^T b^1, \ldots, \beta - v^T b^m \right\} \ .$$

We therefore would like v and β to satisfy:

- $\|v\| = 1$, and
- $\min \left\{ v^T a^1 - \beta, \ldots, v^T a^k - \beta, \beta - v^T b^1, \ldots, \beta - v^T b^m \right\}$ is maximized.

This yields the following optimization model:

$$\begin{array}{lllll} \mathrm{PCP}: & \mathrm{maximize}_{v,\beta,\delta} & \delta & & \\ & \mathrm{s.t.} & v^T a^i - \beta & \geq & \delta, \quad i = 1, \ldots, k \\ & & \beta - v^T b^i & \geq & \delta, \quad i = 1, \ldots, m \\ & & \|v\| & = & 1, \\ & & v \in \Re^n, \beta \in \Re & & \end{array}$$

Now notice that PCP is *not* a convex optimization problem, due to the presence of the constraint "$\|v\| = 1$".

2.3 Convex Reformulation of PCP

To obtain a convex optimization problem equivalent to PCP, we perform the following transformation of variables:

$$x = \frac{v}{\delta} \quad , \quad \alpha = \frac{\beta}{\delta} \ .$$

Then notice that $\delta = \frac{\|v\|}{\|x\|} = \frac{1}{\|x\|}$, and so maximizing δ is equivalent to maximizing $\frac{1}{\|x\|}$, which is equivalent to minimizing $\|x\|$. This yields the following reformulation of PCP:

$$\begin{array}{llll} \mathrm{minimize}_{x,\alpha} & \|x\| & & \\ \mathrm{s.t.} & & & \\ & x^T a^i - \alpha & \geq & 1, \quad i = 1, \ldots, k \\ & \alpha - x^T b^i & \geq & 1, \quad i = 1, \ldots, m \\ & x \in \Re^n, \alpha \in \Re & & \end{array}$$

Since the function $f(x) = \frac{1}{2}\|x\|^2 = \frac{1}{2}x^T x$ is monotone in $\|x\|$ we have that the point that minimizes the function $\|x\|$ also minimizes the function $\frac{1}{2}x^T x$. We therefore write the pattern classification problem in the following form:

$$\begin{array}{llrcll} \mathrm{CQP}: & \mathrm{minimize}_{x,\alpha} & \frac{1}{2}x^T x & & & \\ & \mathrm{s.t.} & x^T a^i - \alpha & \geq & 1, & i = 1, \ldots, k \\ & & \alpha - x^T b^i & \geq & 1, & i = 1, \ldots, m \\ & & x \in \Re^n, \alpha \in \Re & & & \end{array}$$

Notice that CQP is a convex program with a differentiable objective function. We can solve CQP for the optimal $x = x^*$ and $\alpha = \alpha^*$, and compute the optimal solution of PCP as:

$$v^* = \frac{x^*}{\|x^*\|} \quad , \quad \beta^* = \frac{\alpha^*}{\|x^*\|} \quad , \quad \delta^* = \frac{1}{\|x^*\|} \ .$$

Problem CQP is a convex quadratic optimization problem in $n+1$ variables, with $k+m$ linear inequality constraints. There are very many software packages that are able to solve quadratic programs such as CQP. However, one difficulty that might be encountered in practice is that the number of points that are used to define the linear decision rule might be very large (say, $k + m \geq 1,000,000$ or more). This can cause the model to become too large to solve without developing special-purpose algorithms.

3 Constructing the Dual of CQP

As it turns out, the Lagrange dual of CQP yields much insight into the structure of optimal solution of CQP, and also suggests several algorithmic approaches for solving the problem. In this section we derive the dual of CQP.

We start by creating the Lagrangian function. Assign a nonnegative multiplier λ_i to each constraint "$1 - x^T a^i + \alpha \leq 0$" for $i = 1, \ldots, k$ and a nonnegative multiplier γ_i to each constraint "$1 + x^T b^i - \alpha \leq 0$" for $i = 1, \ldots, m$. Think of the λ_i as forming the vector $\lambda = (\lambda_1, \ldots, \lambda_k)$ and the γ_i as forming the vector $\gamma = (\gamma_1, \ldots, \gamma_m)$. The Lagrangian then is:

$$\begin{aligned} L(x,\alpha,\lambda,\gamma) &= \frac{1}{2}x^T x + \sum_{i=1}^{k} \lambda_i (1 - x^T a^i + \alpha) + \sum_{j=1}^{m} \gamma_j (1 + x^T b^j - \alpha) \\ &= \frac{1}{2}x^T x - x^T \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right) + \left(\sum_{i=1}^{k} \lambda_i - \sum_{j=1}^{m} \gamma_j \right) \alpha + \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j \end{aligned}$$

We next create the dual function $L^*(\lambda, \gamma)$:

$$L^*(\lambda,\gamma) = \text{minimum}_{x,\alpha} L(x,\alpha,\lambda,\gamma) \ .$$

In solving this unconstrained minimization problem, we observe that $L(x, \alpha, \lambda, \gamma)$ is a convex function of x and α for fixed values of λ and γ. Therefore $L(x, \alpha, \lambda, \gamma)$ is minimized over x and α when

$$\begin{aligned} \nabla L_x(x,\alpha,\lambda,\gamma) &= 0 \\ \nabla L_\alpha(x,\alpha,\lambda,\gamma) &= 0 \ . \end{aligned}$$

The first condition above states that the value of x will be:

$$x = \sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \ , \tag{1}$$

and the second condition above states that (λ, γ) must satisfy:

$$\sum_{i=1}^{k} \lambda_i - \sum_{j=1}^{m} \gamma_j = 0 \ . \tag{2}$$

Substituting (1) and (2) back into $L(x, \alpha, \lambda, \gamma)$ yields:

$$L^*(\lambda, \gamma) = \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j - \frac{1}{2} \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right)^T \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right)$$

where (λ, γ) must satisfy (2).

Finally, the dual problem problem is constructed as:

$$\begin{array}{lll} \text{D1}: & \text{maximum}_{\lambda,\gamma} & \sum\limits_{i=1}^{k} \lambda_i + \sum\limits_{j=1}^{m} \gamma_j - \frac{1}{2} \left(\sum\limits_{i=1}^{k} \lambda_i a^i - \sum\limits_{j=1}^{m} \gamma_j b^j \right)^T \left(\sum\limits_{i=1}^{k} \lambda_i a^i - \sum\limits_{j=1}^{m} \gamma_j b^j \right) \\ & \text{s.t.} & \sum\limits_{i=1}^{k} \lambda_i - \sum\limits_{j=1}^{m} \gamma_j = 0 \\ & & \lambda \geq 0, \gamma \geq 0 \\ & & \lambda \in \Re^k, \gamma \in \Re^m \ . \end{array}$$

By utilizing (1), we can re-write this last formulation with the extra variables x as:

$$\begin{array}{lll} \text{D2}: & \text{maximum}_{\lambda,\gamma,x} & \sum\limits_{i=1}^{k} \lambda_i + \sum\limits_{j=1}^{m} \gamma_j - \frac{1}{2} x^T x \\ & \text{s.t.} & x - \left(\sum\limits_{i=1}^{k} \lambda_i a^i - \sum\limits_{j=1}^{m} \gamma_j b^j \right) = 0 \\ & & \sum\limits_{i=1}^{k} \lambda_i - \sum\limits_{j=1}^{m} \gamma_j = 0 \\ & & \lambda \geq 0, \gamma \geq 0 \\ & & \lambda \in \Re^k, \gamma \in \Re^m \ . \end{array}$$

Comparing D1 and D2, it might seem wiser to use D1 as it uses fewer variables and fewer constraints. But notice one disadvantage of D1. The quadratic portion of the objective function in D1, expressed as a quadratic form of the variables (λ, γ) will look something like:

$$(\lambda, \gamma)^T Q (\lambda, \gamma)$$

where Q has dimension $(m+k) \times (m+k)$. The size of this matrix grows with the square of $(m+k)$, which is bad. In contrast, the growth in the problem size of D2 is only linear in $(m+k)$.

Our next result shows that if we have an optimal solution to the dual problem D1 (or D2, for that matter), then we can easily write down an optimal solution to the original problem CQP.

Property 1 *If $(\lambda^*, \gamma^*) \neq 0$ is an optimal solution of D1, then:*

$$\begin{aligned} x^* &= \sum_{i=1}^{k} \lambda_i^* a^i - \sum_{j=1}^{m} \gamma_j^* b^j && (3) \\ \alpha^* &= \frac{1}{2\sum_{i=1}^{k} \lambda_i^*} (x^*)^T \left(\sum_{i=1}^{k} \lambda_i^* a^i + \sum_{j=1}^{m} \gamma_j^* b^j \right) && (4) \end{aligned}$$

is an optimal solution to CQP.

We will prove this property in the next section, as a consequence of the Karush-Kuhn-Tucker optimality conditions for CQP.

4 The Karush-Kuhn-Tucker Conditions for CQP

The problem CQP is:

$$
\begin{array}{llrcll}
\text{CQP}: & \text{minimize}_{x,\alpha} & \frac{1}{2}x^Tx & & & \\
& \text{s.t.} & x^Ta^i - \alpha & \geq & 1, & i = 1, \dots, k \\
& & \alpha - x^Tb^i & \geq & 1, & i = 1, \dots, m \\
& & x \in \Re^n, \alpha \in \Re & & &
\end{array}
$$

The KKT conditions for this problem are:

- Primal feasibility:

$$
\begin{array}{ll}
x^Ta^i - \alpha \geq 1, & i = 1, \dots, k \\
\alpha - x^Tb^j \geq 1, & j = 1, \dots, m \\
x \in \Re^n, \alpha \in \Re &
\end{array}
$$

- The Gradient Condition:

$$x - \sum_{i=1}^{k} \lambda_i a^i + \sum_{j=1}^{m} \gamma_j b^j = 0$$

$$\sum_{i=1}^{k} \lambda_i - \sum_{j=1}^{m} \gamma_j = 0$$

- Nonnegativity of Multipliers:

$$\lambda \geq 0 \quad , \quad \gamma \geq 0$$

- Complementarity:

$$\lambda_i(1 - x^T a^i + \alpha) = 0 \quad i = 1, \ldots, k$$

$$\gamma_j(1 + x^T b^j - \alpha) = 0 \quad j = 1, \ldots, m$$

5 Insights from Duality and the KKT Conditions

5.1 The KKT Conditions Yield Primal and Dual Solutions

Since CQP is the minimization of a convex function under linear inequality constraints, the KKT conditions are necessary and sufficient to characterize an optimal solution.

Property 2 *If $(x, \alpha, \lambda, \gamma)$ satisfies the KKT conditions, then (x, α) is an optimal solution to CQP and (λ, γ) is an optimal solution to D1.*

Proof: From the primal feasibility conditions, and the nonnegativity conditions on (λ, γ), we have that (x, α) and (λ, γ) are primal and dual feasible, respectively. We therefore need to show equality of the primal and dual objective functions to prove optimality.

If we call $z(x, \alpha)$ the objective function of the primal problem evaluated at (x, α) and $v(\lambda, \gamma)$ the objective function of the dual problem evaluated at (λ, γ) we have:

$$z(x, \alpha) = \frac{1}{2} x^T x$$

and

$$v(\lambda, \gamma) = \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j - \frac{1}{2} \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right)^T \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right) .$$

However, we also have from the gradient conditions that:

$$x = \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right) \ .$$

Substituting this in above yields:

$$\begin{aligned}
v(\lambda,\gamma) - z(x,\alpha) &= \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j - x^T x \\
&= \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j - x^T \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right) \\
&= \sum_{i=1}^{k} \lambda_i \left(1 - x^T a^i\right) + \sum_{j=1}^{m} \gamma_j \left(1 + x^T b^j\right) + \alpha \left(\sum_{i=1}^{k} \lambda_i - \sum_{j=1}^{m} \gamma_j \right) \\
&= \sum_{i=1}^{k} \lambda_i \left(1 - x^T a^i + \alpha\right) + \sum_{j=1}^{m} \gamma_j \left(1 + x^T b^j - \alpha\right) \\
&= 0
\end{aligned}$$

and so (x,α) and (λ,γ) are optimal for their respective problems. ■

5.2 Nonzero values of x, λ, and γ

If $(x,\alpha,\lambda,\gamma)$ satisfy the KKT conditions, then $x \neq 0$ from the primal feasibility constraints. This in turn implies that $\lambda \neq 0$ and $\gamma \neq 0$ from the gradient conditions.

5.3 Complementarity

The complementarity conditions imply that the dual variables λ_i and γ_j are only positive if the corresponding primal constraint is tight:

$$\lambda_i > 0 \Rightarrow \quad (1 - x^T a^i + \alpha) = 0$$

$$\gamma_j > 0 \Rightarrow \quad (1 + x^T b^j - \alpha) = 0$$

Equality in the primal constraint means that the corresponding point, a^i or b^j, is at the minimum distance from the hyperplane. In combination with the observation above that $\lambda \neq 0$ and $\gamma \neq 0$, this implies that the optimal hyperplane will have points of both classes at the minimum distance.

5.4 A Further Geometric Insight

Consider the example of Figure 2. In Figure 2, the hyperplane is determined by the three points that are at the minimum distance from it. Since the points we are separating are not likely to exhibit collinearity, we can conclude in general that most of the dual variables will be zero at the optimal solution.

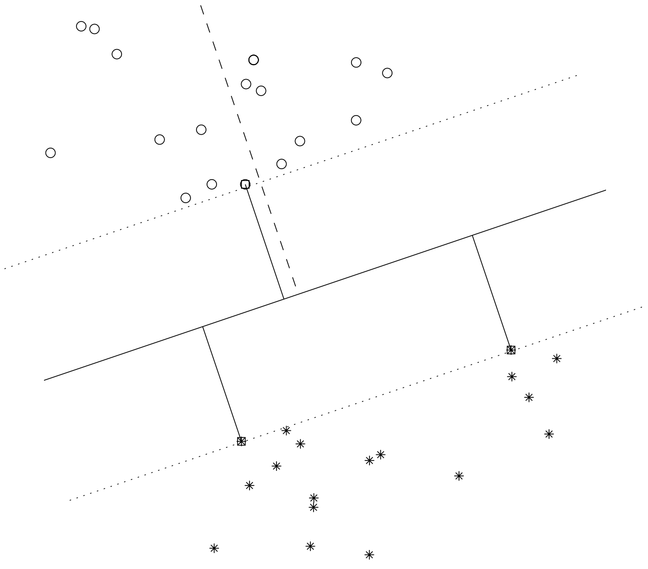

Figure 2: A separating hyperplane determined by three points.

More generally, we would expect that at most $n+1$ points will lie at the minimum distance from the hyperplane. Therefore, we would expect that all but at most $n+1$ dual variables will be zero at the optimal solution.

5.5 The Geometry of the Normal Vector

Consider the example shown in Figure 3. In Figure 3, the points corresponding to active primal constraints are labeled a^1, a^2, a^3, b^1, b^2.

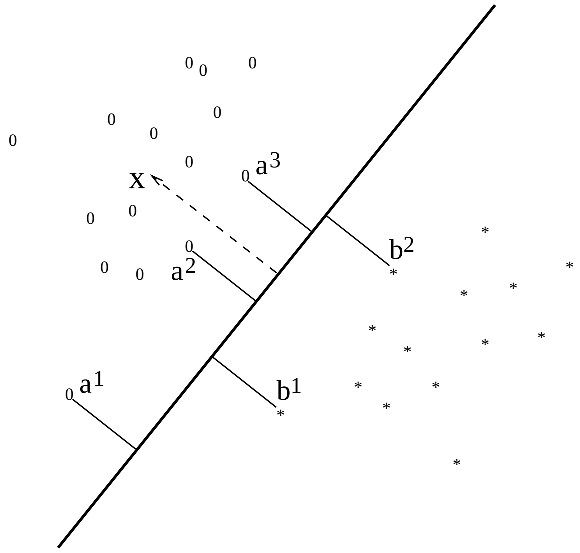

Figure 3: Separating hyperplane and points with tight primal constraints.

In this example, we see that $x^T(a^1 - a^2) = x^T a^1 - x^T a^2 = (1 + \alpha) - (1 + \alpha) = 0$. Here we only used the fact that $1 - x^T a^i + \alpha = 0$.

In general we have the following property:

Property 3 *The normal vector x to the separating hyperplane is orthogonal to any difference between points of the same class whose primal constraints are active.*

Proof: Without loss of generality, we can consider the points $a^1, \ldots, a^{\hat{\imath}}$ and $b^1, \ldots, b^{\hat{\jmath}}$ to be the points corresponding to tight primal constraints. This means that $x^T a^i = 1 + \alpha$, $i = 1, \ldots, \hat{\imath}$, and also that $x^T b^j = \alpha - 1$, $j = 1, \ldots, \hat{\jmath}$. From this it is clear that $x^T(a^i - a^j) = 0$, $i, j = 1, \ldots, \hat{\imath}$ and $x^T(b^i - b^j) = 0$, $i, j = 1, \ldots, \hat{\jmath}$. ■

5.6 More Geometry of the Normal Vector

From the gradient conditions, we have:

$$\begin{aligned} x &= \sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \\ &= \sum_{i=1}^{\hat{\imath}} \lambda_i a^i - \sum_{j=1}^{\hat{\jmath}} \gamma_j b^j \end{aligned}$$

where we amend our notation so that the points a^i correspond to active constraints for $i = 1, \ldots, \hat{\imath}$ and the points b^j correspond to active constraints for $j = 1, \ldots, \hat{\jmath}$.

If we set $\delta = \sum_{i=1}^{\hat{\imath}} \lambda_i = \sum_{j=1}^{\hat{\jmath}} \gamma_j$, we have:

$$x = \delta \left(\sum_{i=1}^{\hat{\imath}} \frac{\lambda_i}{\delta} a^i - \sum_{j=1}^{\hat{\jmath}} \frac{\gamma_j}{\delta} b^j \right)$$

From this last equation we see that x is the scaled difference between a convex combination of $a^1, \ldots, a^{\hat{\imath}}$ and a convex combination of $b^1, \ldots, b^{\hat{\jmath}}$.

5.7 Proof of Property 1

We will now prove Proposition 1, which asserts that (3 - 4) yields an optimal solution to CQP given an optimal solution (λ, γ) of D1. We start by writing down the KKT optimality conditions for D1:

- Feasibility:

$$\sum_{i=1}^{k} \lambda_i - \sum_{j=1}^{m} \gamma_j = 0$$

$$\lambda \geq 0, \gamma \geq 0$$

$$\lambda \in \Re^k, \gamma \in \Re^m$$

- Gradient Conditions:

$$1 - (a^i)^T \left(\sum_{l=1}^{k} \lambda_l a^l - \sum_{j=1}^{m} \gamma_j b^j \right) + \alpha + \mu_i = 0 \qquad i = 1, \ldots, k$$

$$1 + (b^j)^T \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{l=1}^{m} \gamma_l b^l \right) - \alpha + \nu_j = 0 \qquad j = 1, \ldots, m$$

$$\mu \geq 0, \nu \geq 0$$

$$\mu \in \Re^k, \nu \in \Re^m$$

- Complementarity:

$$\lambda_i \mu_i = 0 \quad i = 1, \ldots, k$$

$$\gamma_j \nu_j = 0 \quad j = 1, \ldots, m$$

Now suppose that (λ, γ) is an optimal solution of D1, whereby (λ, γ) satisfy the above KKT conditions for some values of α, μ, ν. We will now show that the point (x, α) is optimal for CQP, where x is defined by (3), namely

$$x = \sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \ ,$$

and α arises in the KKT conditions. Substituting the expression for x in the gradient conditions, we obtain:

$$
\begin{aligned}
1 = x^T a^i - \alpha - \mu_i \leq x^T a^i - \alpha \qquad & i = 1, \dots, k, \\
1 = -x^T b^j + \alpha - \nu_j \leq -x^T b^j + \alpha \quad & j = 1, \dots, m.
\end{aligned}
$$

Therefore (x, α) is feasible for CQP. Now let us compare the objective function values of (x, α) in the primal and (λ, γ) in the dual. The primal objective function value is:

$$z = \frac{1}{2} x^T x \ ,$$

and the dual objective function value is:

$$v = \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j - \frac{1}{2} \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right)^T \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right) \ .$$

Substituting in the equation:

$$x = \sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j$$

and taking differences, we compute the duality gap between these two solutions to be:

$$
\begin{aligned}
v - z &= \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j - x^T x \\
&= \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j - x^T \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right) \\
&= \sum_{i=1}^{k} \lambda_i \left(1 - x^T a^i \right) + \sum_{j=1}^{m} \gamma_j \left(1 + x^T b^j \right) + \alpha \left(\sum_{i=1}^{k} \lambda_i - \sum_{j=1}^{m} \gamma_j \right)
\end{aligned}
$$

$$\begin{aligned}
&= \sum_{i=1}^{k} \lambda_i \left(1 - x^T a^i + \alpha\right) + \sum_{j=1}^{m} \gamma_j \left(1 + x^T b^j - \alpha\right) \\
&= -\sum_{i=1}^{k} \lambda_i \mu_i - \sum_{j=1}^{m} \gamma_j \nu_j \\
&= 0
\end{aligned}$$

which demonstrates that (x, α) must therefore be an optimal primal solution.

To finish the proof we must validate the expression for α in (4). If we multiply the first set of expressions of the gradient conditions by λ_i and the second set by $-\gamma_j$ we obtain:

$$\begin{aligned}
0 = \lambda_i \left(1 - x^T a^i + \alpha + \mu_i\right) &= \lambda_i - \lambda_i x^T a^i + \lambda_i \alpha \qquad i = 1, \ldots, k \\
0 = -\gamma_j \left(1 + x^T b^j - \alpha + \nu_j\right) &= -\gamma_j - \gamma_j x^T b^j + \gamma_j \alpha \qquad j = 1, \ldots, m
\end{aligned}$$

Summing these, we obtain:

$$\begin{aligned}
0 &= \sum_{i=1}^{k} \lambda_i - \sum_{j=1}^{m} \gamma_j - x^T \left(\sum_{i=1}^{k} \lambda_i a^i + \sum_{j=1}^{m} \gamma_j b^j \right) + \alpha \left(\sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{m} \gamma_j \right) \\
&= -x^T \left(\sum_{i=1}^{k} \lambda_i a^i + \sum_{j=1}^{m} \gamma_j b^j \right) + 2 \sum_{i=1}^{k} \lambda_i \alpha
\end{aligned}$$

which implies that:

$$\alpha = \frac{1}{2 \sum_{i=1}^{k} \lambda_i} x^T \left(\sum_{i=1}^{k} \lambda_i a^i + \sum_{j=1}^{m} \gamma_j b^j \right)$$

$\blacksquare$

5.8 Final Comments

The fact that we can expect to have at most $n + 1$ dual variables different than zero in the optimal solution (out of a possible number which could be as high as $k + m$) is very important. It immediately suggests that we might want to develop solution methods that try to limit the number of dual variables that are different from zero at any iteration. Such algorithms are called "Active Set Methods", which we will visit in the next lecture.

6 Pattern Classification without strict Linear Separation

The are very many instances of the pattern classification problem where the observed data points do not give rise to a linear classifier, i.e., a hyperplane $H_{v,\beta}$ that separates $a^1, \ldots, a^k$ from $b^1, \ldots, b^m$. If we still want to construct a classification function that is based on a hyperplane, we will have to allow for some error or "noise" in the data. Alternatively, we can allow for separation via a non-linear function rather than a linear function.

6.1 Pattern Classification with Penalties

Consider the following optimization problem:

$$\begin{array}{lllll} \text{QP3}: & \text{minimize}_{x,\alpha,\xi} & \frac{1}{2}x^T x + C\sum_{i=1}^{k+m} \xi_i & & \\ & \text{s.t.} & x^T a^i - \alpha \geq 1 - \xi_i & i = 1, \ldots, k & \\ & & \alpha - x^T b^j \geq 1 - \xi_{k+j} & j = 1, \ldots, m & \\ & & \xi_i \geq 0 & i = 1, \ldots, k+m. & \end{array} \qquad (5)$$

In this model, the ξ_i variables allow the constraints to be violated, but at a cost of C, where we presume that C is a large number. Here we see that C represents the tradeoff between the competing objectives of producing a hyperplane that is far from the points $a^i, i = 1, \ldots, k$ and $b^i, i = 1, \ldots, m$, and that penalizes points for being close to the hyperplane.

The dual of this quadratic optimization problem turns out to be:

$$\begin{array}{lll} \text{D3}: & \text{maximize}_{\lambda,\gamma} & \sum_{i=1}^{k} \lambda_i + \sum_{j=1}^{k} \gamma_j - \frac{1}{2} \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right)^T \left(\sum_{i=1}^{k} \lambda_i a^i - \sum_{j=1}^{m} \gamma_j b^j \right) \\ & \text{s.t.} & \sum_{i=1}^{k} \lambda_i - \sum_{j=1}^{m} \gamma_j = 0 \\ & & \lambda_i \leq C \quad i = 1, \ldots, k \\ & & \gamma_j \leq C \quad j = 1, \ldots, m \\ & & \lambda \geq 0, \quad \gamma \geq 0 \ . \end{array} \tag{6}$$

In an analogous fashion to Property 1 stated earlier for the separable case, one can derive simple formulas to directly compute an optimal primal solution (x^*, α^*, ξ^*) from an optimal dual solution (λ^*, γ^*).

As an example of the above methodology, consider the pattern classification data shown in Figure 4.

Figure 5 and Figure 6 show solutions to this problem with different values of the penalty parameter $C = 10$ and $C = 100$, respectively.

6.2 Pattern Classification via Non-linear Mappings

Another way to separate sets of points that are not separable using a linear classifier (a hyperplane) is to create a non-linear transformation, usually to a higher-dimensional space, in such a way that the transformed points are separable by a hyperplane in the transformed space (or are nearly separable, with the aid of penalty terms).

Suppose we have on hand a mapping:

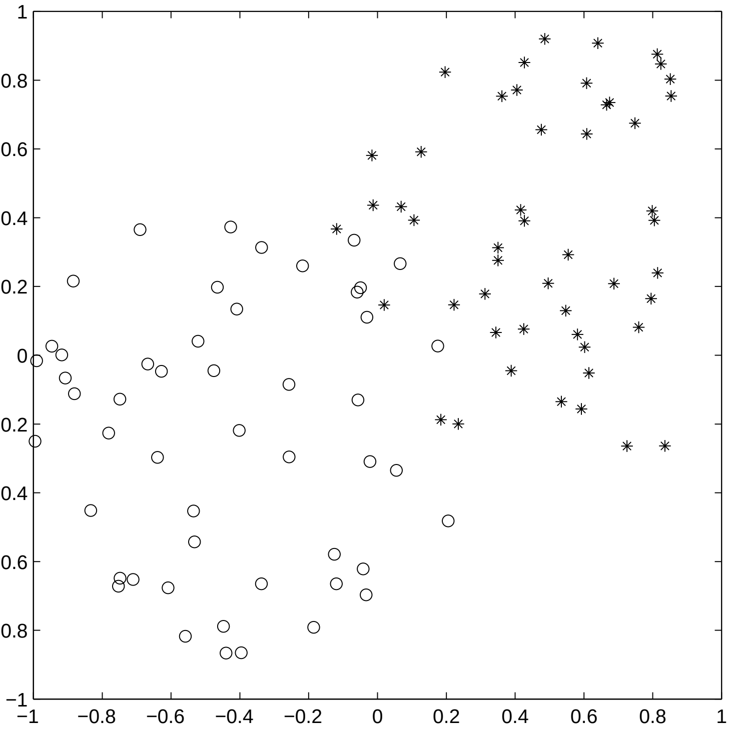

Figure 4: Pattern classification data that cannot be separated by a hyperplane.

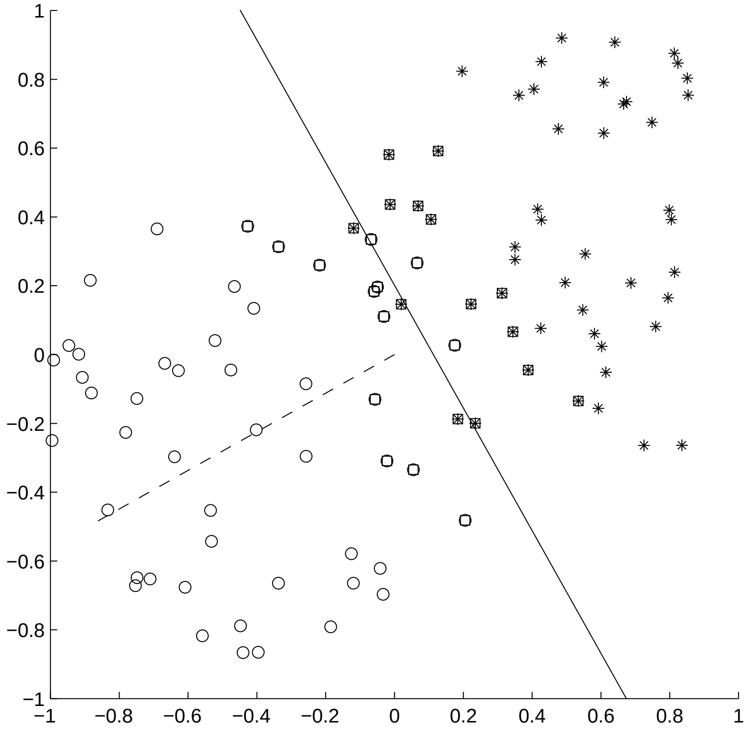

Figure 5: Solution to the problem with $C = 1.0$.

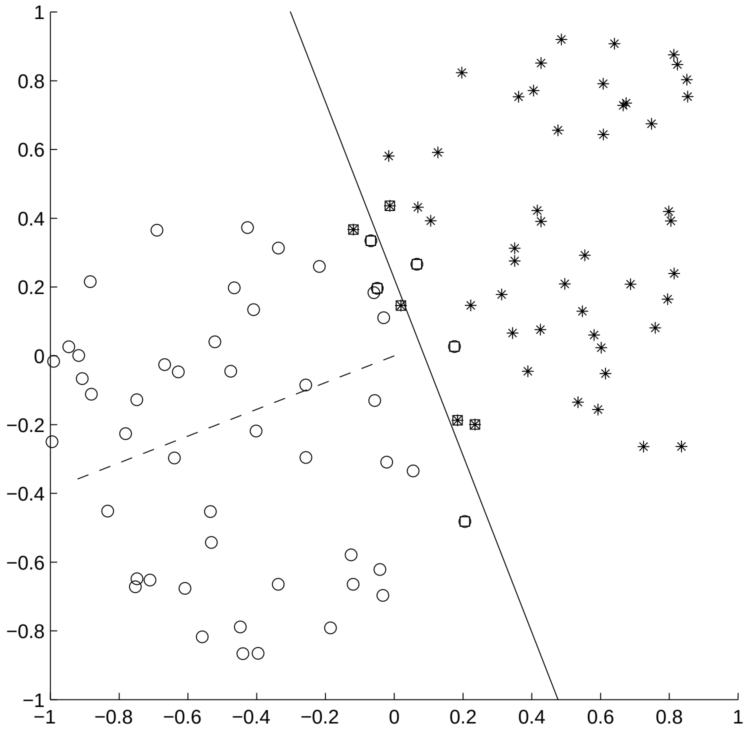

Figure 6: Solution to the problem with $C = 1,000.0$.

$$\phi(\cdot) : \Re^n \mapsto \Re^l$$

where one should think of l as satisfying $l >> n$. Under this mapping, we have:

$$\begin{array}{rcl} a^i & \mapsto & \tilde{a}^i = \phi(a^i), \quad i = 1, \ldots, k \\ \\ b^i & \mapsto & \tilde{b}^i = \phi(b^i), \quad i = 1, \ldots, m. \end{array}$$

We then solve for our separating hyperplane

$$H_{\tilde{v},\tilde{\beta}} = \left\{ \tilde{x} \in \Re^l \mid \tilde{v}^T \tilde{x} = \tilde{\beta} \right\}$$

via the methods described earlier. If we are given a new point c that we would like to classify, we compute

$$\tilde{c} = \phi(c)$$

and use the rule:

- If $\tilde{v}^T \tilde{c} > \tilde{\beta}$, then we declare that c has property P.
- If $\tilde{v}^T \tilde{c} < \tilde{\beta}$, then we declare that c does not have property P.

Quite often we do not have to explicitly work with the function $\phi(\cdot)$, as we now demonstrate. Consider the pattern classification problem shown in Figure 7. In Figure 7, the two classes of points are clearly not separable with a hyperplane. However, it does seem that there is a smooth function that would easily separate the two classes of points.

We can of course solve the linear separation problem (with penalties) for this problem. Figure 8 shows the linear classifier for this problem, computed using the penalty value $C = 100$. As Figure 8 clearly shows, this separator is clearly not adequate.

We might think of trying to find a quadratic function (as opposed to a linear function) that will separate our points. In this case we would seek to find values of:

$$Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{pmatrix} \quad , \quad q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \quad , \quad d$$

satisfying:

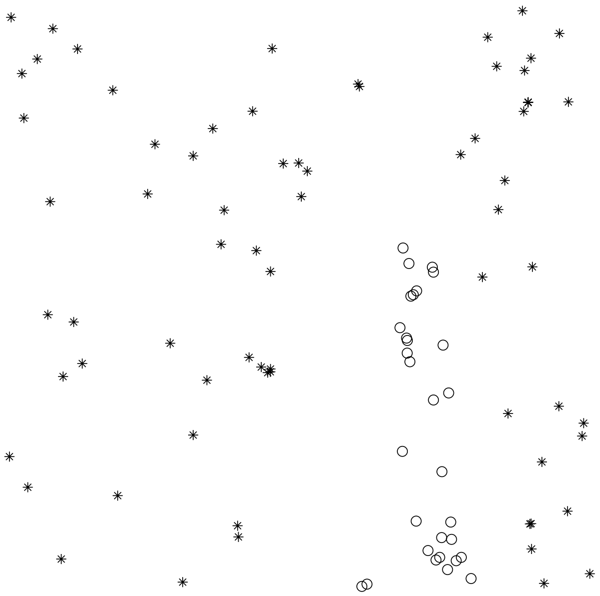

Figure 7: Two classes of points that can be separated by a nonlinear surface.

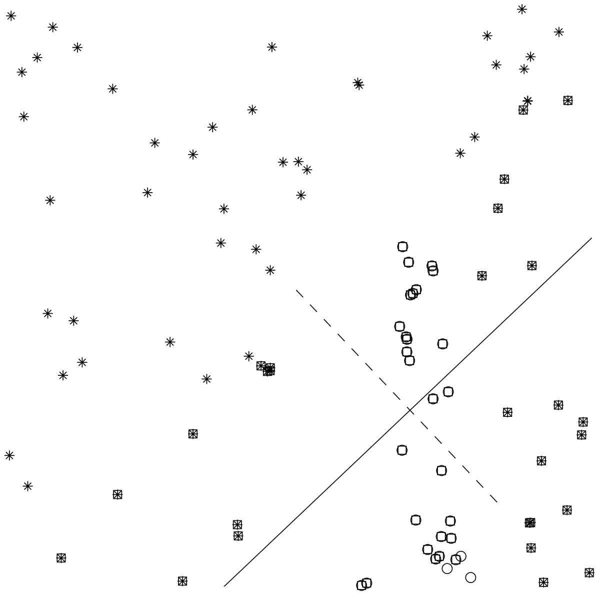

Figure 8: Illustration of a linear classifier for a non-separable case.

- $(a^i)^T Q a^i + q^T a^i + d > 0$ for all $i = 1, \ldots, k$
- $(b^i)^T Q b^i + q^T b^i + d < 0$ for all $i = 1, \ldots, m$

We would then use this data to classify any new point c as follows:

- If $c^T Qc + q^T c + d > 0$, then we declare that c has property P.
- If $c^T Qc + q^T c + d < 0$, then we declare that c does not have property P.

Although this problem seems much more complicated than linear classification, indeed it is really linear classification in a slightly higher-dimensional space! To see why this is true, let us look at our 2-dimensional example in detail. Let one of the a^i values be denoted as the data (a_1^i, a_2^i). Then

$$\begin{aligned}
(a^i)^T Q a^i + q^T a^i + d &= (a_1^i)^2 \times Q_{11} + 2a_1^i a_2^i \times Q_{12} + (a_2^i)^2 \times Q_{22} + a_1^i \times q_1 + a_2^i \times q_2 + d \\
&= \left((a_1^i)^2, 2a_1^i a_2^i, (a_2^i)^2, a_1^i, a_2^i\right)^T (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) + d \\
&= (\tilde{a}_1, \tilde{a}_2, \tilde{a}_3, \tilde{a}_4, \tilde{a}_5)^T (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) + d
\end{aligned}$$

where

$$\tilde{a} := \phi(a) = \phi(a_1, a_2) := (a_1^2, 2a_1 a_2, a_2^2, a_1, a_2) \ .$$

Notice here that this problem is one of linear classification in $\Re^5$. We therefore can solve this problem using the usual optimization formulation, but now stated in the higher-dimensional space. The problem we wish to solve then becomes:

$$\begin{array}{llcll}
\text{HPCP :} & \text{maximize}_{Q,q,d,\delta} \quad \delta & & & \\
\text{s.t.} & \left((a_1^i)^2, 2a_1^i a_2^i, (a_2^i)^2, a_1^i, a_2^i\right)^T (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) + d & \geq & \delta, & i = 1, \ldots, k \\
& -\left((b_1^i)^2, 2b_1^i b_2^i, (b_2^i)^2, b_1^i, b_2^i\right)^T (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) - d & \geq & \delta, & i = 1, \ldots, m \\
& \| (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) \| & = & 1, &
\end{array}$$

This problem then can be transformed into a convex quadratic program by using the transformation used in Section 2.3. If one solves the problem this way, the optimized quadratic separator turns out to be:

$$-24.723c_1^2 - 0.261c_1c_2 - 1.706c_2^2 + 14.438c_1 - 2.794c_2 - 0.163$$

Figure 9 shows the solution to this problem. Notice that the solution indeed separates the two classes of points.

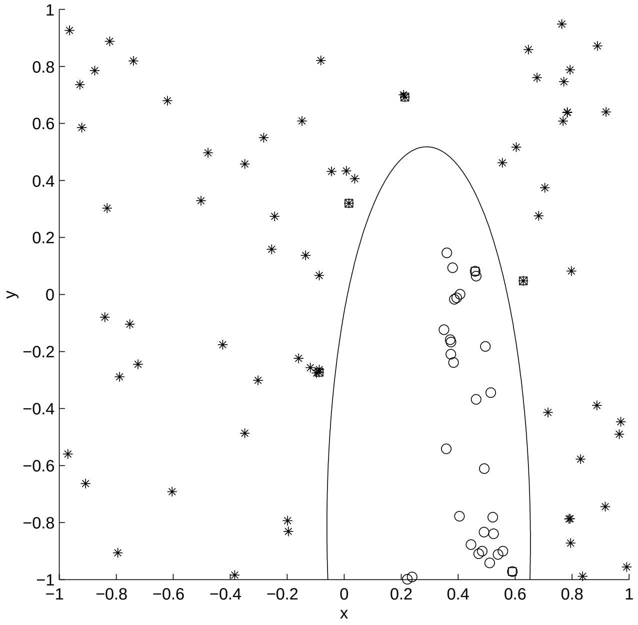

Figure 9: Illustration of separation by a quadratic separator.

6.3 Pattern Classification via Ellipsoids

As it turns out, the quadratic surface separator described in the previous section can be developed further. Recall the model:

$$
\begin{array}{lrcll}
\text{HPCP :} & \text{maximize}_{Q,q,d,\delta} \quad \delta & & & \\
\text{s.t.} & \left((a_1^i)^2, 2a_1^i a_2^i, (a_2^i)^2, a_1^i, a_2^i\right)^T (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) + d & \geq & \delta, & i = 1, \ldots, k \\
 & -\left((b_1^i)^2, 2b_1^i b_2^i, (b_2^i)^2, b_1^i, b_2^i\right)^T (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) - d & \geq & \delta, & i = 1, \ldots, m \\
 & \| (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) \| & = & 1, &
\end{array}
$$

Suppose that we would like the resulting quadratic surface to be an ellipsoid. This corresponds to requiring that the matrix

$$Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{pmatrix}$$

be an SPSD matrix. Suppose that we would like this ellipsoid to be as "round" as possible, which means that we would like the condition number

$$\kappa(Q) := \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)}$$

to be as small as possible. Then our problem becomes:

$$
\begin{array}{lrcll}
\text{RP :} & \text{minimize}_{Q,q,d} \quad \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)} & & & \\
\text{s.t.} & \left((a_1^i)^2, 2a_1^i a_2^i, (a_2^i)^2, a_1^i, a_2^i\right)^T (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) + d & \geq & 0, & i = 1, \ldots, k \\
 & -\left((b_1^i)^2, 2b_1^i b_2^i, (b_2^i)^2, b_1^i, b_2^i\right)^T (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) - d & \geq & 0, & i = 1, \ldots, m \\
 & \| (Q_{11}, Q_{12}, Q_{22}, q_1, q_2) \| & = & 1, & \\
 & Q \text{ is SPSD .} & & &
\end{array}
$$

As it turns out, this problem can be re-cast as a convex optimization problem, using the tools of the new field of *semi-definite programming*. Figure 10 shows an example of a solution to a pattern classification problem obtained by solving the problem RP.

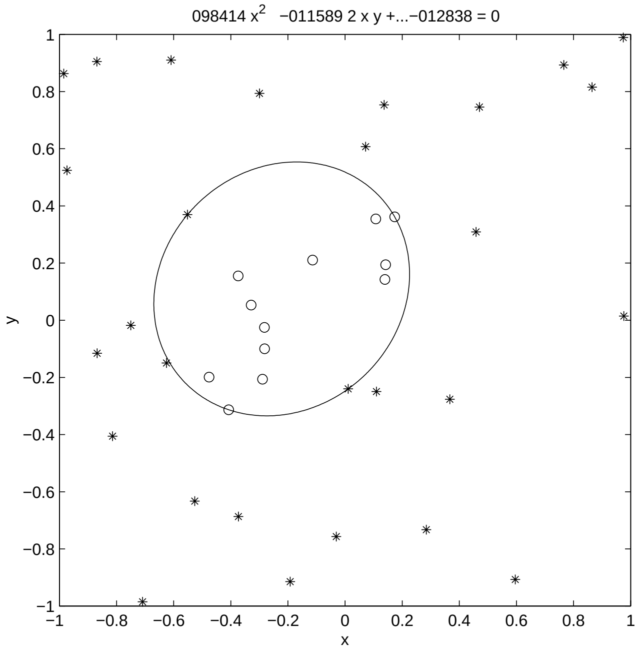

Figure 10: Illustration of an ellipsoidal separator using semi-definite programming.