Pattern separation via ellipsoids and conic programming

François Glineur

Faculté Polytechnique de Mons

9th Belgian-French-German Conference on Optimization

Namur, September 7, 1998

Objective

Machine learning

- Separation problem
- Classification

Mathematical programming

- Linear programming
- Interior-point methods
- SQL conic programming

Goal

Solve the *separation problem*
using *SQL conic programming*

How

Use *ellipsoids* to perform the separation

Outline

Conic programming

- Background
- SQL conic programming
- Modelling

Pattern separation

- Problem definition
- Describing an ellipsoid
- Four separation algorithms
- Exploitation

Computational experiments

- Comparison between our algorithms
- Comparison with other methods

Conclusions

Conic programming

Background

Operations research : Model $\Rightarrow$ *best* decision
Optimization : Minimize function of n variables
Mathematical programming : Add constraints

General mathematical program is *too difficult* : no algorithm with performance guarantee, existence of local minima

Linear programming : special case, very efficient algorithms, no local minima

Convex programming : Generalize linear programming while keeping its good properties

- No local minima
- Efficient interior-point methods
- Model much more problems than LP

Every convex program can be stated in a *conic* form

$$\text{(CONE)} \quad \min_{x \in \mathbb{R}^n} c^T x \quad \text{s.t.} \quad x \in C \cap (b + L)$$

C is a closed, pointed and solid convex cone, b and c are vectors and L is a linear subspace.

Interior-point methods solve (CONE) with ϵ relative accuracy using

$$\mathcal{O}(\sqrt{\nu} \log \frac{1}{\epsilon})$$

iterations (damped Newton steps) where ν depends only on the structure of C (Nesterov and Nemirovsky).

Duality

$$\text{(DUAL)} \quad \min_{y \in \mathbb{R}^n} b^T y \quad \text{s.t.} \quad y \in C^* \cap (c + L^T)$$

C^* is the dual cone. Also a conic program. Weak duality, strong duality with a *Slater* condition (otherwise : nonzero duality gap, optimum not attained, etc.)

Self-scaled cones

Special type of cone $\Rightarrow$ *long step primal-dual* interior point method, very efficient *in practice*

Classification : any self-scaled cone is the Cartesian product of *primitive* self-scaled cones

- The nonnegative reals $\mathbb{R}_+$ ($\Rightarrow$ LP)
- The second order cone $\mathbb{L}^n_+$

$$\mathbb{L}^n_+ = \{(r, x) \in \mathbb{R} \times \mathbb{R}^n \mid r \geq \|x\|\}$$

- The cone of positive semidefinite matrices with real entries $\mathbb{S}^n_+$ ($\Rightarrow$ Semidefinite Programming)
- The cone of positive semidefinite matrices with complex entries
- The cone of positive semidefinite matrices with quaternion entries
- The cone of 3×3 positive semidefinite matrices with octonion entries

We use only *real* self-scaled cones $\mathbb{R}^n_+$, $\mathbb{L}^n_+$ and $\mathbb{S}^n_+$: a conic program involving these cones is called a *SQL conic program*.

Modelling

- Includes linear and semidefinite programs
- Model nonlinear (convex) constraints like $xy \geq 1$, $E \in \mathbb{S}^n_+$, $\lambda_{max}(E) \leq 1$
- Minimize nonlinear (convex) objectives like $\frac{x^2}{y}$, $\|x\|$, $\lambda_{max}(E)$
- Handle free variables in a very natural way (using $\mathbb{L}^n_+$)
- But only *convex* programs (e.g. impossible to minimize $\frac{x}{y}$)

SQL conic programs $\equiv$ convex programs that are *easy to describe*.

Pattern separation

Problem definition

Pattern $\equiv$ vector combining n numerical attributes characterizing an object. Assume these objects are naturally grouped into c classes.

Objective : find a partition of $\mathbb{R}^n$ into c disjoint components corresponding to the classes.

Classification

Some objects grouped into classes and some unknown objects :

1. Separate the patterns of well-known objects $\equiv$ *learning* phase
2. Use that partition to classify the unknown objects $\equiv$ *generalization* phase

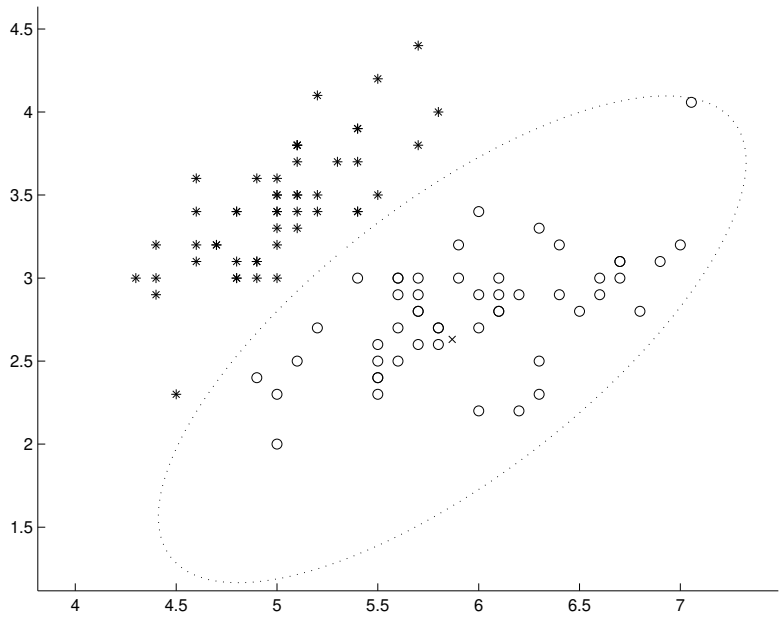
Examples : medical diagnosis, species identification, credit approval

Formulation

Reduce the problem with c classes to c independent problems involving 2 classes $\Rightarrow$ consider w.l.o.g. two classes : $A = \{a_i\}$ and $B = \{b_j\}$

Main idea : use *ellipsoids* for separation i.e. find $\mathcal{E}$ s.t.

$$a_i \in \mathcal{E} \ \forall i \text{ and } b_j \notin \mathcal{E} \ \forall j$$



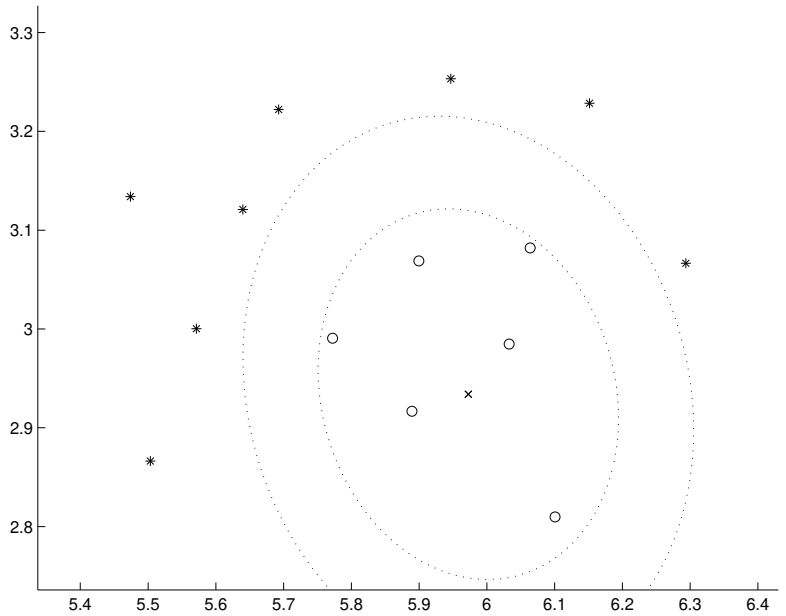
An ellipsoid $\equiv$ a center $c \in \mathbb{R}^n$ and a p.s.d. matrix $E \in \mathbb{S}^n_+$

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid (x-c)^T E(x-c) \leq 1\}$$

Separation ratio

We want the best possible separation ⇒ optimize the *separation ratio*

Pair of *similar* ellipsoids with the same center
Separation ratio $\rho \equiv$ ratio of sizes ($\rho = 1.5$ below)



Maximize ρ

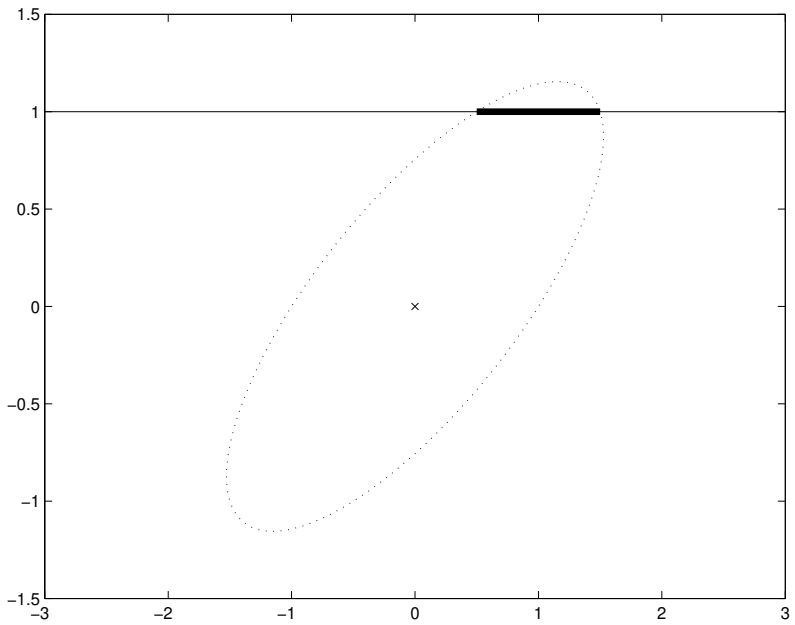
$$\max \rho \quad \text{s.t.} \quad \begin{cases} (a_i - c)^T E (a_i - c) \leq 1 \ \forall i \\ (b_j - c)^T E (b_j - c) \geq \rho^2 \ \forall j \\ E \in \mathbb{S}^n_+ \end{cases}$$

Two problems

1. ρ^2 is nonlinear $\Rightarrow$ let $k = \rho^2$, maximize k
2. $(x-c)^T E(x-c)$ is also nonlinear (nonconvex) $\Rightarrow$ *homogeneous* ellipsoid

Homogeneous ellipsoid

Ellipsoid in $\mathbb{R}^n \Leftrightarrow$ **Centered** Ellipsoid in $\mathbb{R}^{n+1}$



Replace $(x-c)^T E(x-c)$ with $(1,x)^T \tilde{E}(1,x)$

Optimize $\tilde{E} \in \mathbb{S}_+^{n+1}$

Recover optimal E, c with optimal $\tilde{E}$

Maximum separation ratio

$$\min -k \quad \text{s.t.} \quad \begin{cases} (1,a_i)^T \tilde{E} (1,a_i) \leq 1 \ \forall i \\ (1,b_j)^T \tilde{E} (1,b_j) \geq k \ \forall j \\ \tilde{E} \in \mathbb{S}_+^{n+1} \end{cases}$$

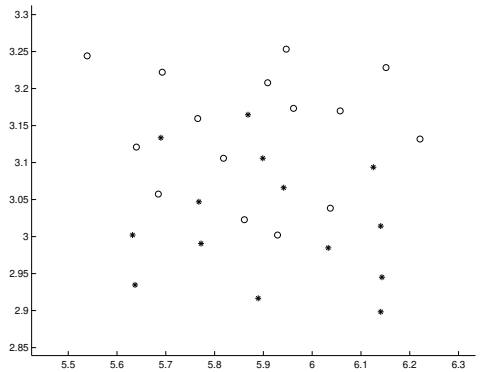
This is a SQL conic program

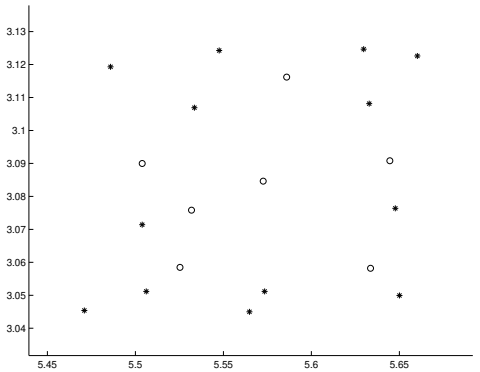
Properties

1. This program maximizes the separation ratio between a_i's and b_j's
2. Independence from the coordinate system

Drawback If there is no separating ellipsoid ?

- Convex hulls and A and B intersect
- Other cases





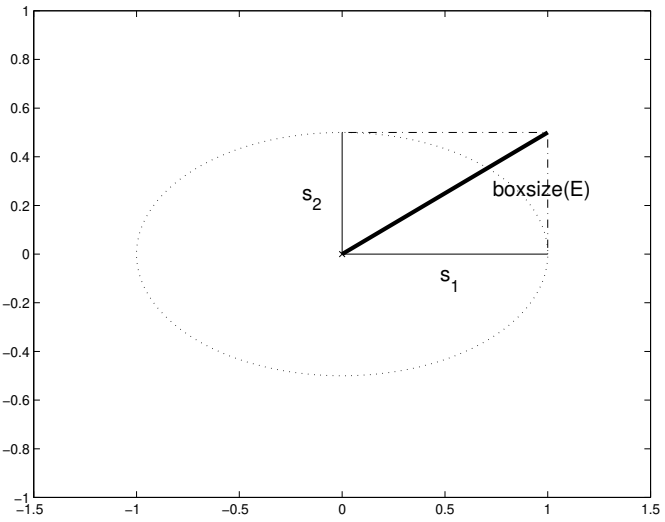
Minimum volume

Idea : find the ellipsoid containing the a_i's with minimum volume

Works when there is no separating ellipsoid

True volume of $\mathcal{E}$: difficult to model $\Rightarrow$ use a related measure using semi-axes s_i

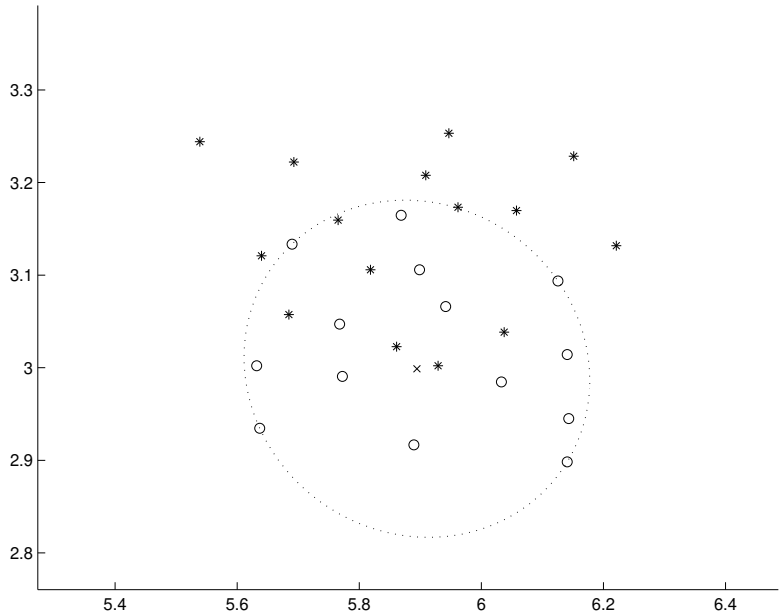
$$\text{boxsize(E)} = \sqrt{\sum_{i=1}^{n} \lambda_i(E)^{-1}} = \sqrt{\sum_{i=1}^{n} s_i^2}$$

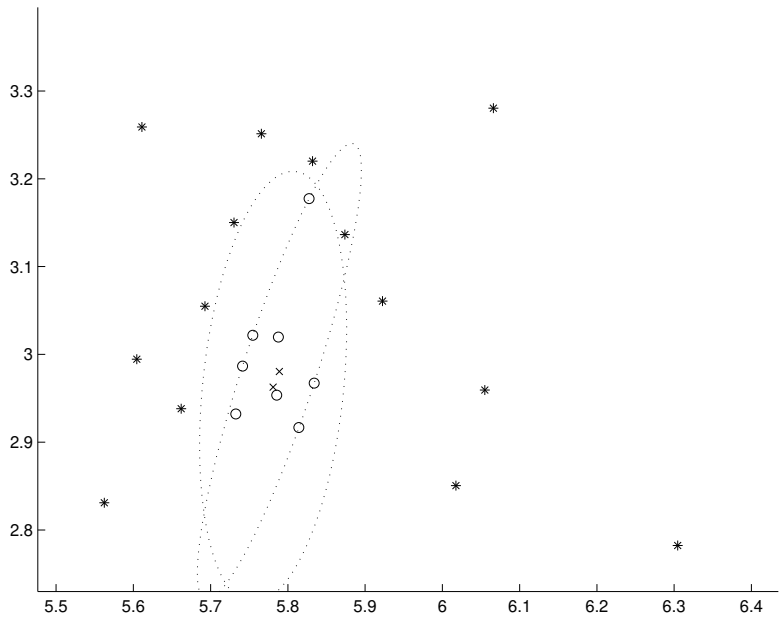


Model as a SQL conic program

Drawbacks

- Doesn't use the b_j's
- Scaling dependent


3.3
3.2
3.1
3
2.9
2.8
5.4
5.6
5.8
6
6.2
6.4


3.3
3.2
3.1
3
2.9
2.8
5.5
5.6
5.7
5.8
5.9
6
6.1
6.2
6.3
6.4

Maximum sum

Individual separation ratios ρ_j for each b_j

Maximum separation ratio $\equiv$ maximize the smallest ρ_j : *worst case* method

Idea : maximize *most of* the ρ_j's by maximizing $\sum \rho_j^2$

$$\min \; -\sum_j k_j \quad \text{s.t.} \quad \begin{cases} (1,a_i)^T \tilde{E}(1,a_i) \leq 1 \; \forall i \\ (1,b_j)^T \tilde{E}(1,b_j) = k_j \; \forall j \\ \tilde{E} \in \mathbb{S}_+^{n+1} \end{cases}$$

Properties

1. Works when there is no separating ellipsoid
2. Independence from the coordinate system

Drawback : May miss a separating ellipsoid

Explanation : Formulation $\equiv$ maximize the ρ_j's quadratic mean $\Rightarrow$ large ρ_j's dominate the objective $\Rightarrow$ small ρ_j's are not maximized

Solution : Combine the ρ_j's into an objective such that small values have *more influence* than large ones

Minimum squared sum

Idea : maximize the harmonic mean of the ρ_j's

Even better : maximize the harmonic mean of the ρ_j^4's

$$\min \sum_i k_i^2 \quad \text{s.t.} \quad \begin{cases} (1,a_i)^T \tilde{E} (1,a_i) = k_i \ \forall i \\ (1,b_j)^T \tilde{E} (1,b_j) \geq 1 \ \forall j \\ \tilde{E} \in \mathbb{S}_+^{n+1} \end{cases}$$

This is equivalent to

$$\min t \quad \text{s.t.} \quad \begin{cases} (1,a_i)^T \tilde{E} (1,a_i) = k_i \ \forall i \\ (1,b_j)^T \tilde{E} (1,b_j) \geq 1 \ \forall j \\ \tilde{E} \in \mathbb{S}_+^{n+1} \\ (t,k_1,\ldots,k_{n_a}) \in \mathbb{L}_+^{n_a+1} \end{cases}$$

This is a SQL conic program

Properties

1. Works when there is no separating ellipsoid
2. Independence from the coordinate system
3. Tries to maximize small ρ_j's

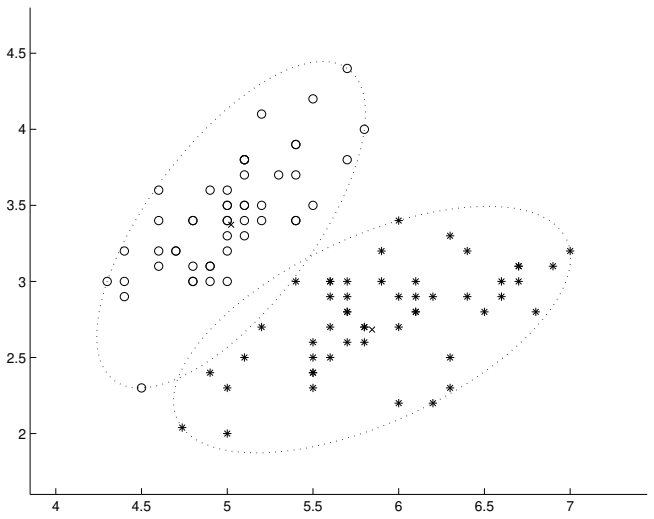
Exploitation method

To classify an unknown pattern p with $\mathcal{E}$ compute

$$p_{\mathcal{E}} = (p-c)^T E(p-c)$$

1. $p_{\mathcal{E}} \leq 1 \Rightarrow p$ belongs to the class of a_i's
2. $p_{\mathcal{E}} > 1 \Rightarrow p$ belongs to the class of b_j's

Not symmetric : permute a_i's and b_j's $\Rightarrow$ gives another ellipsoid $\mathcal{E}'$



Mixed strategy using both ellipsoids

1. $p_{\mathcal{E}} \leq p_{\mathcal{E}'} \Rightarrow p$ belongs to the class of a_i's
2. $p_{\mathcal{E}} > p_{\mathcal{E}'} \Rightarrow p$ belongs to the class of b_j's

When both ellipsoids agree, mixed strategy agrees too

Numerical experiments

Description

Implementation using MATLAB

- Short development cycles
- Availability of *SDPPack*, a package solving SQL conic programs

Experiments conducted on a standard PC

Cross-validation methodology

1. Divide the data set into two parts : a *learning set* and a *validation set*. Use only the learning set to compute the ellipsoids.
2. Classify the learning patterns $\Rightarrow$ *learning* error (how well the algorithm performed the separation)
3. Classify the validation patterns $\Rightarrow$ *testing* error (how well the algorithm generalizes)

Error = % of misclassified patterns.

Data sets

1. **Fisher's Iris**. Classify iris plants into three species (150 patterns, 4 characteristics)
2. **Wisconsin Breast Cancer**. Predict the benign or malignant nature of a breast tumor (683 patterns, 9 characteristics)
3. **Boston Housing**. Predict whether a housing value is above or below the median (596 patterns, 12 characteristics)
4. **Pima Indians Diabetes**. Predict whether a patient is showing signs of diabetes (768 patterns, 8 characteristics)

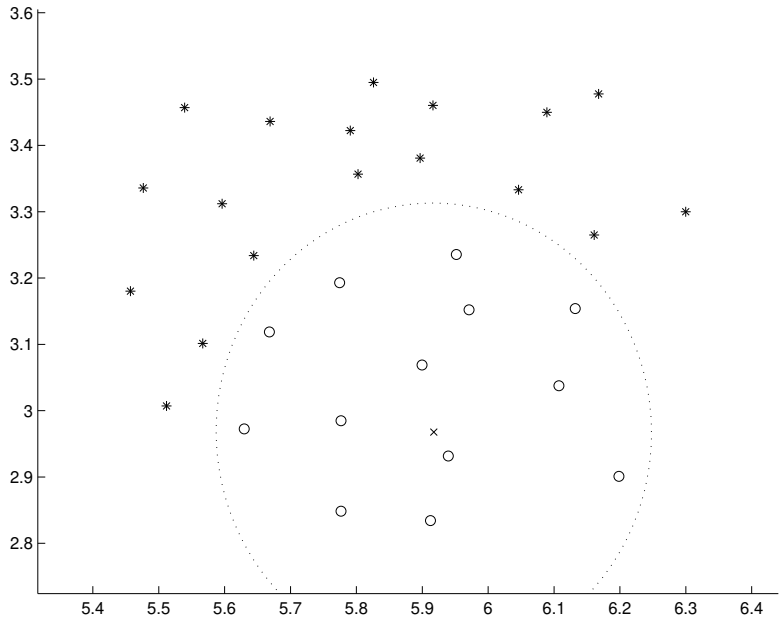
Sets from the Repository of Machine Learning Databases and Domain Theories maintained by the University of California at Irvine, widely used

We tested our four methods both with a 20% learning set and a 50% learning set.

Exploitation methods

Learning error Usually : one single ellipsoid strategy better, mixed strategy follows and other single ellipsoid strategy much worse

Explanation : one class has usually a *natural* ellipsoidal shape, the other is a *complement*



Testing error Mixed ellipsoid strategy always better ⇒ use it for all other tests

Comparison between our methods

We make the distinction between *easy* and *difficult* problems

1. Small problems, with a separating ellipsoid (Iris, Breast Cancer)

   Learning error *Maximum separation ratio* unbeatable, *Minimum volume* worst

   Testing error *Minimum volume* performs best, followed by *Maximum sum*

   Overlearning effect

2. Large problems, with no separating ellipsoid (Housing, Diabetes)

   Learning and testing error *Minimum squared sum* is a clear winner

   Explanation : the other methods cannot handle the absence of a separating ellipsoid

Comparison with other methods

Comparison with LAD (Logical Analysis of Data) and best other method found in the literature

	Best ellipsoidal		LAD	Best other
Training	20 %	50 %	50 %	Variable (% tr.)
Cancer	5.1 %	4.2 %	**3.1** %	3.8 % (80 %)
Housing	15.8 %	**12.4** %	16.0 %	16.8 % (80 %)
Diabetes	28.5 %	28.9 %	28.1 %	**24.1** % (75 %)

- Competitive error rates
- Best results on the Housing problem, even with 20 % learning
- 50 % learning not always better than 20 % learning ($\Rightarrow$ overlearning)
- Results with small learning set already acceptable (*quick* generalization)

Conclusions

SQL Conic programming

- Efficient interior-point methods
- Model many convex problems

Pattern separation

- Four separation methods
- Homogeneous ellipsoid

Computational experiments

- *Minimum volume* for easy problems
- *Minimum squared sum* for difficult problems
- Viable way of performing classification

Future research directions

- *Successive separations.* Use a hierarchical method ⇒ improved accuracy
- *Combining ellipsoids.* This idea is twofold
  1. Combine the ellipsoids from different methods ⇒ improved robustness
  2. Jackknife ⇒ lower sensibility to outliers, reduced overlearning effect.
- *Bi-ellipsoid optimization.* Optimize both ellipsoids from the mixed strategy at the same time, maximizing a relevant criterion ⇒ improved accuracy