Mathematical Programming in Machine Learning

Olvi L. Mangasarian

University of Wisconsin – Madison

www.cs.wisc.edu/~olvi

Based in part on joint work with:

Paul S. Bradley, Nick Street & Kristin P. Bennett

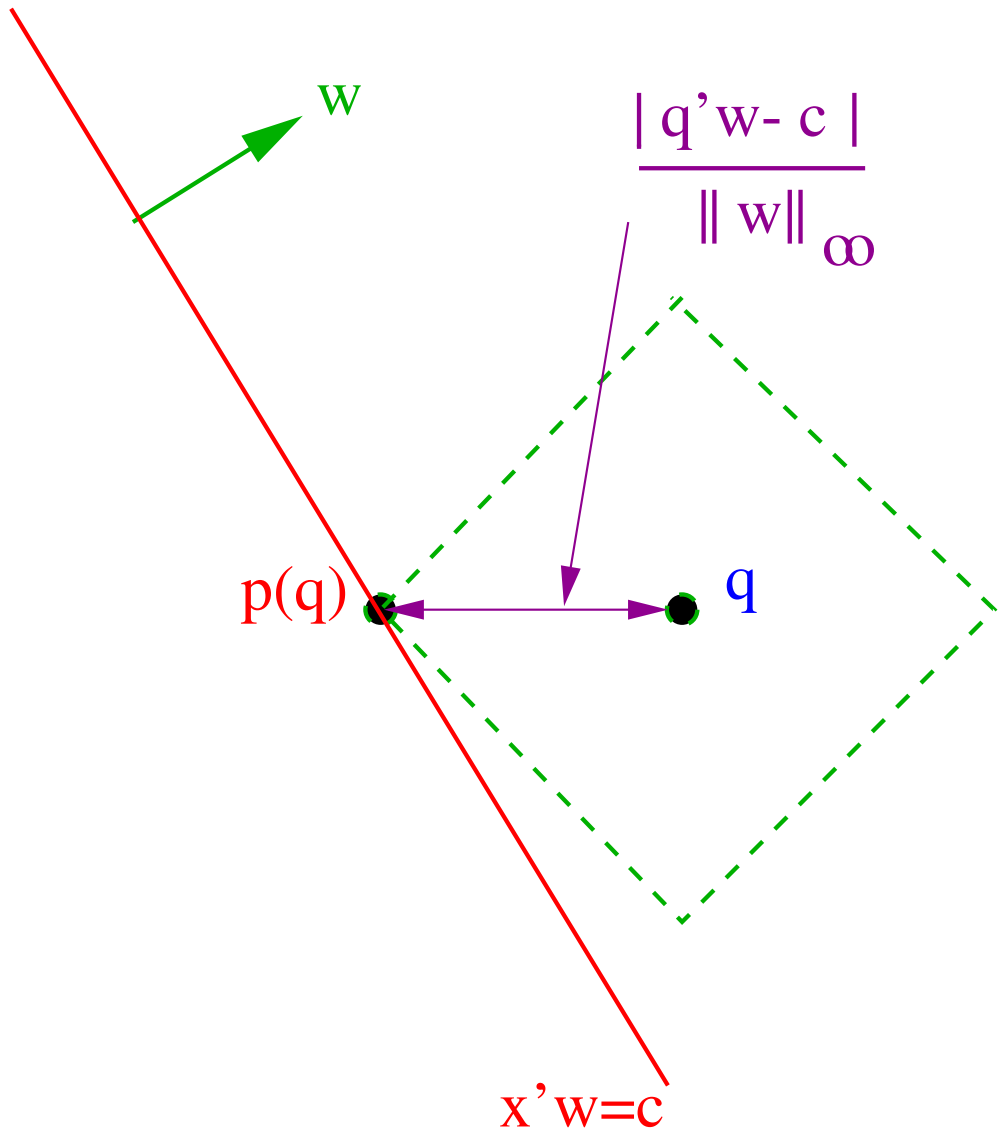
"Nothing happens in the universe that does not have ...... a maximum or minimum"

-Leonhard Euler 1707-1783

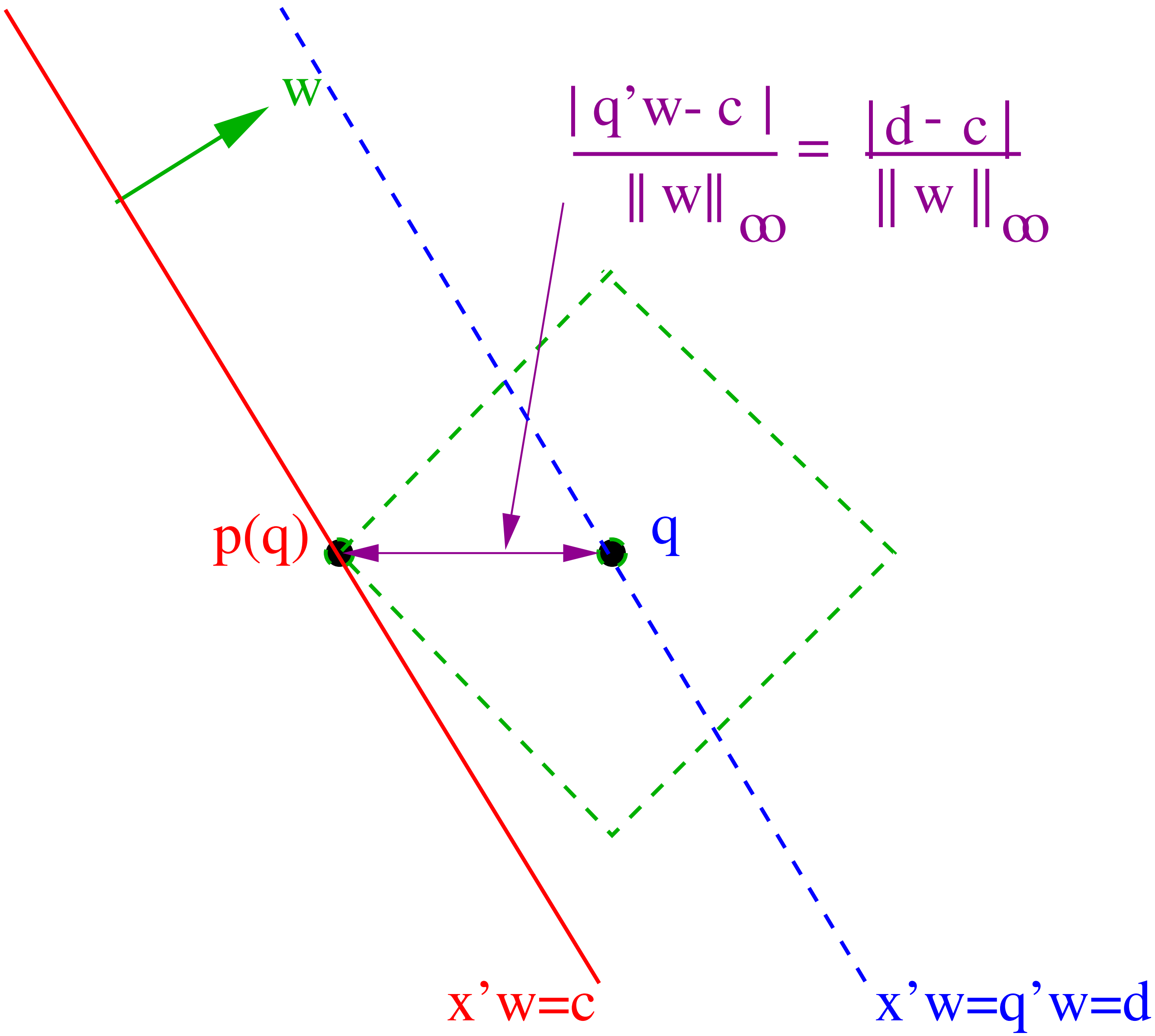
Outline

- Background material: arbitrary-norm distance to a plane
- Large margin & minimum support classifiers
  - 1-norm large margin linear & nonlinear classifiers 1965
  - 2-norm large margin classifiers 1968
  - Minimum support vector classifiers 1998
- Linear programming chunking for massive data discrimination 1998
- Feature selection via concave minimization 1998
- k-Median clustering via bilinear programming 1996
- Conclusion

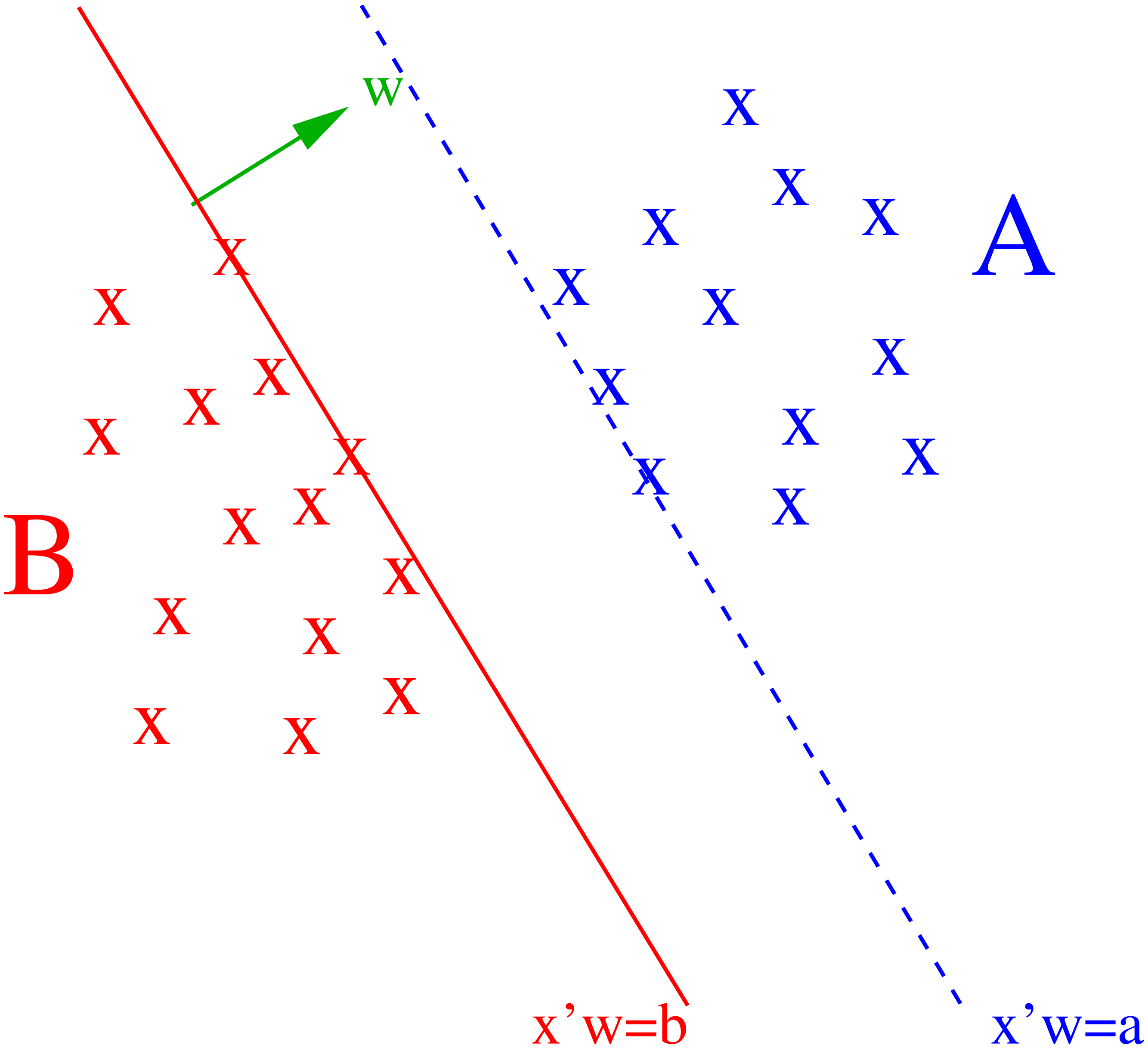
1-Norm Projection on a Plane



1-Norm Distance Between Parallel Planes



1-Norm Large Margin Classifier



1-Norm Large Margin Classifier

Linear Programming Formulation 1965

$$
\begin{array}{rrcrcrcrcl}
\displaystyle\max_{(w,a,b)\in R^n\times R\times R} & & & & & a & - & b & & \\
 & \text{s.t.} & Aw & - & ae & & & & \geq & 0, \\
 & & Bw & & & & - & be & \leq & 0, \\
-e & \leq & w & & & & & & \leq & e.
\end{array}
$$

Proposition: For linearly separable A and B each solution $(\bar{w}, \bar{a}, \bar{b})$ of the solvable linear program above, where e is a vector of ones, satisfies:

$$\|\bar{w}\|_\infty = 1, \quad \bar{a} - \bar{b} > 0.$$

Hence, $\bar{a} - \bar{b}$ is the maximum 1-norm margin between A and B.

1-Norm Large Margin Classifier

Nonlinearly Separable A & B

Linear Programming Formulation 1965

$$
\begin{array}{rrrrrrrr}
\max\limits_{(w,a,b)\in R^k\times R\times R} & & & & a & - & b & \\
 & \text{s.t.} & f(A)w & - & ae & & & \geq & 0, \\
 & & f(B)w & & & - & be & \leq & 0, \\
-e & \leq & w & & & & & \leq & e.
\end{array}
$$

Proposition: For nonlinearly separable A and B each solution $(\bar{w}, \bar{a}, \bar{b})$ of the solvable linear program above, where $f : R^n \longrightarrow R^k$, satisfies:

$$\|\bar{w}\|_\infty = 1, \ \ \bar{a} - \bar{b} > 0.$$

Hence, $\bar{a} - \bar{b}$ is the maximum 1-norm margin between $f(A)$ and $f(B)$.

2-Norm Large Margin Classifier
Mathematical Programming Formulation 1968

$$
\begin{array}{rrrrrrrrr}
 & \displaystyle\max_{(w,a,b)\in R^n\times R\times R} & & & & a & - & b & & \\
 & \text{s.t.} & Aw & - & ae & & & \geq & 0, \\
 & & Bw & & & - & be & \leq & 0, \\
-e & \leq & w & & & & & \leq & e, \\
 & & w'w & & & & & \geq & \frac{1}{2}.
\end{array}
$$

Proposition: For linearly inseparable A and B each solution $(\bar{w}, \bar{a}, \bar{b})$ of the solvable mathematical program above satisfies:

$$\bar{w}'\bar{w} = \frac{1}{2}, \quad \bar{a} - \bar{b} \leq 0.$$

Hence, $\sqrt{2}(\bar{b} - \bar{a})$ is the maximum 2-norm margin between parallel planes bounding A and B.

Minimum Support Vector Classifiers 1998

Two functions of an SVM:

- Separation by a nonlinear surface in original space of features
- Margin maximization between separating planes in higher dimensional space

Use:

- An indefinite kernel
  - for separation
- A forcing function f for support vector suppression as surrogate for margin maximization; f possibly defined by:
  - a positive definite kernel
  - some arbitrary norm

Kernels & Related Functions

- **Kernel** $K(A,B): R^{m\times n} \times R^{n\times \ell} \longrightarrow R^{m\times \ell}$

For column vectors x **and** y **in** R^n:

- $K(x', A')$ **is a row vector in** R^m
- $K(x', y)$ **is a real number**

Examples ($a \in R^m$, $\mu \in R$ **and** d **integer**):

- **Polynomial Kernel** $(\pm AA' \pm \mu aa')^d_\bullet$,
  where $[\cdot]^d_\bullet$ **denotes component-wise exponentiation as in MATLAB.**
- **Neural Network Kernel** $(\pm AA' \pm \mu aa')_{\bullet *}$,
  where $[\cdot]_{\bullet *}$ **denotes the step function:** $R \longrightarrow \{0,1\}$**component-wise.**

Separating Surface & Decision Function
Kernel Induced

- A *separating surface induced by an arbitrary kernel* K*:*

$$K(x', A')Du = \gamma$$

  - $A \in R^{m \times n}$ *represents* m *points in feature space* R^n
  - *Each row* A_i *of* A *belongs to class 1 or -1 depending on whether:*
    - D_{ii} *is 1 or -1, where* $D \in R^{m \times m}$ *is a given diagonal matrix of* ± 1.
  - $K(x', A') \in R^m$ *fixed by given kernel* K*, while parameters* $u \in R^m$ *and* $\gamma \in R$ *are determined by solving a mathematical program (QP or LP).*
  - u *is closely related to (but may be different from) dual multipliers of support vectors.*

- $x \in R^n$ *is classified in class 1 or -1 according to whether the* **decision function:**

$$(K(x', A')Du - \gamma)_*,$$

yields 1 or 0 respectively.

Minimum Support Vector Machine

K: An Arbitrary Kernel

$$
\begin{array}{lrcl}
\min\limits_{u,\gamma,y} & \nu e'y + f(u) & & \\
\text{s.t.} & D(K(A, A')Du - e\gamma) + y & \geq & e \\
 & y & \geq & 0.
\end{array}
$$

- f is some convex function on R^m, typically some norm or seminorm
  - All else is linear in (u, γ, y)
- ν is some positive parameter:
  - weights separation error $e'y$ versus suppression of parameter u
  - Two interpretations for u suppression:
    - Minimize the number of support vectors
    - Conventional interpretation: maximize margin between bounding parallel planes in R^k when f is induced by a positive definite kernel K.

MSVM: Above mathematical program and resulting decision function:

$$(K(x', A')Du - \gamma)_*$$

Robust Linear Programming Discrimination

K. P. Bennett & OLM 1992

$$
\begin{array}{rrcl}
\displaystyle\min_{w,\gamma,y} & e'y & & \\
\text{s.t.} & D(Aw - e\gamma) + y & \geq & e \\
 & y & \geq & 0.
\end{array}
$$

Special case of MSVM if we set $w = A'Du$ and $f(u) = 0$:

$$
\begin{array}{rrcl}
\displaystyle\min_{u,\gamma,y} & e'y & & \\
\text{s.t.} & D(AA'Du - e\gamma) + y & \geq & e \\
 & y & \geq & 0.
\end{array}
$$

Quadratic Examples of Specific $f(u)$

$$f(u) = \frac{1}{2}u'Hu$$

$$H = DK(A, A')D$$

K Positive Definite – Conventional SVM Dual Problem

$$\begin{array}{ll} \min\limits_{u \in R^m} & \frac{1}{2}u'DK(A, A')Du - e'u \\ \text{s.t.} & \begin{array}{rcl} e'Du & = & 0 \\ 0 \leq u & \leq & \nu e. \end{array} \end{array}$$

H=I

K Completely Arbitrary – Possibly Indefinite

Always Convex Dual Problem

$$\begin{array}{ll} \min\limits_{r \in R^m} & \frac{1}{2}r'DK(A, A')K(A, A')'Dr - e'r \\ \text{s.t.} & \begin{array}{rcl} e'Dr & = & 0 \\ 0 \leq r & \leq & \nu e. \end{array} \end{array}$$

$$u = DK(A, A')'Dr$$

Piecewise-Linear Examples of Specific $f(u)$
Linear Programs

$$f(u) = \|u\|_1$$

K Arbitrary – Linear Programming Primal

$$\begin{array}{rrcl} \displaystyle\min_{u,\gamma,y,s} & \nu e'y + e's & & \\ \text{s.t.} & D(K(A,A')Du - e\gamma) + y & \geq & e \\ & s \geq u & \geq & -s \\ & y & \geq & 0. \end{array}$$

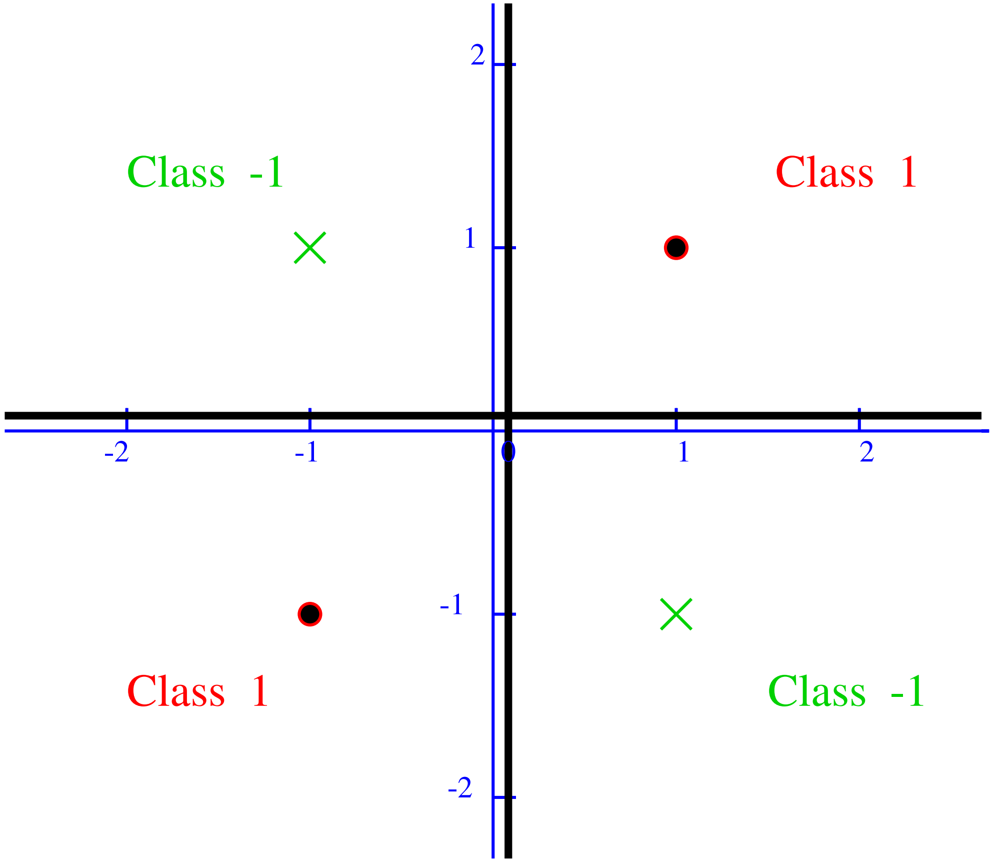
$$f(u) = \|K(A,A')Du\|_1$$

K Arbitrary – Linear Programming Primal

$$\begin{array}{rrcl} \displaystyle\min_{u,\gamma,y,s} & \nu e'y + e's & & \\ \text{s.t.} & D(K(A,A')Du - e\gamma) + y & \geq & e \\ & s \geq K(A,A')Du & \geq & -s \\ & y & \geq & 0. \end{array}$$

A Simple Example: XOR in R^2

PD Quadratic Kernel: $K(A, A') = (AA' + ee')^2_{\bullet}$



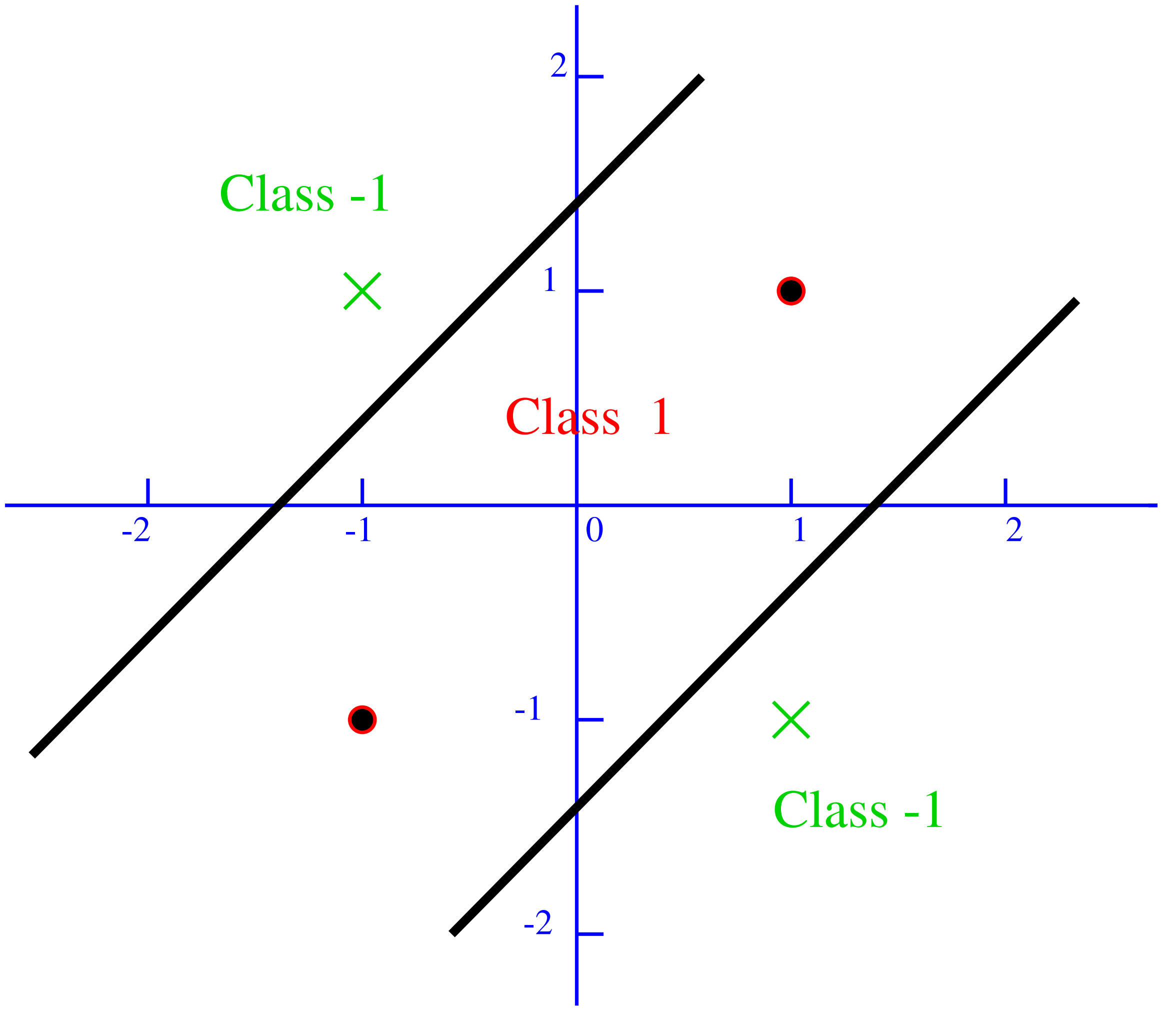
Same decision function: $(x_1 x_2)_*$ obtained by:

- $f(u) = \frac{1}{2} u' D K(A, A') D u$ **(Conventional SVM) (QP)**
- $f(u) = \frac{1}{2} \|u\|_2^2$ **(QP)**
- $f(u) = \|u\|_1$ **(LP)**

Alternate Decision Function for Same Problem, Same Kernel

$$(2 - (x_1 - x_2)^2)_*$$

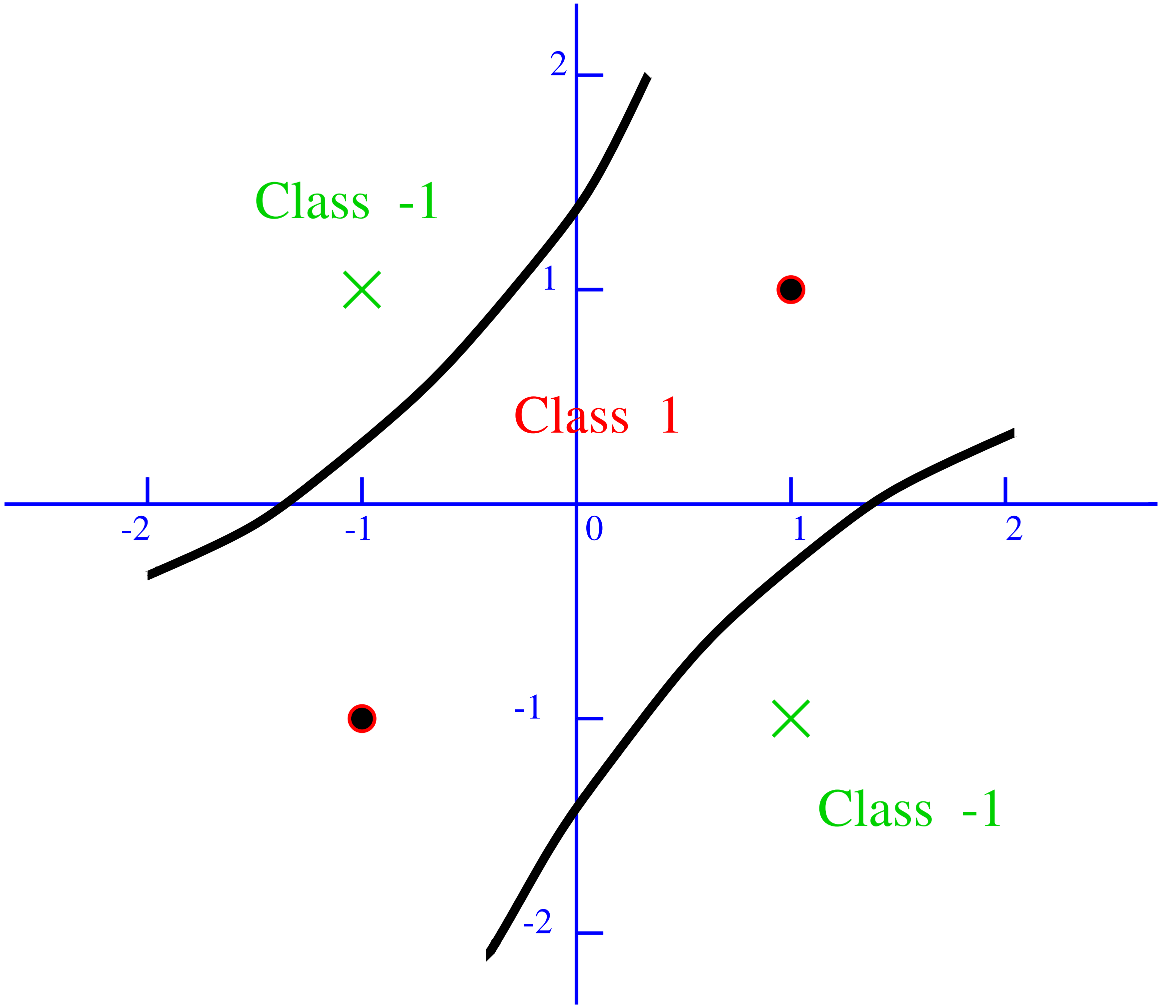
Obtained by $f(u) = \|u\|_1$ (LP Alternate Solution)



Decision Function for Same Problem, Same Kernel

$$(2 + x_1 x_2 - (x_1 - x_2)^2)_*$$

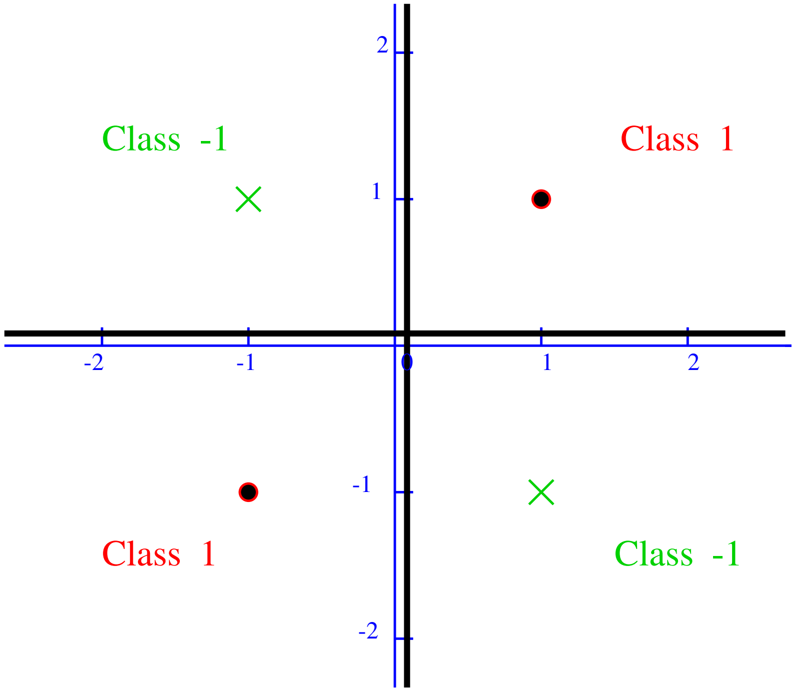
Obtained by $f(u) = \|K(A, A')Du\|_1$ (LP)



Decision Function for Same Problem, Different Kernel

ND Cubic Kernel: $K(A, A') = (-AA' - ee')^3_\bullet$

Conventional SVM: $f(u) = \frac{1}{2}u'DK(A, A')Du$ Failed



Same decision function: $(x_1x_2)_*$ for:

- $f(u) = \frac{1}{2}\|u\|_2^2$ **(QP)**
- $f(u) = \|u\|_1$ **(LP)**
- $f(u) = \|K(A, A')Du\|_1$ **(LP)**

Ability to extract correct separating function from ND kernel!

LPC: Linear Program Chunking

Motivation: Discrimination over Massive Datasets

- **Consider the general linear program:**

$$\min_{x} \{c'x \,|\, Hx \geq b\},$$

- **where $c \in R^n$, $H \in R^{m \times n}$ and $b \in R^m$.**
- **Note:** ***m may be orders of magnitude larger than n.***
- **Linear Program Chunking Algorithm (LPC) solves this LP with massive constraints.**
  - **Monotonicity of sub-problem objective values.**
  - **Finite termination.**
  - **In dual form, equivalent to block-column generation method related to [Gilmore-Gomory 1961, Dantzig-Wolfe 1960].**

LPC Algorithm for $\min_x \{c'x \,|\, Hx \geq b\}$

Algorithm: Partition $[H \quad b]$ into ℓ blocks, possibly of different sizes, as follows:

$$\begin{bmatrix} H & b \end{bmatrix} = \begin{bmatrix} H^1 & b^1 \\ \vdots & \vdots \\ H^\ell & b^\ell \end{bmatrix}.$$

Assume that original LP and all subproblems below, have vertex solutions and that subproblem solutions are nondegenerate. At iteration $j = 1, \ldots$ compute x^j by solving the following linear program:

$$x^j \in \arg \text{vertex} \min \left\{ c'x \;\middle|\; \begin{array}{lcl} H^{(j \bmod \ell)} x & \geq & b^{(j \bmod \ell)} \\ \bar{H}^{(j \bmod \ell)-1} x & \geq & \bar{b}^{(j \bmod \ell)-1} \end{array} \right\},$$

- $[\bar{H}^0 \quad \bar{b}^0]$ is empty.
- $[\bar{H}^j \quad \bar{b}^j]$ = set of active constraints of subproblem j with positive optimal Lagrange multipliers. (Support Vectors)
- Stop when $c'x^j = c'x^{j+\nu}$ for some input integer ν. Typically $\nu = 4$.

Finite Termination of LPC Algorithm

Theorem: The sequence $\{x^j\}$ generated by the LPC Algorithm has the following properties:

(i) The sequence $\{c'x^j\}$ of objective function values is nondecreasing and is bounded above by the global minimum of $\min_x \{c'x \,|\, Hx \geq b\}$.

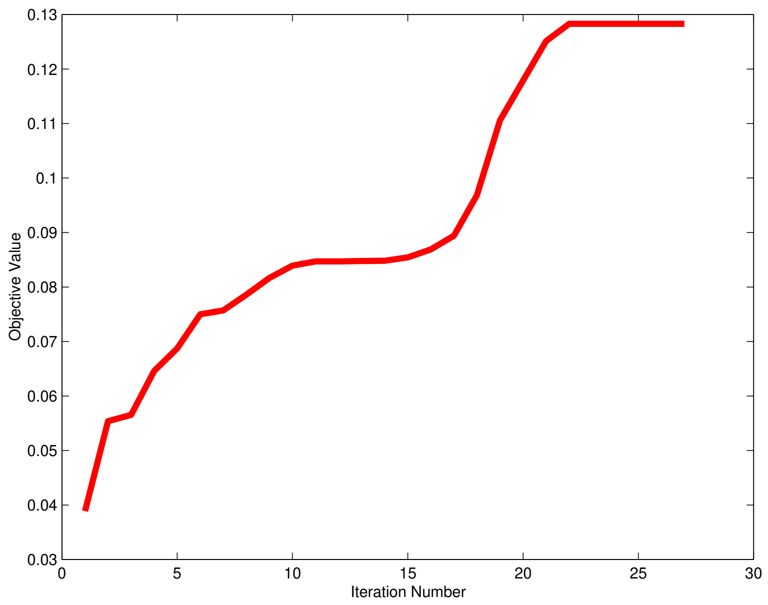
(ii) The sequence of objective function values $\{c'x^j\}$ becomes constant, that is: $c'x^{j+1} = c'x^j$ for all $j \geq \bar{j}$ for some $\bar{j} \geq 1$.

(iii) For $j \geq \bar{j}$, active constraints of the j-th subproblem at x^j with positive multipliers remain active for iteration $j+1$.

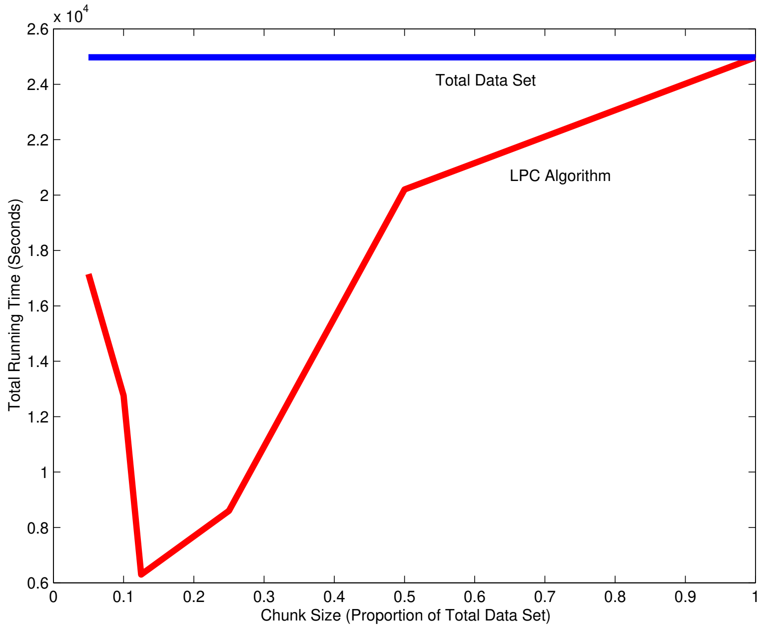
(iv) For all $j \geq \bar{j}$, x^j is a solution of the original linear program $\min_x \{c'x \mid Hx \geq b\}$.

- Useful for SVM application and general LPs.

Discriminating between 200,000 points in R^{32}



(a) Objective function values versus iteration number for chunk size of 5%.



(b) Total running time (seconds) versus chunk size.

- LPC reduced running time from 6.94 hours by ordinary LP to 1.75 hours with chunk size 12.5%.
- For all chunk sizes $\leq 50\%$, LPC produced solution in less time than MINOS.

Discrimination of 500,000 and 1,000,000 points in R^{32}

- **500,000 Points:**
  - **Chunk size = 12.5%.**
  - **Solution after 25.9 hours and 8 iterations.**
  - **Subproblem objective constant over $\nu = 4$ iterations with no change in support vectors.**
- **1,000,000 Points:**
  - **Chunk size = 2%.**
  - **Solution after 231.3 hours and 63 iterations.**
  - **Subproblem objective constant over $\nu = 4$ iterations with no change in support vectors.**

Feature Selection

- **Given:** Two finite point sets $\mathcal{A}$ and $\mathcal{B}$ in n–dimensional real space.
- **Problem:** Utilizing *as few of the original features (dimensions) as possible* discriminate between the points of $\mathcal{A}$ and $\mathcal{B}$ by a separating plane, $x'w = \gamma$.
  - Remove irrelevant or redundant features.
- **Typical Computational Results**
  - Breast Cancer Prognosis features $32 \to 4$
  - Tenfold cross–validation $59.9\% \to 74.1\%$

Mathematical Programming Formulation

- **Given:** Two finite point sets $\mathcal{A}$ and $\mathcal{B}$ in n–dimensional real space.
- **Target:** Two objectives *simultaneously*: separability & sparsity
  - Compute a scalar γ, and $w \in R^n$ with a maximum number of zeros (sparsity), such that:

$$\left\{ \begin{array}{l} x'w > \gamma \text{ for "most" } x \in \mathcal{A} \\ x'w < \gamma \text{ for "most" } x \in \mathcal{B} \end{array} \right\} \text{ (separation)}$$

- **Twofold Objective:**

$$(1-\lambda) \text{ separation } + \lambda \text{ sparsity }, \quad \lambda \in [0,1)$$

Separation of $\mathcal{A}$ and $\mathcal{B}$ via Linear Programming

$A:\ m \times n$ matrix, $B:\ k \times n$ matrix

Find a plane $x'w = \gamma$, with normal w and distance $\frac{|\gamma|}{\|w\|_2}$ from origin such that:

$$Aw > e\gamma,\ Bw < e\gamma,$$

where e is a vector of ones of appropriate dimension

$$\Updownarrow$$

$$Aw \geq e\gamma + e,\ Bw \leq e\gamma - e$$

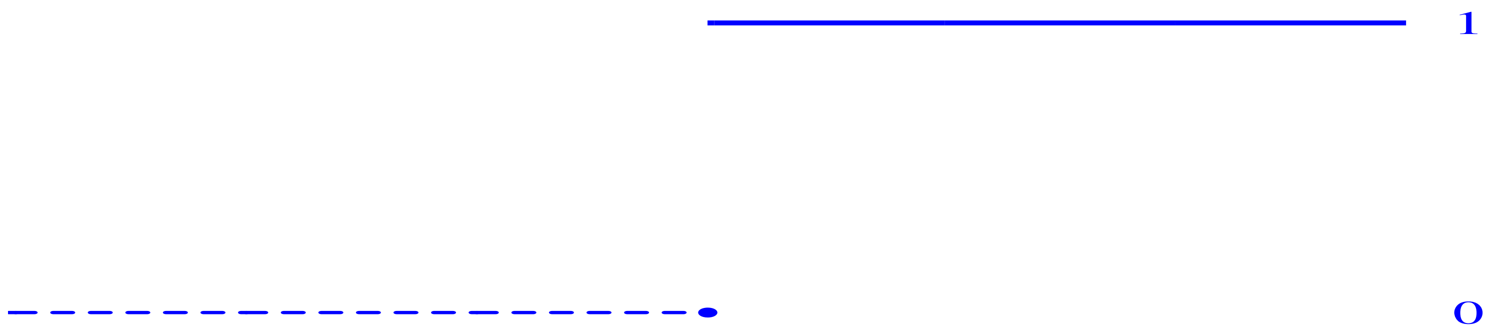
$$\Downarrow$$

$$\min_{w,\gamma,y,z} \left\{ \frac{e'y}{m} + \frac{e'z}{k} \,\middle|\, Aw + y \geq e\gamma + e,\ Bw - z \leq e\gamma - e,\ y \geq 0,\ z \geq 0 \right\}$$

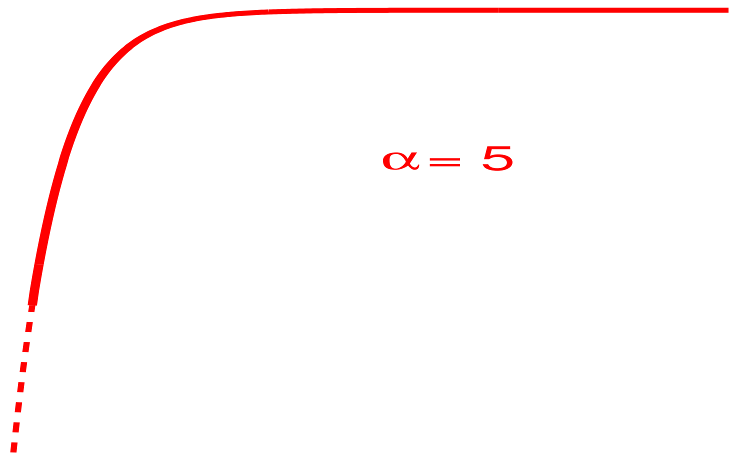
Step Function on Nonnegative Real Line

$$v \geq 0$$

$$v_* = \begin{cases} 1 & \text{if } v > 0, \\ 0 & \text{if } v = 0 \end{cases}$$



Smooth Approximation $1 - \epsilon^{-\alpha v}$ to Step Function on Nonnegative Real Line



$(1-\lambda)$ Separation $+\ \lambda$ (Feature Suppression)

$\lambda \in [0,1)$

$$\min_{w,\gamma,y,z} \left\{ (1-\lambda)\left(\frac{e'y}{m}+\frac{e'z}{k}\right)+\lambda e' \mid w \mid_* \;\middle|\; \begin{array}{c} -Aw+e\gamma+e \le y, \\ Bw-e\gamma+e \le z, \\ y \ge 0,\ z \ge 0 \end{array} \right\}$$

$$\Updownarrow$$

$$\text{(FS)} \min_{\boldsymbol{w},\boldsymbol{\gamma},\boldsymbol{y},\boldsymbol{z},\boldsymbol{v}} \left\{ (\mathbf{1}-\boldsymbol{\lambda})\left(\frac{\boldsymbol{e'y}}{\boldsymbol{m}} + \frac{\boldsymbol{e'z}}{\boldsymbol{k}}\right) + \boldsymbol{\lambda e' v_*} \;\middle|\; \begin{array}{c} -\boldsymbol{Aw} + \boldsymbol{e\gamma} + \boldsymbol{e} \le \boldsymbol{y}, \\ \boldsymbol{Bw} - \boldsymbol{e\gamma} + \boldsymbol{e} \le \boldsymbol{z}, \\ \boldsymbol{y} \ge \mathbf{0}, \boldsymbol{z} \ge \mathbf{0}, \\ -\boldsymbol{v} \le \boldsymbol{w} \le \boldsymbol{v} \end{array} \right\}$$

$$\Updownarrow \; v_* \cong e - \varepsilon^{-\alpha v}, \alpha > 0$$

$$\text{(FSV)} \min_{w,\gamma,y,z,v} \left\{ (1-\lambda)\left(\frac{e'y}{m} + \frac{e'z}{k}\right) + \lambda(n - e'\varepsilon^{-\alpha v}) \;\middle|\; \begin{array}{c} -Aw + e\gamma + e \leq y, \\ Bw - e\gamma + e \leq z, \\ y \geq 0, z \geq 0, \\ -v \leq w \leq v \end{array} \right\}$$

Successive Linearization Algorithm: SLA
(Stepless Frank-Wolfe)

$$(w^{i+1}, \gamma^{i+1}, y^{i+1}, z^{i+1}, v^{i+1}) \in$$

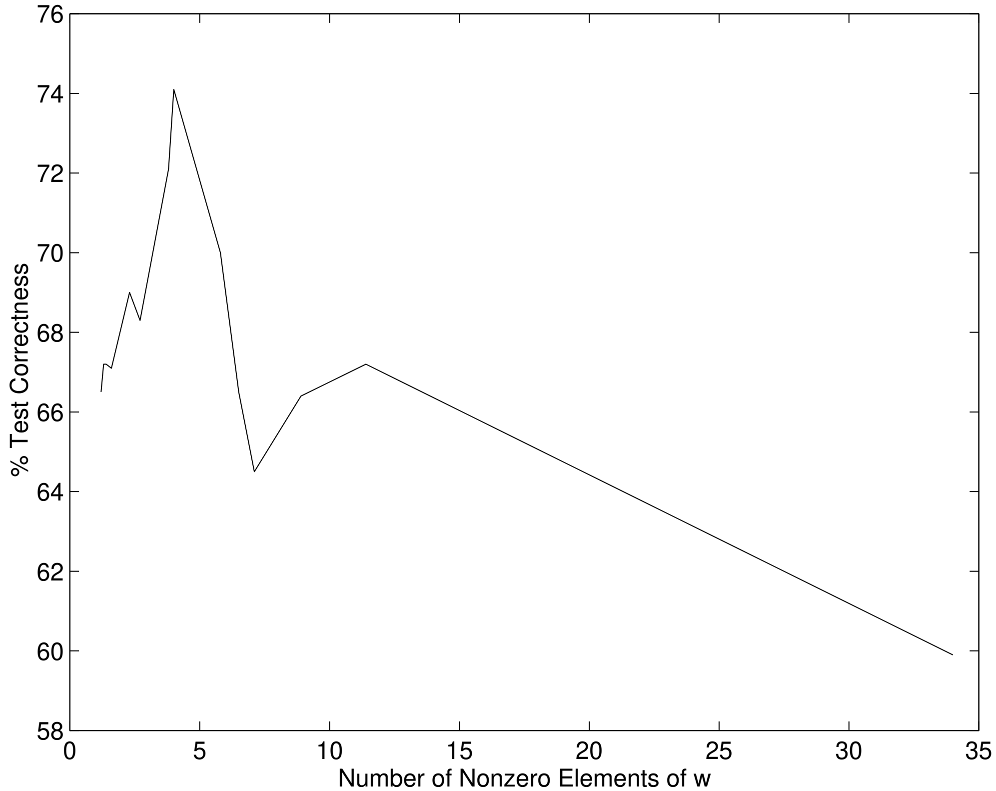
$$\arg\text{vertex} \min_{w,\gamma,y,z,v} \left\{ \begin{array}{c} (1-\lambda)\left(\frac{e'(y-y^i)}{m} + \frac{e'(z-z^i)}{k}\right) \\ \\ +\lambda\alpha\varepsilon^{-\alpha v^i}(v-v^i) \end{array} \middle| \begin{array}{c} -Aw + e\gamma + e \leq y, \\ Bw - e\gamma + e \leq z, \\ y \geq 0,\ z \geq 0, \\ -v \leq w \leq v \end{array} \right\}$$

Stop when minimum $= 0$

Finite Termination

SLA generates strictly decreasing function values and terminates in a finite number of steps (typically 3 to 7 linear programs) at a global minimum or stationary point.

Feature Selection Example
Wisconsin Breast Cancer Prognosis Database
% Test Set Correctness Versus No. of Features Used



- **147 patients: 28 patients cancer recurred within 2 years, 119 patients did not**
- **23.7% increase in test set correctness (over plain linear separation).**
- **Decrease features used from 32 (by plain linear separation) to 4.**

Clustering in Data Mining

General Objective

- **Given:** A database of m points in n-dimensional realspace R^n.
- **Problem:** Extract hidden distinct properties by clustering the database.

Specific Objective

- **Given:** m points in n-dimensional real space R^n, fixed number k of clusters.
- **Problem:** Determine k centers in R^n so that the sum of "distances" of each point to the nearest center is minimized.
  - Polyhedral (1-norm) distance used
    - De-emphasizes outliers
    - Leads to simple, finite algorithm
- **Fast finite k-Median Algorithm**
  - Alternate Between Two Linear Programs
  - Linear Programs Solved in Closed Form
  - Better than k-Mean Algorithm at Extracting Distinct Survival Curves.

Concave Minimization Formulation

- **Given:** Set $\mathcal{A}$ of m points in R^n represented by the matrix $A \in R^{m\times n}$, and a number k of desired clusters.
- **Problem:** Determine centers C_ℓ, $\ell = 1, \ldots, k$ in R^n such that the sum of the minima over $\ell \in \{1, \ldots, k\}$ of the 1-norm distance between each point A_i, $i = 1, \ldots, m$, and cluster centers C_ℓ, $\ell = 1, \ldots, k$ is minimized.
- **Objective:** Sum of minima of k linear functions, hence it is piecewise-linear concave.
- **Difficulty:** Minimizing a general piecewise-linear concave function over a polyhedral set is NP-hard.

Clustering via Concave Minimization

- **Reformulation**

$$\min_{C,D} \sum_{i=1}^{m} \min_{\ell=1,\ldots,k} \{e'D_{i\ell}\}$$

$$\text{s.t.} \quad -D_{i\ell} \le A_i' - C_\ell \le D_{i\ell},\ i = 1,\ldots,m,\ \ell = 1,\ldots,k$$

$$\Downarrow$$

$$\min_{C_\ell \in R^n, D_{i\ell} \in R^n, T_{i\ell} \in R} \sum_{i=1}^{m} \sum_{\ell=1}^{k} T_{i\ell}\{e'D_{i\ell}\}$$

$$-D_{i\ell} \le A_i' - C_\ell \le D_{i\ell}, i = 1\ldots,m,\ \ell = 1,\ldots,k$$

$$\textstyle\sum_{\ell=1}^{k} T_{i\ell} = 1 \qquad T_{i\ell} \ge 0,\ i = 1,\ldots,m,\ \ell = 1,\ldots,k$$

Finite K-Median Clustering Algorithm

(Minimizing Piecewise-Linear Concave Function)

0) $\boxed{\textbf{Initialization}}$ **Choose k arbitrary initial cluster centers at, say approximately, densest, second densest,$\cdots$, k-densest locations.**

1) $\boxed{\textbf{Cluster Assignment}}$ **Assign points to cluster with nearest cluster center in the 1-norm.**

2) $\boxed{\textbf{Center Update}}$ **Recompute location of center for each cluster as the cluster median (closest point to all cluster points in 1-norm).**

Stop if cluster centers are unchanged, else go to 1).

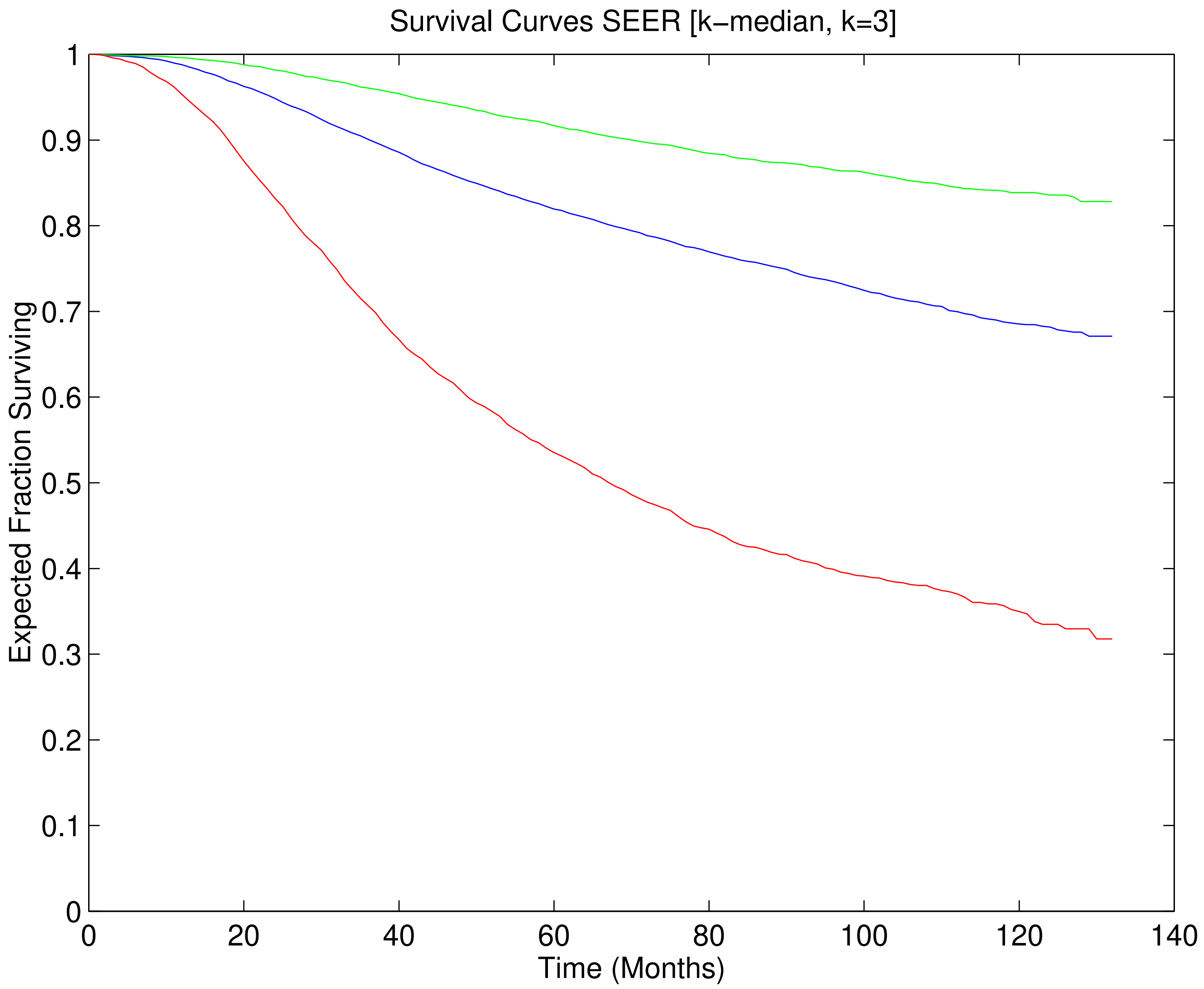
k-Mean Algorithm
Theoretical and Computational Considerations

- **Underlying Problem for p-norm**

$$\min_{C,D} \sum_{i=1}^{m} \min_{\ell=1,\ldots,k} \|D_{i\ell}\|_p \text{ s.t.} -D_{i\ell} \leq A_i' - C_\ell \leq D_{i\ell},\ i = 1, \ldots, m,\ \ell = 1, \ldots, k$$

- **$p \neq 1$**
  - *Not* concave minimization over a polyhedral set.
- **k-Mean Algorithm $p = 2$**
  - Finds stationary point *not* of above problem.
  - Finds stationary point of above problem with $\|D_{i\ell}\|_2$ replaced by $\|D_{i\ell}\|_2^2$.
  - Without squared term, subproblem becomes Weber Problem – locate center in R^n closest (2-norm) to a finite number of points – has no closed form solution!
  - k-Mean Algorithm is more sensitive to outliers

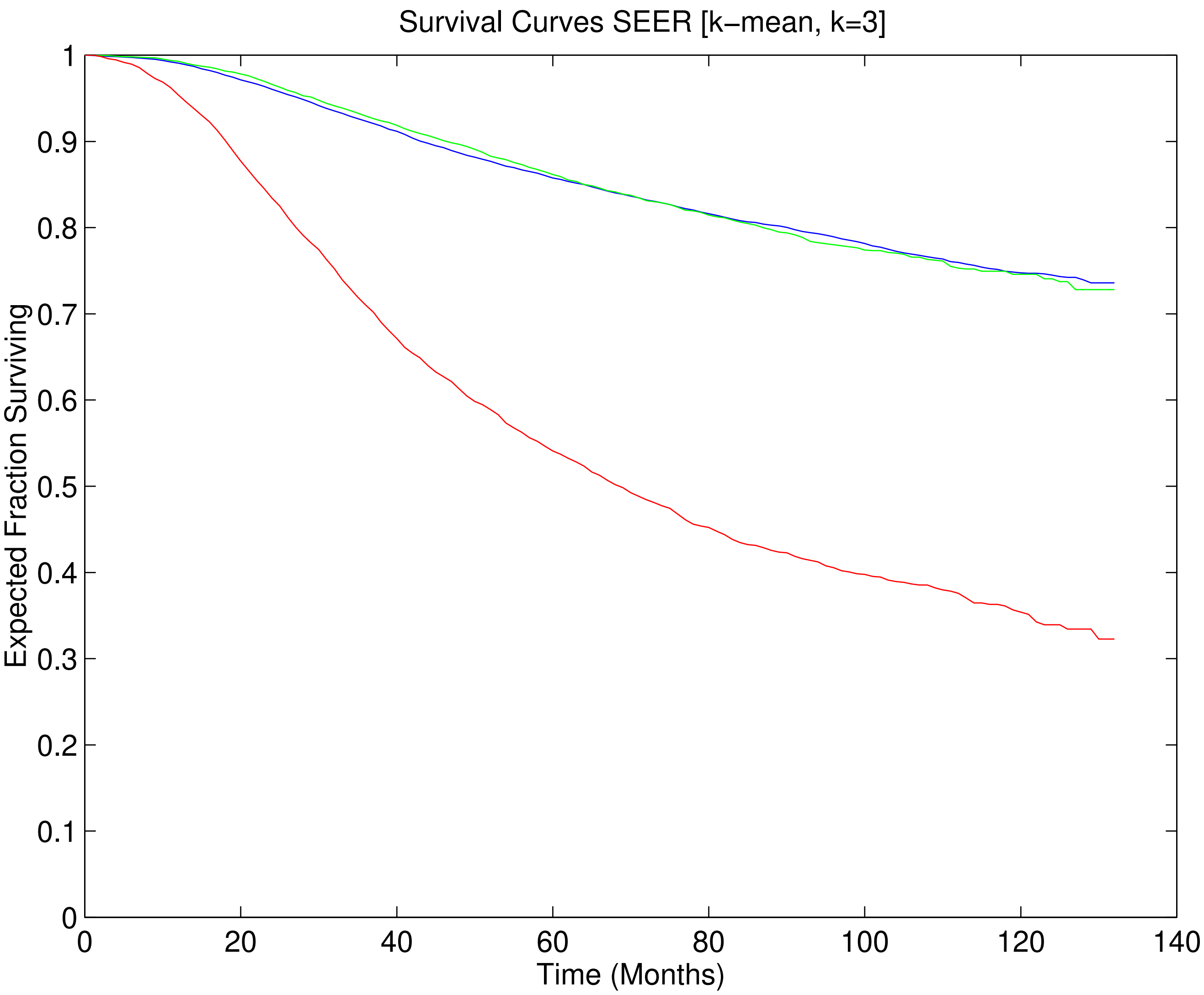
Disease-Free Survival Curves for the Three Clusters
National Cancer Institute SEER Database: 21,960 Cases
Data mined for distinct survival curves
k-Median Algorithm



Disease-Free Survival Curves for the Three Clusters

National Cancer Institute SEER Database: 21,960 Cases

k-Mean Algorithm



Conclusion

- **Mathematical programming**
  - **Versatile tool for modeling & solving a variety of problems including:**
- **Linear & nonlinear classification**
  - **Large margin**
  - **Minimum support**
- **Massive data discrimination**
- **Feature extraction**
- **Data mining**
- **Collaboration between mathematical programming, machine learning and data mining is very effective and should be vigorously pursued.**

References

Except for references 2 & 3, all are at: www.cs.wisc.edu/~olvi

1. OLM: "Arbitrary-norm separating plane", Mathematical Programming Technical Report 97-07, May 1997, Operations Research Letters, to appear.

2. OLM: "Linear and nonlinear separation of patterns by linear programming", Operations Research 13, 1965, 444-452.

3. OLM: "Multi-surface method of pattern separation", IEEE Trans. on Information Theory, IT-14, 1968, 801-807.

4. OLM: "Generalized Support Vector Machines", Mathematical Programming Technical Report 98-14, October 1998.

5. P. S. Bradley & OLM:"Massive data discrimination via linear support vector machines", Mathematical Programming Technical Report 98-05, May 1998.

6. P. S. Bradley, OLM & W. Nick Street: "Feature selection via mathematical programming", INFORMS Journal on Computing 10, 1998, 209-217.

7. P. S. Bradley, OLM & W. Nick Street: "Clustering via concave minimization", in "Advances in Neural Information Processing Systems -9-", (NIPS*96), M. C. Mozer and M. I. Jordan and T. Petsche, editors, MIT Press, Cambridge, MA, 1997, 368-374.

8. OLM & D. R. Musicant:"Successive Overrelaxation for SupportVector Machines", Mathematical Programming Technical Report 98-18, November 1998.