Efficient active set methods for support vector machines.

Katya Scheinberg

University of Texas, Austin, April 5, 2007

Outline

Convex QP for SVM

The active set method

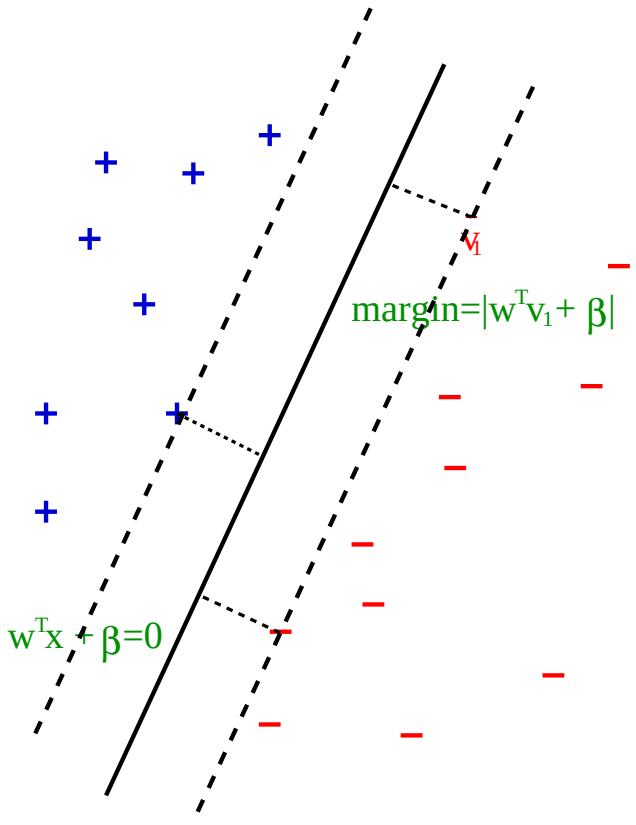
Exploiting structure

Computational results

Parametric (regularization) path

Computing the path

Linear Classification



maximize margin, s.t. $||w|| = 1$

minimize $||w||^2$, s.t. margin $= 1$

Soft margin linear separation problem

Two sets of points: $V_+ \subset \mathbf{R^k}$ and $V_- \subset \mathbf{R^k}$. Total number of points: n

$$\begin{aligned} \min_{w,\beta} \quad & \frac{1}{2} w^T w \\ \text{s.t.} \quad & w^T v_i + \beta \leq -1, \quad v_i \in V_- \\ & w^T v_i + \beta \geq 1, \quad v_i \in V_+ \end{aligned}$$

Soft margin linear separation problem

Two sets of points: $V_+ \subset \mathbf{R^k}$ and $V_- \subset \mathbf{R^k}$. Total number of points: n

$$\begin{aligned} \min_{w,\beta} \quad & \frac{1}{2} w^T w \\ \text{s.t.} \quad & w^T v_i + \beta \leq -1 + \xi_i, \quad v_i \in V_- \\ & w^T v_i + \beta \geq 1 - \xi_i, \quad v_i \in V_+ \end{aligned}$$

Soft margin linear separation problem

Two sets of points: $V_+ \subset \mathbf{R^k}$ and $V_- \subset \mathbf{R^k}$. Total number of points: n

$$\min_{w,\beta} \quad \frac{1}{2} w^T w + c \sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad w^T v_i + \beta \leq -1 + \xi_i, \quad v_i \in V_-$$

$$w^T v_i + \beta \geq 1 - \xi_i, \quad v_i \in V_+$$

$$\xi \geq 0, \qquad i = 1, \ldots, n.$$

Soft margin linear separation problem

Two sets of points: $V_+ \subset \mathbf{R^k}$ and $V_- \subset \mathbf{R^k}$
Total number of points: n

$$\begin{aligned} \min_{\xi,w,\beta} \quad & \frac{1}{2} w^T w + c \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(w^T v_i + \beta) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi \geq 0, \qquad i = 1, \ldots, n. \end{aligned}$$

$$y_i = 1 \text{ if } v_i \in V_+$$

$$y_i = -1 \text{ if } v_i \in V_-$$

Dual problem

At optimality $w^* = \sum_{i=1}^{n} \alpha_i y_i v_i, \quad 0 \leq \alpha_i \leq c$

$$\begin{aligned} \min_{\alpha,\beta,\xi} \quad & \frac{1}{2}\alpha^T Q\alpha + c\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & -Q\alpha + y\beta - \xi_i \leq -1, \quad i = 1,\ldots,n \\ & \xi \geq 0, 0 \leq \alpha_i \leq c \qquad i = 1,\ldots,n, \end{aligned}$$

$$Q := D_y V V^T D_y \quad \Leftrightarrow \quad Q_{ij} = y_i y_j v_i^T v_j$$

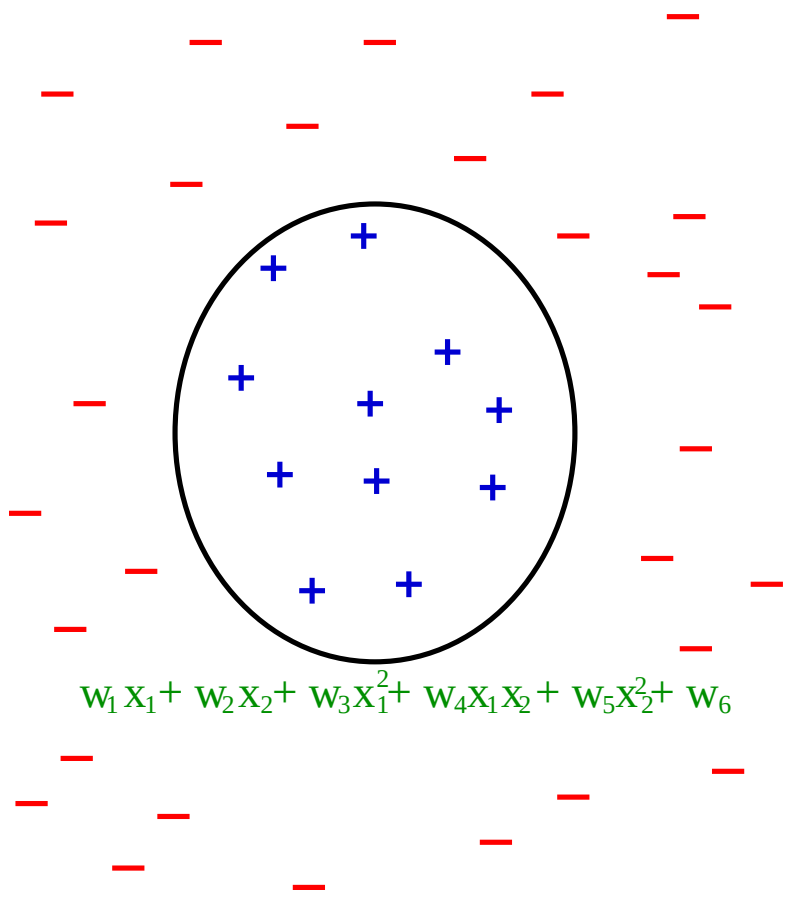
Dual problem

At optimality $w^* = \sum_{i=1}^{n} \alpha_i y_i v_i, \quad 0 \le \alpha_i \le c$

$$\begin{aligned} \min_{\alpha,\beta,\xi} \quad & \frac{1}{2}\alpha^T Q\alpha + c\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & -Q\alpha + y\beta + s_i - \xi_i = -1, \quad i = 1,\ldots,n \\ & s_i \ge 0, \xi \ge 0, 0 \le \alpha_i \le c \qquad i = 1,\ldots,n, \end{aligned}$$

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ \text{s.t.} \quad & y^T\alpha = 0, \\ & 0 \le \alpha \le c, \end{aligned}$$

Nonlinear Classification



Kernel Operation: nonlinear separation

Separating by a linear surface in a higher dimensional space.

Mapping: $v_i \rightarrow \phi(v_i)$

$Q_{ij} = y_i y_j v_i^T v_j \rightarrow Q_{ij} = y_i y_j \phi(v_i)^T \phi(v_j) = y_i y_j K(v_i, v_j)$

Compute inner products only without the actual mapping

Kernel operation: $K(v_i, v_j) = \phi(v_i)^T \phi(v_j)$

Examples:

1. $K(v_i, v_j) = \exp^{-||v_i - v_j||^2/2\sigma^2}$
2. $K(v_i, v_j) = (v_i^T v_j / a_1 + a_2)^d$

Optimality conditions

Convex quadratic optimization problem

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^{T} Q\alpha - e^{T}\alpha \\ \text{s.t.} \quad & y^{T}\alpha = 0, \\ & 0 \leq \alpha \leq c, \end{aligned}$$

KKT conditions

$$\begin{aligned} & \alpha_i s_i = 0, \quad i = 1, \ldots, n, \\ & (c - \alpha_i)\xi_i = 0 \quad i = 1, \ldots, n, \\ & y^{T}\alpha = 0, \\ & -Q\alpha + y\beta + s - \xi = -e, \\ & 0 \leq \alpha \leq c, \ s \geq 0, \ \xi \geq 0. \end{aligned}$$

Active set

Given a dual basic feasible solution, (α, β, s, ξ), we partition $I = \{1, \ldots, n\}$ into I_0, I_c and I_s:

- $\forall i \in I_0\ \xi_i = 0$ and $\alpha_i = 0$, ($s_i \geq 0$?)
- $\forall i \in I_c\ s_i = 0$ and $\alpha_i = c$, ($\xi_i \geq 0$?)
- $\forall i \in I_s\ s_i = \xi_i = 0$ and $0 < \alpha_i < c$.

$I_0 \cup I_c \cup I_s = I$ and $I_0 \cap I_c = I_c \cap I_s = I_0 \cap I_s = \emptyset$.

Based on the partition (I_0, I_c, I_s) we define Q_{ss} (Q_{cs}, Q_{sc}, Q_{cc}, Q_{0s}, Q_{00}), y_s (y_c, y_0) and α_s (α_c, α_0)

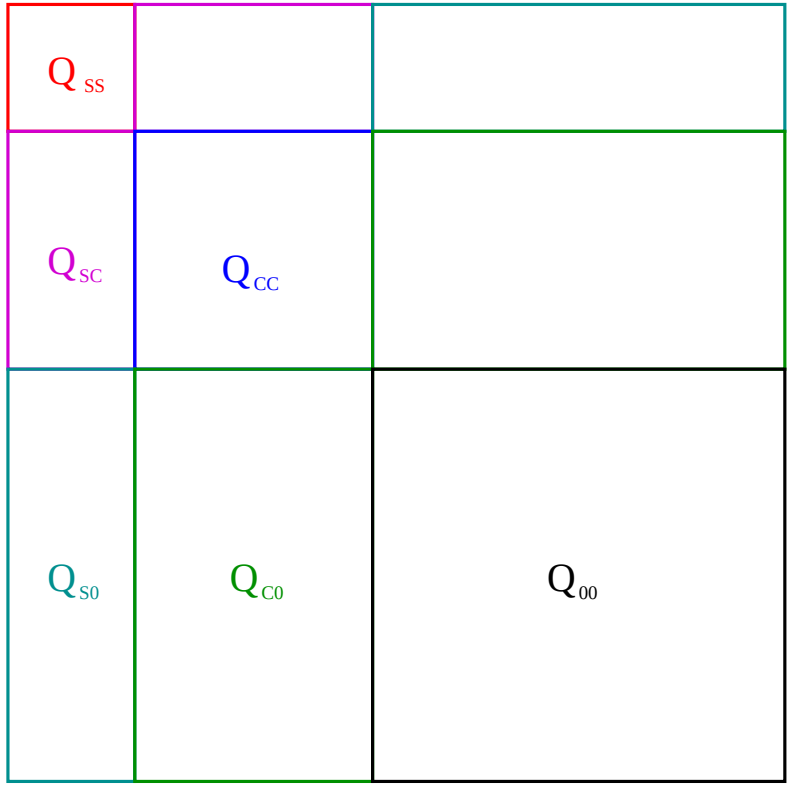
Active set method

$$
\begin{aligned}
&\alpha_i s_i = 0, \quad i = 1, \ldots, n, \\
&(c - \alpha_i)\xi_i = 0 \quad i = 1, \ldots, n, \\
&y^T \alpha = 0, \\
&-Q\alpha + y\beta + s - \xi = -e, \\
&0 \le \alpha \le ce, \; s \ge 0, \; \xi \ge 0.
\end{aligned}
$$

Reduced system

$$
\begin{aligned}
&y_s^T \alpha_s = -y_c^T \alpha_c, \\
&-Q_{ss}\alpha_s + y_s\beta = -e_s + cQ_{sc}e_c, \\
&0 \le \alpha_s \le ce, \\
&s_0 = Q_{0s}\alpha_s - y_0\beta - e_0 - cQ_{0c}e_c, \\
&\xi_c = -Q_{cs}\alpha_s + y_c\beta + e_c + cQ_{cc}e_c.
\end{aligned}
$$

Partition of Q



Algorithm

Step 1

(i) Solve

$$\begin{aligned} \min_{\alpha_s} \quad & \frac{1}{2}\alpha_s^T Q_{ss}\alpha_s + ce^T Q_{cs}\alpha_s - e^T\alpha_s \\ \text{s.t.} \quad & y_s^T\alpha_s = -y_c^T\alpha_c \end{aligned}$$

(ii) From the current iterate make a step toward the solution until for some $i \in I_s$ $(\alpha_s)_i = 0$ or $(\alpha_s)_j = c$ or until solution is reached.

(iii) If for some $i \in I_s$, $(\alpha_s)_i = 0$
Then update $I_s = I_s \backslash \{i\}$, $I_0 = I_0 \cup \{i\}$, and go to step (i).

(iv) If for some $i \in I_s$, $(\alpha_s)_i = c$
then update $I_s = I_s \backslash \{i\}$, $I_c = I_c \cup \{i\}$, and go to step (i).

(v) If the optimum is reached in step (ii), proceed to **Step 2**.

Algorithm

Step 2

(i) Compute s_0

$$s_0 = -Q_{0s}\alpha_s - y_0\beta + 1 - cQ_{0c}e$$

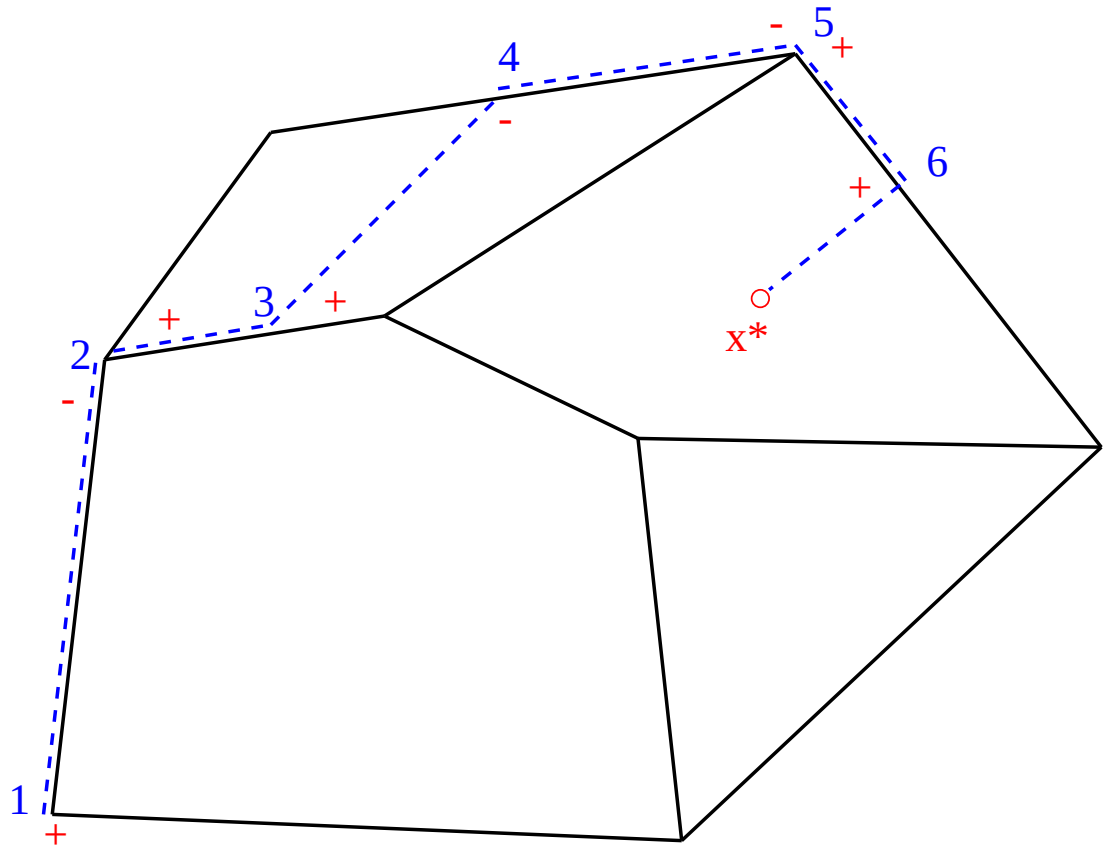
and ξ_c

$$\xi_c = Q_{cs}\alpha_s + y_c\beta - 1 + cQ_{cc}e$$

(ii) Find $i_0 = \text{argmin}_i\{s_i : i \in I_0\}$.
Find $i_c = \text{argmin}_i\{\xi_i : i \in I_c\}$.

(iii) If $s_{i_0} \geq 0$ and $\xi_{i_c} \geq 0$ then the current solution is optimal, **Exit.**
If $s_{i_0} \leq \xi_{i_c}$, then $I_s = I_s \cup \{i_0\}$ and $I_0 = I_0 \backslash \{i_0\}$.
Else, $I_s = I_s \cup \{i_c\}$ and $I_c = I_c \backslash \{i_c\}$.
Go to **Step 1.**

Active set method



Workload

Step 1

(i) Solve a system with matrix

$$\begin{bmatrix} Q_{ss} & y \\ y^T & 0 \end{bmatrix}.$$

If factorization $Q_{ss} = G_s G_s^T$ is available, then work is $\mathbf{O(n_s^2)}$.

Workload

Step 1

(i) Solve a system with matrix

$$\begin{bmatrix} Q_{ss} & y \\ y^T & 0 \end{bmatrix}.$$

If factorization $Q_{ss} = G_s G_s^T$ is available, then work is $\mathbf{O(n_s^2)}$.

(ii) Step toward solution. $\mathbf{O(n_s)}$

Workload

Step 1

(i) Solve a system with matrix

$$\begin{bmatrix} Q_{ss} & y \\ y^T & 0 \end{bmatrix}.$$

If factorization $Q_{ss} = G_s G_s^T$ is available, then work is $\mathbf{O(n_s^2)}$.

(ii) Step toward solution. $\mathbf{O(n_s)}$

(iii) If for some $i \in I_s$, $(\alpha_s)_i = 0$, then update $I_s = I_s \backslash \{i\}$, $I_0 = I_0 \cup \{i\}$, update G_s by removing a row. $\mathbf{O(n_s^2)}$

Workload

Step 1

(i) Solve a system with matrix

$$\begin{bmatrix} Q_{ss} & y \\ y^T & 0 \end{bmatrix}.$$

If factorization $Q_{ss} = G_s G_s^T$ is available, then work is $\mathbf{O(n_s^2)}$.

(ii) Step toward solution. $\mathbf{O(n_s)}$

(iii) If for some $i \in I_s$, $(\alpha_s)_i = 0$, then update $I_s = I_s \backslash \{i\}$, $I_0 = I_0 \cup \{i\}$, update G_s by removing a row. $\mathbf{O(n_s^2)}$

(iv) If for some $i \in I_s$, $(\alpha_s)_i = c$ then update $I_s = I_s \backslash \{i\}$, $I_c = I_c \cup \{i\}$, update $e^T Q_{cs}$ and G_s by removing a row. $\mathbf{O(n_s^2) + O(n_c)}$

Workload

Step 2

(i)

$$s_0 = -Q_{0s}\alpha_s - y_0\beta + 1 - cQ_{0c}e$$

$$\xi_c = Q_{cs}\alpha_s + y_c\beta - 1 + cQ_{cc}e$$

$$\mathbf{O(n_s n)}$$

Workload

Step 2

(i)

$$s_0 = -Q_{0s}\alpha_s - y_0\beta + 1 - cQ_{0c}e$$

$$\xi_c = Q_{cs}\alpha_s + y_c\beta - 1 + cQ_{cc}e$$

$\mathbf{O(n_s n)}$

(ii) Find $i_0 = \text{argmin}_i\{s_i : i \in I_0\}$, $i_c = \text{argmin}_i\{\xi_i : i \in I_c\}$. $\mathbf{O(n)}$

Workload

Step 2

(i)

$$s_0 = -Q_{0s}\alpha_s - y_0\beta + 1 - cQ_{0c}e$$

$$\xi_c = Q_{cs}\alpha_s + y_c\beta - 1 + cQ_{cc}e$$

$\mathbf{O(n_s n)}$

(ii) Find $i_0 = \text{argmin}_i\{s_i : i \in I_0\}$, $i_c = \text{argmin}_i\{\xi_i : i \in I_c\}$. $\mathbf{O(n)}$

(iii) If $s_{i_0} \leq \xi_{i_c}$, then $I_s = I_s \cup \{i_0\}$ and $I_0 = I_0 \backslash \{i_0\}$.
Update G_s by adding a row
Else, $I_s = I_s \cup \{i_c\}$ and $I_c = I_c \backslash \{i_c\}$.
Update $e^T Q_{cs}$ and G_s by adding a row

$\mathbf{O(n_s^2) + O(n_c)}$

Results

Problem	dim	n	C	σ	$\lvert I_s \rvert$	$\lvert I_c \rvert$	SVM^{light}	SVM-QP
Letter-G	16	20000	100	0	17	1056	1052	128
Letter-G	16	20000	100	100	241	39	19	3
Letter-G	16	20000	100	40	346	8	11	4
Web	300	49749	100	40	1834	678	1881	336
Web	300	49749	100	100	1404	905	1900	272
Adult	123	32561	100	100	1317	9953	7385	848
Adult	123	32561	100	200	685	10594	4864	630
USPS	676	266079	100	100	2906	0	1713	1650
USPS	676	266079	100	1000	1371	1	1349	966

Parametric SVM

$$
\begin{aligned}
\min_{\alpha,\beta,\xi} \quad & \frac{1}{2}\alpha^T Q\alpha + c\sum_{i=1}^{n}\xi_i \\
\text{s.t.} \quad & -Q\alpha + y\beta - \xi_i \le -1, \quad i = 1,\ldots,n \\
& \xi \ge 0, 0 \le \alpha_i \le c \qquad i = 1,\ldots,n,
\end{aligned}
$$

Parametric SVM

$$
\begin{aligned}
\min_{\alpha,\beta,\xi} \quad & \frac{1}{2}\alpha^T Q\alpha + c\sum_{i=1}^{n}\xi_i \\
\text{s.t.} \quad & -Q\alpha + y\beta - \xi_i \leq -1, \quad i = 1,\ldots,n \\
& \xi \geq 0, 0 \leq \alpha_i \leq c \qquad i = 1,\ldots,n,
\end{aligned}
$$

- If $c = 0$ then $w = 0$ is the optimal solution, no good separation.
- If $c = \infty$ then solution minimizes the hinge loss function. Not regularized, maybe overfi tting.
- The best option is somewhere in between, but only can be found by cross validation.
- The path of solutions for all c is piecewise linear. In theory computing the entire path is as “easy” as solving the problem with an active set method for just one value of c.

Parametric path

$$
\begin{aligned}
&\alpha_i s_i = 0, \quad i = 1, \ldots, n, \\
&(c - \alpha_i)\xi_i = 0 \quad i = 1, \ldots, n, \\
&y^T \alpha = 0, \\
&-Q\alpha + y\beta + s - \xi = -e, \\
&0 \leq \alpha \leq ce, \; s \geq 0, \; \xi \geq 0.
\end{aligned}
$$

Reduced system

$$
\begin{aligned}
&y_s^T \alpha_s = -c(y_c^T e_c), \\
&-Q_{ss}\alpha_s + y_s\beta = -e_s + cQ_{sc}e_c, \\
&0 \leq \alpha_s \leq ce, \\
&s_0 = Q_{0s}\alpha_s - y_0\beta - e_0 - cQ_{0c}e_c, \\
&\xi_c = -Q_{cs}\alpha_s + y_c\beta + e_c + cQ_{cc}e_c.
\end{aligned}
$$

Parametric Algorithm

(i) Find linear functions $\beta(c)$ and $\alpha_s(c)$ satisfying

$$y_s^T \alpha_s = -c y_c^T e,$$
$$-Q_{ss}\alpha_s + y_s \beta = -e + c Q_{sc} e,$$

Parametric Algorithm

(i) Find linear functions $\beta(c)$ and $\alpha_s(c)$ satisfying

$$y_s^T \alpha_s = -c y_c^T e,$$
$$-Q_{ss}\alpha_s + y_s \beta = -e + c Q_{sc} e,$$

(ii) Compute $s_0(c) = -Q_{0s}\alpha_s(c) - y_0\beta(c) + 1 - cQ_{0c}e$
and $\xi_c(c) = Q_{cs}\alpha_s(c) + y_c\beta(c) - 1 + cQ_{cc}e$

Parametric Algorithm

(i) Find linear functions $\beta(c)$ and $\alpha_s(c)$ satisfying

$$y_s^T \alpha_s = -c y_c^T e,$$
$$-Q_{ss}\alpha_s + y_s\beta = -e + cQ_{sc}e,$$

(ii) Compute $s_0(c) = -Q_{0s}\alpha_s(c) - y_0\beta(c) + 1 - cQ_{0c}e$ and $\xi_c(c) = Q_{cs}\alpha_s(c) + y_c\beta(c) - 1 + cQ_{cc}e$

(iii) Find the smallest value $c^* \geq c$ such that $i \in I_s$ $(\alpha_s(c^*))_i = 0$ or $(\alpha_s(c^*))_i = c^*$ or $s_0(c^*)_j = 0$ or $\xi_c(c^*)_k = 0$. If $c^* = \infty$, then **Done.**

Parametric Algorithm

(i) Find linear functions $\beta(c)$ and $\alpha_s(c)$ satisfying

$$y_s^T \alpha_s = -c y_c^T e,$$
$$-Q_{ss}\alpha_s + y_s\beta = -e + cQ_{sc}e,$$

(ii) Compute $s_0(c) = -Q_{0s}\alpha_s(c) - y_0\beta(c) + 1 - cQ_{0c}e$
and $\xi_c(c) = Q_{cs}\alpha_s(c) + y_c\beta(c) - 1 + cQ_{cc}e$

(iii) Find the smallest value $c^* \geq c$ such that $i \in I_s$ $(\alpha_s(c^*))_i = 0$ or $(\alpha_s(c^*))_i = c^*$ or $s_0(c^*)_j = 0$ or $\xi_c(c^*)_k = 0$. If $c^* = \infty$, then **Done.**

(iv) If $(\alpha_s(c^*))_i = 0$, update $I_s = I_s \backslash \{i\}$, $I_0 = I_0 \cup \{i\}$. Go to (i).
If $(\alpha_s(c^*))_i = c^*$, update $I_s = I_s \backslash \{i\}$, $I_c = I_c \cup \{i\}$. Go to (i).

(v) If $s_0(c^*)_j = 0$, update $I_s = I_s \cup \{j\}$, $I_0 = I_0 \backslash \{j\}$. Go to (i).
If $\xi_c(c^*)_k = 0$, update $I_s = I_s \cup \{k\}$, $I_c = I_c \backslash \{k\}$. Go to (i).

Computational comparison

Instance	Initial C	Final C	n	n_s	n_c	Time	Loops	Time	Brk Pnts
letter	0.001	0.1	20000	204	1447	32	3210	292	2004
letter	0.1	10	20000	250	266	11	942	128	645
letter	10	1000	20000	346	8	8	570	102	452
letter	0.001	0.1	20000	40	1489	27	3009	59	1502
letter	0.1	10	20000	204	1447	32	3210	100	925
letter	10	1000	20000	241	39	7	598	125	778
web-7a	0.001	0.1	24692	219	1419	38	3622	11	55
web-7a	0.1	10	24692	261	1303	37	3612	1215	2442
web-7a	10	1000	24692	984	453	86	3351	4300	2642
adult	0.001	0.1	16100	20	7825	112	14379	89	2217
adult	0.1	10	16100	64	6254	89	11758	724	6399
adult	10	1000	16100	483	5219	138	14265	8097	6648

Results for the approximate path

Instance	Initial C	Final C	Time$_{ex}$	Brk Pnts	Time$_{app}$	Brk Pnts
letter	0.001	0.1	292	2004	49	1516
letter	0.1	10	128	645	17	432
letter	10	1000	102	452	15	295
letter	0.001	0.1	59	1502	19	1033
letter	0.1	10	100	925	19	646
letter	10	1000	125	778	17	478

Exact path vs. approximate path