Column Generation for the Minimum Hyperplane Clustering Problem

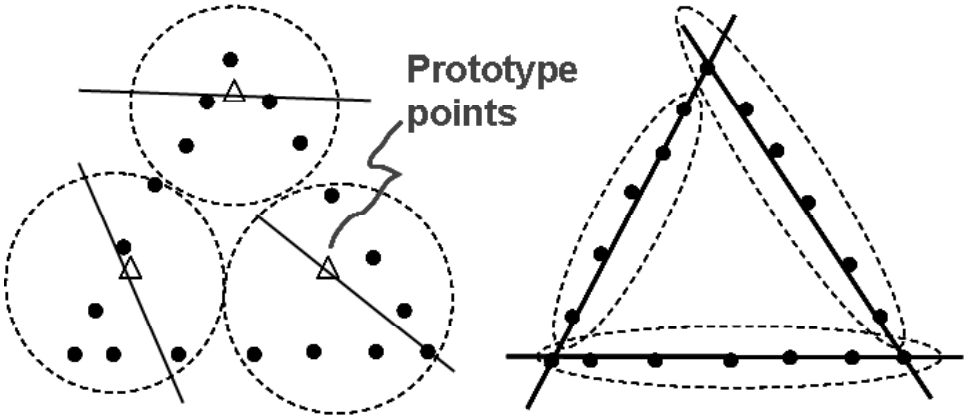
Kanika Dhyani[1]
joint work with Edoardo Amaldi[1] & Alberto Ceselli[2]

[1]Dipartimento di Elettronica e Informazione
Politecnico di Milano

[2]Dipartimento di Tecnologie dell'Informazione
Universita' degli Studi di Milano

Clustering

Classical clustering w.r.t. prototypal points: minimize some dissimilarity measure

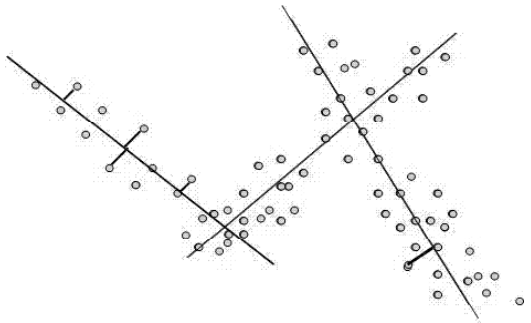


Clustering w.r.t linear subspaces to extract collinearity: Hyperplane clustering, special case with all subspaces of dim. $d - 1$

fit piecewise linear model to data

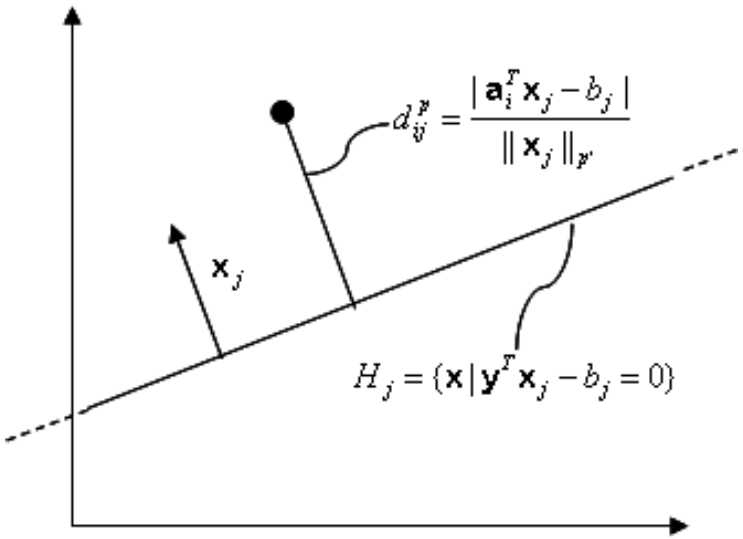
The hyperplane clustering problem

Given $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m | \mathbf{a}_i \in \mathbb{R}^d\}$,find k hyperplanes $\mathcal{H}_j = \{\mathbf{a} \mid \mathbf{a} \in \mathbb{R}^d, \mathbf{a}\mathbf{x}_j = b_j\}, \forall j \leq k$, and an assignment of points so as to minimize some total distance measure f



Objectives: minimize k or f.

Variants depend on objective and f.



Fitting points in $\mathbb{R}^2$ with $k = 2$ is already NP-complete.

Variants

minimize k-HPC: For given fixed k, we want to minimize f ($f = \sum_{i,j} d_{ij}^p$, where $p = 1, 2$).

When $p = 2$ we get **k-plane clustering** [Bradley & Mangasarian 00].

Bottleneck version : minimize $\max_{ij} d_{ij}$.

Also called hyperplane cover problem [Langerman & Morin 05] .

Previous work & applications

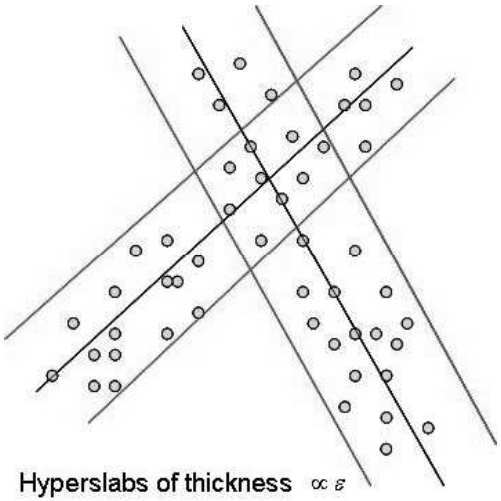
Previous work

- Min-PFS [Amaldi & Mattavelli 02]
- Heuristic for k-plane clustering [Bradley & Mangasarian 00]
- Projective clustering [Agarwal et al. 04]

Applications

- Line detection [Amaldi & Mattavelli 02]
- Piecewise linear model fitting [Amaldi & Mattavelli 02, Trecate et al. 03]
- Data mining (medical data) [Bradley & Mangasarian 00]
- Geometric optimization (shape fitting) [Agarwal et al. 02]

For given fixed $\epsilon > 0$ (max error of deviation), determine minimum # k of hyperplanes $\mathcal{H}_j, j \leq k$, s.t. each point lies within $\pm\epsilon$ deviation from the hyperplane it is assigned to.



Let $\mathbf{a}_j$ be the j-th row of a matrix $\mathbf{A}^{m \times n}$, then we have

$$\mathbf{a}_j \mathbf{x} \leq b_j + \epsilon \text{ and } \mathbf{a}_j \mathbf{x} \geq b_j - \epsilon.$$

Def: A subset of points lying in a hyperslab of thickness proportional to ϵ is called an ϵ-H-cluster.

Formulation

$z_{ij} = 1$ if point i is assigned to ϵ-H-cluster j, else $z_{ij} = 0$
$y_j = 1$ if ϵ-H-cluster j appears in the solution, else $y_j = 0$

$$\min \sum_j y_j$$

$$-(\mathbf{a}_j \mathbf{x} - b_j) \leq \epsilon + M(1 - z_{ij}) \quad \forall i \leq p, \forall j \leq k$$

$$(\mathbf{a}_j \mathbf{x} - b_j) \leq \epsilon + M(1 - z_{ij}) \quad \forall i \leq p, \forall j \leq k$$

$$||\mathbf{x}||_\ell = 1$$

$$\sum_{j=1}^{k} z_{ij} \geq 1 \quad \forall i \leq p$$

$$z_{ij} \leq y_j \quad \forall i \leq p, \forall j \leq k$$

$$\mathbf{x} \in \mathbb{R}^{k \times d}, b \in \mathbb{R}^k$$

$$z_{ij} \in \{0, 1\} \quad \forall i \leq p, \forall j \leq k$$

$$y_j \in \{0, 1\} \quad \forall j \leq k, M \gg 0$$

Formulated *min*-HPC into Master and Pricing sub-problem.

Developed column generation (CG) approach based on SCIP (Mixed Integer Programming (MIP) solver) /AMPL.

Master Problem

Let S be set of all feasible ϵ-H-clusters

$$D_{is} = \begin{cases} 1 & \text{if feasible subsystem } s \text{ contains point } i \\ 0 & \text{else} \end{cases}$$

Variables:

$$y_s = \begin{cases} 1 & \text{if } s \in S \text{ appears in } \epsilon\text{-}H \text{ cluster} \\ 0 & \text{else} \end{cases}$$

Master Problem (MP)

$$\min \sum_{s \in \mathcal{S}} y_s$$

$$\text{s.t.} \sum_{s \in \mathcal{S}} D_{is} y_s \geq 1 \qquad \forall i \leq p$$

$$y_s \in \{0, 1\} \qquad \forall s \in \mathcal{S}$$

l_2 pricing sub-problem

- Tackled l_2 pricing with BONMIN (Nonlinear Mixed Integer programming (MINLP) solver) enforcing $\| \mathbf{x} \|_2 = 1$.

$$
\begin{aligned}
\min(1 - \sum_{i=1}^{p} D_i w_i) & & \\
-\epsilon - M(1 - D_i) &\leq A\mathbf{x} - b \quad \forall i \leq p \\
A\mathbf{x} - b &\leq \epsilon + M(1 - D_i) \quad \forall i \leq p \\
\| \mathbf{x} \|_2 &= 1 \\
D_i &\in \{0, 1\} \forall i \in p, M \gg 0 \\
\mathbf{x} &\in \mathbb{R}^d, b \in \mathbb{R}
\end{aligned}
$$

- Extended the Randomized Thermal Relaxation (RTR) developed by Amaldi et al. (05) to deal with weighted pair of inequalities and enforced $\| \mathbf{x} \|_2 = 1$.

Summary RTR algorithm

Consider infeasible system $\mathbf{Ax} \leq \mathbf{b}$

Start with initial $\mathbf{x}_0 \in \mathbb{R}^d$

i-th iterate violation: $v_i = \mathbf{a}_{l_i}\mathbf{x}_i - b_{l_i}$ if $\mathbf{a}_{l_i}\mathbf{x}_i > b_{l_i}$ and 0 otherwise

$$\text{Update: } \mathbf{x}_{i+1} = \begin{cases} \mathbf{x}_i + \eta_i \mathbf{a}_{l_i} & \text{with probability } p_i \text{ if } \mathbf{a}_{l_i}\mathbf{x}_i < b_{l_i} \\ \mathbf{x}_i & \text{otherwise} \end{cases}$$

where

$$\eta_i = \frac{t_i}{t_0}\exp(-v_i/t_i) \text{ and } p_i = \frac{t_i}{t_0}\exp(-v_i/t_i).$$

Control parameter $t_i \longrightarrow 0$ with iteration

Extended RTR algorithm

Consider $\mathbf{Ax} \leq \mathbf{b} + \epsilon$ and $\mathbf{Ax} \geq \mathbf{b} - \epsilon$

Start from $\mathbf{x}_0 \in \mathbb{R}^d$

$$\text{Violation: } v_i = \begin{cases} -\mathbf{a}_{l_i}\mathbf{x}_i + b_{l_i} + \epsilon & \text{if } \mathbf{a}_{l_i}\mathbf{x}_i - b_{l_i} \leq \epsilon \\ \mathbf{a}_{l_i}\mathbf{x}_i - b_{l_i} + \epsilon & \text{if } -\mathbf{a}_{l_i}\mathbf{x}_i + b_{l_i} \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Update: } \mathbf{x}_{i+1} = \begin{cases} \mathbf{x}_i + \eta_i \mathbf{a}_{l_i} & \text{if } \mathbf{a}_{l_i}\mathbf{x}_i \leq b_{l_i} + \epsilon \\ \mathbf{x}_i - \eta_i \mathbf{a}_{l_i} & \text{if } \mathbf{a}_{l_i}\mathbf{x}_i \geq b_{l_i} + \epsilon \\ \mathbf{x}_i & \text{otherwise} \end{cases}$$

Extended RTR

Speed-up strategies

- Block update: Choosing satisfied inequalities of largest total weight to give update direction.
- Local search : Look for a scalar δ_j s.t. $\mathbf{x} = (x_1, x_2, \ldots, x_j + \delta_j, \ldots, x_n)$ satisfies pairs of inequalities with maximum weight.

Normalization: Updates are renormalized before next iterate.

Good but still slow!!

∞-pricing sub-problem

$\| \mathbf{x} \|_\infty = max_j \mid x_j \mid$

Note that : $\| \mathbf{x} \|_\infty \leq \| \mathbf{x} \|_2$

$$
\begin{aligned}
\min(1 - \sum_{i=1}^{p} D_i w_i) & & \\
-\epsilon - M(1 - D_i) & \leq & A\mathbf{x} - b \ \forall i \leq p \\
A\mathbf{x} - b & \leq & \epsilon + M(1 - D_i) \ \forall i \leq p \\
\| \mathbf{x} \|_\infty & = & 1 \\
D_i & \in & \{0, 1\} \ \forall i \in p \\
M & \gg & 0 \\
\mathbf{x} & \in & \mathbb{R}^d, b \in \mathbb{R}
\end{aligned}
$$

Reformulation of $\| \mathbf{x} \|_\infty = 1$

$$
\begin{aligned}
x_j \geq 1 - 2(1 - u_j) & \quad \forall j \leq d \\
\sum_{j \leq d} u_j = 1 & \\
-1 \leq x_j \leq 1 & \quad \forall j \leq d \\
u_j \in \{0, 1\} & \quad \forall j \leq d
\end{aligned}
$$

Mixed Integer Linear Program (MILP)

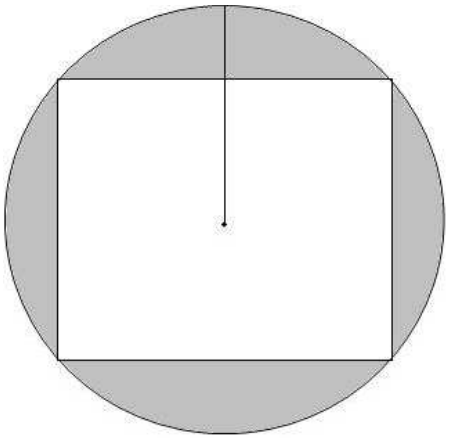
Relaxed pricing sub-problem

$$x_j \geq (1 + \frac{1}{\sqrt{m}})u_j - 1 \quad \forall j \leq d$$

$$x_j \leq -(1 - \frac{1}{\sqrt{m}})v_j + 1 \quad \forall j \leq d$$

$$\sum_{j \leq d}(u_j + v_j) \geq 1$$

$$u_j, v_j \in \{0, 1\} \quad \forall j \leq d$$



Pricing sub-problem along with the above instead of $\| \mathbf{x} \|_\ell = 1$ is a MILP.

Overall greedy algorithm

Used as benchmark for comparison with CG approach.

Strategy : Iteratively extract largest ϵ-h-clusters.

Used approaches developed for pricing sub-problems to determine max ϵ-h-clusters (taking $w_i = 1, 1 \leq i \leq m$).

Depending on normalization different sub-models can be chosen.

Data instances

- UCL repository
  - Wisconsin Breast Cancer ($n = 194, d = 2$)
  - Liver Disorders ($n = 345, d = 6$)
- Random realistic data instances generated in $[0, 1]^d$ with carefully selected ϵ.

Results

Comparison of greedy algorithm solved with various normalizations

	∞-norm		2-norm		Extended RTR	
Instance	time	# of hyp	time	# of hyp	time	# of hyp
25_2	$<$0s	4	$<$ 0s	5	0m2s	3
35_2	$<$0s	4	0m1s	6	0m5s	3
50_2	$<$0s	4	0m1s	6	0m5s	5
60_2	$<$0s	4	0m1s	7	0m6s	3
70_2	$<$0s	4	0m2s	8	0m9s	3
150_8	0m6s	6	0m12s	6	1m1s	4
210_8	0m12s	9	0m12s	6	1m28s	4
250_10	0m20s	8	0m24s	6	2m12s	4
500_4	0m55s	11	0m36s	8	4m57s	6
750_7	2m57s	13	2m50s	10	13m7s	7
750_10	3m42s	12	3m28s	9	12m18s	6
1000_7	6m11s	17	5m9s	11	20m31s	7
1500_6	11m49s	17	7m54s	11	37m57s	7
1500_4	11m 52s	21	8m18s	13	39m14s	7
cancer ($\epsilon = 0.5$)	0m2s	12	0m4s	16	0m47s	5
cancer ($\epsilon = 0.1$)	0m5s	21	0m8s	31	1m10s	6
liver ($\epsilon = 1$)	0m21s	18	0m10s	11	7m39s	15
liver ($\epsilon = 0.5$)	0m24s	24	0m17s	17	12m31s	22

Extended RTR takes longer but provides good solutions

Results

Comparison of CG with greedy algorithm with respective normalizations

	CG						Greedy Approach					
	∞-norm		2-norm		Relaxed		∞-norm		2-norm		Relaxed	
Data	time	# hyp	time	# hyp	time	# hyp	time	# hyp	time	# hyp	time	# hyp
25_2	0m2s	3	<0s	2	<0s	2	<0s	4	<0s	5	<0s	4
35_2	0m9s	3	0m1s	3	<0s	3	<0s	4	0m1s	6	<0s	4
50_2	0m12s	3	0m4s	4	<0s	5	<0s	4	0m1s	5	<0s	4
60_2	0m5s	3	0m4s	3	0m6s	3	<0s	4	0m1s	7	<0s	5
70_2	<0s	3	0m3s	3	0m6s	3	<0s	4	0m2s	8	<0s	5
150_8	0m1s	9	33m19s	4	17m42s	4	0m6s	6	0m10s	6	0m3s	4
210_8	0m3s	7	1m37s	4	1m31s	4	0m12s	9	0m12s	6	0m6s	4
250_10	0m6s	12	164m12s	4	39m53s	4	0m20s	8	0m24s	6	0m9s	4
500_4	0m10s	8	5m38s	5	0m9s	8	0m55s	11	0m36s	8	0m27s	7
750_7	0m37s	11	68m35s	10	0m23s	11	2m57s	13	2m50s	10	1m21s	6
1000_7	1m6s	12	73m3s	7	0m39s	12	6m11s	17	5m9s	11	2m2s	7
1500_4	1m43s	6	40m1s	**7**	1m38s	6	11m49s	17	8m18s	**13**	3m24s	8
cancer ($\epsilon = 0.5$)	0m4s	5	2m18s	**5**	<0s	6	0m2s	12	0m4s	**16**	0m2s	12
cancer ($\epsilon = 0.1$)	0m20s	5	0m3s	**5**	0m7s	6	0m5s	21	0m8s	**31**	0m4s	19

Results

Comparison of CG with initial set of columns from Extended-RTR and those from respective greedy approach

	CG with extended-RTR generated columns						CG with resp. Greedily generated columns					
	∞-norm		2-norm		Relaxed		∞-norm		2-norm		Relaxed	
Data	time	# hyp	time	# hyp	time	# hyp	time	# hyp	time	# hyp	time	# hyp
25_2	0m2s	3	$<$0s	2	$<$0s	2	0m3s	3	1m53s	4	0m3s	4
35_2	0m9s	3	0m1s	3	$<$0s	3	0m1s	4	0m4s	4	0m4s	4
50_2	0m12s	3	0m4s	4	$<$0s	5	0m11s	4	0m10s	5	0m17s	4
60_2	0m5s	3	0m4s	3	0m6s	3	0m6s	4	0m7s	5	0m11s	5
70_2	$<$0s	3	0m3s	3	0m6s	3	0m38s	4	0m16s	6	0m5s	5
150_8	0m1s	9	26m50s	4	17m42s	4	10m0s	6	31m33s	6	3m7s	4
210_8	0m3s	7	1m37s	4	1m31s	4	54m7s	9	14m38s	6	4m34s	4
250_10	0m6s	12	164m12s	4	39m53s	4	26m19s	8	$>$2h	6	6m19s	4
500_4	0m10s	8	5m38s	5	0m9s	8	0m17s	11	0m14s	8	0m6s	7
750_7	0m37s	11	68m35s	10	0m23s	11	2m57s	13	60m32s	10	1m21s	6
750_10	0m58s	10	$>$2h	6	42m10s	6	3m42s	12	598m20s	9	1m22s	5
1000_7	1m6s	12	73m13s	**7**	0m39s	12	6m11s	17	42m27s	**11**	2m2s	7
1500_6	11m49s	17	7m54s	11	4m59s	8	1m19s	17	61m34s	11	74m25s	8
1500_4	1m 43s	6	40m1s	7	1m38s	6	1m7s	21	26m37s	13	0m5s	8
cancer ($\epsilon = 0.5$)	0m4s	5	2m18s	**5**	$<$0s	6	0m2s	12	0m25s	**16**	$<$0s	12

More # of CG iterations needed for CG with initial set of columns from greedy approach.

Concluding remarks

Of course solution is dependent on ϵ.

Greedy heuristic

- Extended-RTR model is computationally heavy but gives good results

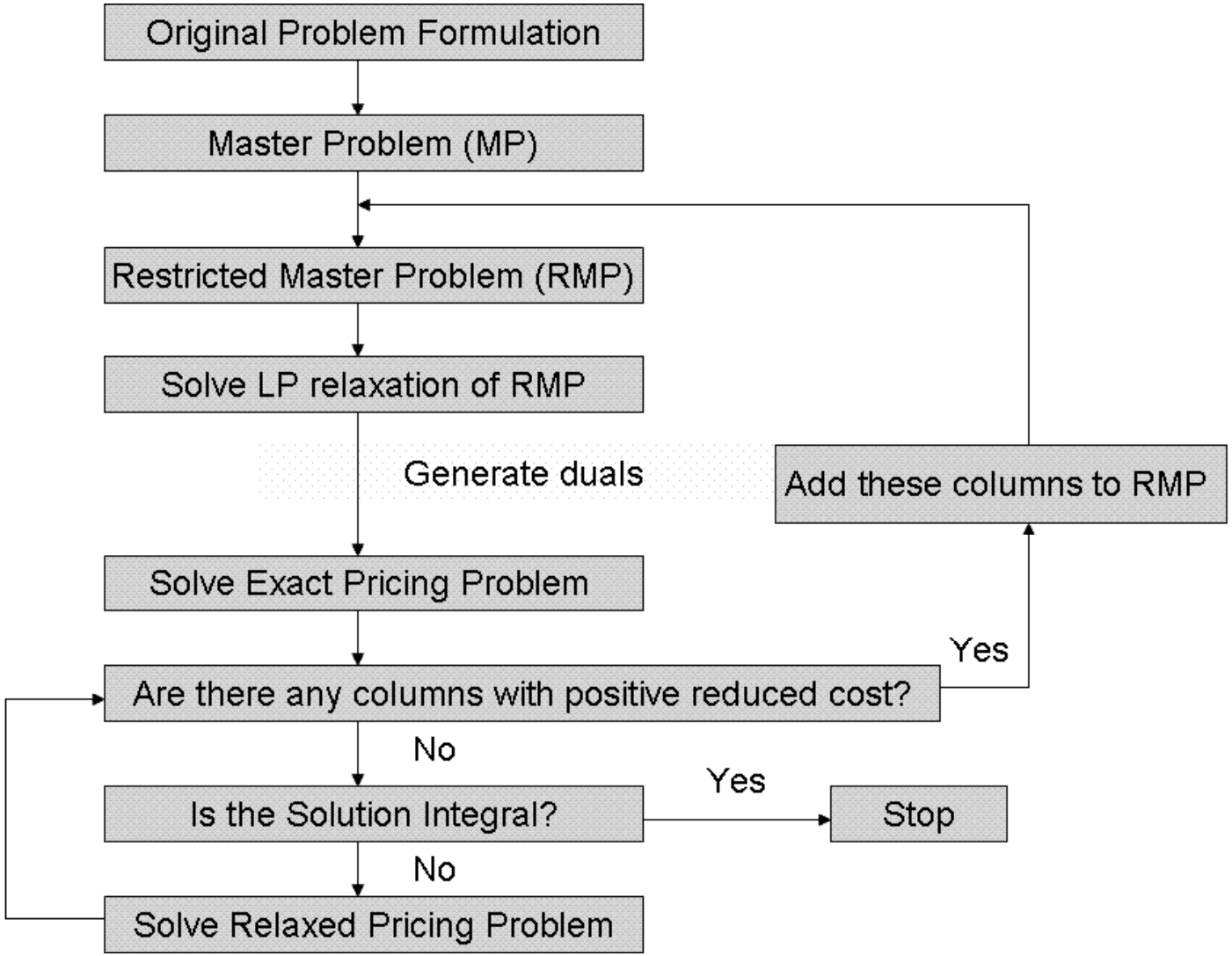
CG more efficient than greedy strategy

- Better results are obtained for all three pricing problems with initial columns generated by Extended-RTR
- l_2 pricing problem yields best solution but at cost of high computation time
- Best variant to get dual bound is relaxed pricing problem with initial columns generated by Extended-RTR

Future Work

- Efficient selection of ϵ.
- Develop refinement procedures based on reassignment points to the hyperslabs (min the sum of 2-norm distances).
- Alternate with different pricing sub-problems.
- Devise branching rules.

Column Generation Schemata



Column Generation Models

Master Problem

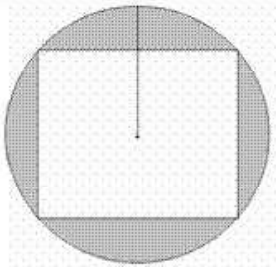
$$\min \sum y_s$$
$$\sum_{s \in S} D_{is} y_s \geq 1 \quad \forall i = 1, \ldots, p$$
$$y_s \in \{0,1\} \quad \forall s \in S$$

Exact Pricer Problem

$$\min(1 - \sum_{i=1}^{p} D_i w_i)$$
$$-\epsilon - M(1 - D_i) \leq Ax - b \quad \forall i \leq p$$
$$Ax - b \leq \epsilon + M(1 - D_i) \quad \forall i \leq p$$
$$\| x \|_2 = 1$$
$$D_i \in \{0,1\} \quad \forall i \in p, M >> 0$$
$$x \in \mathbb{R}^d, b \in \mathbb{R}, M >> 0$$

Relaxed Pricer

$$x \geq (1 + \frac{1}{\sqrt{m}})u - 1$$
$$x \leq (1 - \frac{1}{\sqrt{m}})v + 1$$
$$u + v \geq 1$$
$$u, v \in \{0,1\}^d$$



Infinite Norm Pricer

$$\| x_j \|_\infty = 1 \quad \forall j \leq k$$

Linearized

$$x \geq 1 - 2(1 - u)$$
$$u = 1$$
$$u \in \{0,1\}^d$$

l_2 Pricing Sub-problem-Extended RTR algorithm

Initialization : Pick any $x^0 \in \mathbb{R}^d$,set cycle counter $c = 1$, select maximum number of cycles C and initial T_0

while $c \leq C$ **do**

Initialize set of indices $I = \{1, \ldots, p\}$

repeat cycle through all pairs of inequalities

Pick index k_i, equiprobably and without replacement from I

Compute violation $v_i^{k_i}$ for the k_ith pair of inequalities

set $T = \left(1 - \frac{c}{C}\right) \cdot T_0$ and $\eta_i = \frac{T}{T_0} exp\left(\frac{v_i^{k_i}}{T}\right)$

if $\mathbf{a}^{k_i}\mathbf{x}^i \geq b_{k_i} + \epsilon$ **then** $\mathbf{x}^{i+1} = \mathbf{x}^i - \eta_i \mathbf{a}^{k_i}$

else if $\mathbf{a}^{k_i}\mathbf{x}^i \leq b_{k_i} - \epsilon$ **then** $\mathbf{x}^{i+1} = \mathbf{x}^i + \eta_i \mathbf{a}^{k_i}$

else $\mathbf{x}^{i+1} = \mathbf{x}^i$

set $I = I - \{k_i\}$

until $I = \emptyset$

Update T_0 and set $c = c + 1$

end

return $x^{p.C}$

	CG with extended-RTR generated columns						CG with resp. Greedily generated columns					
	∞-norm		2-norm		Relaxed		∞-norm		2-norm		Relaxed	
Data	time	# hyp	time	# hyp	time	# hyp	time	# hyp	time	# hyp	time	# hyp
25_2	0m2s	3	$<$0s	2	$<$0s	2	0m3s	3	1m53s	4	0m3s	4
35_2	0m9s	3	0m1s	3	$<$0s	3	0m1s	4	0m4s	4	0m4s	4
50_2	0m12s	3	0m4s	4	$<$0s	5	0m11s	4	0m10s	5	0m17s	4
60_2	0m5s	3	0m4s	3	0m6s	3	0m6s	4	0m7s	5	0m11s	5
70_2	$<$0s	3	0m3s	3	0m6s	3	0m38s	4	0m16s	6	0m5s	5
150_8	0m1s	9	26m50s	4	17m42s	4	10m0s	6	31m33s	6	3m7s	4
210_8	0m3s	7	1m37s	4	1m31s	4	54m7s	9	14m38s	6	4m34s	4
250_10	0m6s	12	164m12s	4	39m53s	4	26m19s	8	$>$2h	6	6m19s	4
500_4	0m10s	8	5m38s	5	0m9s	8	0m17s	11	0m14s	8	0m6s	7
750_7	0m37s	11	68m35s	10	0m23s	11	2m57s	13	60m32s	10	1m21s	6
750_10	0m58s	10	$>$2h	6	42m10s	6	3m42s	12	598m20s	9	1m22s	5
1000_7	1m6s	12	73m13s	7	0m39s	12	6m11s	17	42m27s	11	2m2s	7
1500_6	11m49s	17		11	4m59s	8	1m19s	17	61m34s	11	74m25s	8
1500_4	11m 52s	21	40m1s	7	3m24s	8	1m7s	21	26m37s	13	0m5s	8

Results

Comparison of greedy algorithm solved with l_2 norm

	2-norm		Extended RTR	
Instance	time	# of hyp	time	# of hyp
syntetic_25_2	$<$ 0s	5	0m2s	3
syntetic_35_2	0m1s	6	0m5s	3
syntetic_50_2	0m1s	6	0m5s	5
syntetic_60_2	0m1s	7	0m6s	3
syntetic_70_2	0m2s	8	0m9s	3
syntetic_150_8	0m4s	6	1m1s	4
esr_210_8	0m22s	6	1m28s	4
esr_250_10	0m19s	6	2m12s	4
sr_500_4	0m28s	8	4m57s	6
sr_750_7	1m11s	10	13m7s	7
sr_750_10	1m23s	9	12m18s	6
sr_1000_7	2m12s	11	20m31s	7
sr_1500_6	3m28s	11	37m57s	7
sr_1500_4	3m34s	13	39m14s	7
cancer ($\epsilon = 0.5$)	0m3s	16	0m47s	5
cancer ($\epsilon = 0.1$)	0m7s	31	1m10s	6
liver ($\epsilon = 1$)	0m11s	11	7m39s	15
liver ($\epsilon = 0.5$)	0m17s	17	12m31s	22

Extended RTR takes longer but provides good solutions