k-hyperplane clustering problem: column generation and a metaheuristic

Edoardo Amaldi, [a] Stefano Coniglio, [a] Kanika Dhyani [b]

[a] *Dipartimento di Elettronica e Informazione, Politecnico di Milano*
Piazza L. da Vinci 32, 22133, Milano
{amaldi,coniglio}@elet.polimi.it

[b] *Neptuny, Politecnico di Milano*
Via G. Durando 10, 20158, Milano
dhyani@elet.polimi.it

Key words: hyperplane clustering, column generation, metaheuristic

1. Introduction

In the k-*Hyperplane Clustering* problem (k-HC), given a set of m points $P = \{\underline{a}_1, \ldots, \underline{a}_m\}$ in $\mathbb{R}^n$, we have to determine k hyperplanes $H_j = \{\underline{a} \in \mathbb{R}^n \mid \underline{w}_j^T \underline{a} = \gamma_j,\ \underline{w}_j \in \mathbb{R}^n,\ \gamma_j \in \mathbb{R}\}$, with $1 \leq j \leq k$, and assign each point to a single H_j, thus partitioning P into k *h-clusters*, so as to minimize the sum-of-squared 2-norm orthogonal distances d_{ij} from each point to the corresponding hyperplane, where $d_{ij} = \frac{|\underline{w}_j^T \underline{a}_i - \gamma_j|}{\|\underline{w}_j\|_2}$.

k-HC naturally arises in many areas such as data mining [3], operations research [7], line detection in digital images [2] and piecewise linear model fitting [4]. k-HC is $\mathcal{NP}$-hard, since it is $\mathcal{NP}$-complete to decide whether k lines can fit m points in $\mathbb{R}^2$ with zero error [7]. In [3] Bradley and Mangasarian propose a heuristic for k-HC which extends the classical k-means algorithm to the hyperplane case.

The bottleneck version of k-HC in which, given a maximum deviation tolerance ϵ, k is minimized, has been studied in [5]; it is closely related to the MIN-PFS problem of partitioning an infeasible linear system into a minimum number of feasible subsystems [2]. Variants of k-HC where linear subspaces of different dimensions are looked for are also considered, e.g., in [1].

In this work we propose a column generation algorithm and an efficient metaheuristic.

2. Column generation algorithm (CG)

We consider the following set covering master problem (MP):

$$\begin{aligned} \min \quad & \sum_{s\in\mathcal{S}} d_s y_s && && (2.1)\\ s.t. \quad & \sum_{s\in\mathcal{S}} \tilde{x}_{is} y_s \geq 1 && 1 \leq i \leq m && (2.2)\\ & \sum_{s\in\mathcal{S}} y_s \leq k && && (2.3)\\ & y_s \in \{0,1\} && s \in \mathcal{S}, && \end{aligned}$$

where $\mathcal{S}$ is the set of (exponentially many) feasible h-clusters and, for each $s \in \mathcal{S}$, the variable y_s equals 1 if the h-cluster s is in the solution and 0 otherwise. The parameter d_s is the sum-of-squared 2-norm orthogonal distances to the hyperplane H_s of the points contained in the h-cluster s and the parameter $\tilde{x}_{is}$ equals 1 if the point $\underline{a}_i$ is contained in h-cluster s, and 0 otherwise.

We tackle this formulation with a column generation approach. Let $\mathcal{S}' \subset \mathcal{S}$ be the initial pool of columns. Let π_i and μ be the dual variables of constraints (2.2) and (2.3), corresponding to an optimal solution of (MP) when restricted to the only columns in $\mathcal{S}'$ and with the integrality constraints relaxed. The column $s \notin \mathcal{S}'$ with the largest negative reduced cost $\bar{c}_{s'}$, with $\bar{c}_{s'} = d_{s'} - \sum_{i=1}^{m} \pi_i x_{is'} - \mu$, can be obtained by solving the following nonlinear 2-norm pricing problem (PP):

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} ((\underline{w}^T \underline{a}_i - \gamma)^2 - \pi_i) x_i - \mu && && (2.4)\\ s.t. \quad & \|\underline{w}\|_2 \geq 1 && && (2.5)\\ & x_i \in \{0,1\} && 1 \leq i \leq m && (2.6)\\ & \underline{w} \in \mathbb{R}^n, \gamma \in \mathbb{R}, && && \end{aligned}$$

where $(\underline{w}, \gamma)$ are the parameters of $H_{s'}$ and the binary variable x_i is equal to 1 if the point $\underline{a}_i$ is assigned to $H_{s'}$ and 0 otherwise. Note that x_i takes integer values in any optimal solution and hence (2.6) can be relaxed into $x_i \in [0,1]$. (PP) is nonconvex due to (2.5) and can be solved to local optimality with state-of-the-art nonlinear programming solvers such as SNOPT.

Since $\sqrt{n}\,\|\underline{w}\|_2 \geq \|\underline{w}\|_1$, substituting $\|\underline{w}\|_1 \geq 1$ for (2.5) yields $\|\underline{w}\|_2 \geq \frac{1}{\sqrt{n}}$ and thus a relaxation of k-HC. This 1-norm constraint can be linearized with standard techniques, see [5]. Given any 1-norm solution, a corresponding feasible 2-norm solution can be obtained by fixing the point-to-hyperplane assignment and recomputing the hyperplane parameters in the closed-form proposed in [3].

To speed up convergence, a dual-stabilization technique [8] is used. At each iteration t that is a multiple of the frequency parameter f, the current dual vector

$\underline{\pi}_t$ is replaced by a convex combination $\underline{\pi}'_t$ of $\underline{\pi}_t$ and the previous dual vector $\underline{\pi}_{t-1}$, namely $\underline{\pi}'_t := \eta\underline{\pi}_t + (1-\eta)\underline{\pi}_{t-1}$ with $\eta \in (0,1)$.

3. Point-Reassignment metaheuristic (PR)

Our Point-Reassignment metaheuristic (PR) relies on a simple criterion to identify, at each iteration, points which are likely to be *ill-assigned* in the current solution, based on the distance ratio $\frac{d_{ij}}{\min_{j'\neq j} d_{ij'}}$.

Starting from a randomly generated solution, at each iteration the set I of possibly ill-assigned points is identified as follows. Let m_j, with $1 \leq j \leq k$, be the number of points currently assigned to hyperplane H_j and rank them w.r.t. the ratio $\frac{d_{ij}}{\min_{j'\neq j} d_{ij'}}$. Indeed, points with large ratio have larger distance w.r.t. H_j and are close to another hyperplane $H_{j'}$, hence being more likely to be ill-assigned. Given a control parameter α ("temperature"), the set I then contains the $\alpha \cdot m_j$ points of each cluster with the largest ratio.

A move consists in assigning each point $\underline{a}_i$ in I to the closest hyperplane which differs from the current one $\underline{a}_i$ is assigned to, and in assigning the points in $P \setminus I$ to the closest H_j, if it is not already the case. The hyperplane parameters are then recomputed in the closed-form described in [3]. To avoid cycling and try to escape from local minima, we adopt two Tabu Search features (see e.g. [6]): a list of forbidden moves of length l and a partial aspiration criterion.

Since early solutions are expected to have a larger number of ill-assigned points, the control parameter α is initially set to a large enough value α_0 and is then progressively decreased to stabilize the search process, thus progressively reducing the variability that is introduced at each iteration. More precisely, α is updated as $\alpha_t = \alpha_0\rho^t$, where t is the index of the current iteration and $\rho \in (0,1)$ determines the speed at which α is driven to 0. When $\alpha = 0$, I becomes empty and, after all points are assigned to the closets H_j, the algorithm terminates in a local minimum. The best solution found is stored and returned.

4. Computational results and conclusions

We compare CG and PR with a multi-start version of Bradley and Mangasarian's algorithm (BM, [3]) on a set of challenging, randomly generated instances [4]. CG is implemented in AMPL using SNOPT and CPLEX as solvers. PR and BM are implemented in C++. Tests are run on an Intel Xeon machine, with 2.8 GHz and 2 GB RAM, equipped with Linux and gcc 4.1.2.

CG is tested on 8 instances with $m = 20 - 70$, $n = 2 - 3$ and $k = 3 - 6$. η is set to 0.7 and is reduced to 0.4 when 90% of the dual variables become zero. The frequency f is set to 5. Because of the nonlinearities in the 2-norm pricing problem, CG with 2-norm gets often stuck in local minima and leads to poor quality solutions. CG with 1-norm finds solutions with very small objective function values, but since the 1-norm pricing problem is a non-trivial mixed-integer linear program, the overall computation time scales poorly with the size of the instances.

PR is tested on 95 instances with $m = 100 - 2500$, $n = 2 - 6$ and $k = 3 - 8$. Parameters are set to $\alpha_0 = 0.9$, $\rho = 0.6$, $l = 2$. We compare the best solutions found by running PR and BM for a fixed amount of time and restarting them from randomly generated solutions each time a local minimum is found. The time limit is set to 120 and 180 seconds for instances with up to, respectively, 750 points and 2500 points. PR finds better results in 89 cases out of 95 (with strictly better solutions in 59 cases). On average, BM yields solutions worse than those found by PR by a factor of 25%. Neglecting the 30 instances for which both algorithms find solutions of the same value (since they may be optimal), the factor amounts to 35.5%.

References

[1] P. K. Agarwal and C. M. Procopiouc, *Approximation algorithms for the projective clustering*, Journal of Algorithms **46** (2003), no. 2, 115–139.

[2] E. Amaldi and M. Mattavelli, *The* MIN PFS *problem and piecewise linear model estimation*, Discrete Appl. Math. **118** (2002), 115–143.

[3] P.S. Bradley and O.L. Mangasarian, *k-plane clustering*, J. of Global Optimization **16** (2000), no. 1, 23–32.

[4] S. Coniglio and F. Italiano, *Hyperplane clustering and piecewise linear model fitting*, Master's thesis, DEI, Politecnico di Milano, 2007.

[5] K. Dhyani, *Optimization models and algorithms for the hyperplane clustering problem*, Ph.D. thesis, Politecnico di Milano, 2009.

[6] F. Glover and M. Laguna, *Tabu search*, Kluwer Academic Publishers, 1997.

[7] N. Megiddo and A. Tamir, *On the complexity of locating linear facilities in the plane*, Operations Research Letters **1** (1982), 194–197.

[8] A. Pessoa, E. Uchoa, M. Poggi de Aragão, and R. Rodrigues, *Algorithms over arc-time indexed formulations for single and parallel machine scheduling problems*, Tech. Report 8, Universidade Federal Fluminense, 2008.