Column Clustering Problem

Business background and requirements

Large corporations internally process hundreds of thousands of documents. Document tagging is a critical business requirement. While a classification system for documents exists, the specific set of tags has not yet been defined.

Feature engineering enables the representation of the text corpus as a matrix, where each column represents a potential tag (feature) and each row represents a document. A natural approach is to train a classification method by selecting a subset of columns for each class, allowing the class of a document to be estimated by maximizing the weight sum over the selected columns.

Ideal tags, as per the business requirements, are mutually exclusive and should enable perfect classification using the described method. There is a Ground Truth data that allows to train the classification in an offline mode, then, use it online. This problem is reformulated below as a Discrete Optimization problem.

Less formal problem statement

Pick a small, unique set of columns (50–250) for each of the K clusters—no column reused across clusters, although some columns may be dropped (not included in any cluster). For any data row, sum its values over each cluster's column set; the cluster with the highest sum "wins" and becomes the prediction. Choose column sets that maximize how often the winning cluster matches the row's true label.

Objective

Select disjoint column subsets—one per cluster—to maximize classification accuracy using a simple scoring rule.

Notation

1. Data matrix: A ∈ $\mathbb{R}$^(N×M), with $a_{ij} \geq 0$ and $\|a_i\|_2 = 1$ for all rows i;
2. Clusters: K classes with true labels $y_i \in \{1,\dots,K\}$;
3. Binary variables: $x_{i,q} = 1$ if column j is assigned to cluster q, else 0.

Constraints

1. Cardinality per cluster: $F_{min} \leq \sum_{j=1}^{M} x_{j,q} \leq F_{max}$, for all q = 1,…,K;
2. Disjointness (no column reuse): $\sum_{q=1}^{K} x_{j,q} \leq 1$, for all j = 1,…,M.

Classification rule

1. For row i, compute cluster scores:

$$s_i^q = \sum_{j=1}^{M} a_{ij} \cdot x_{j,q}, \quad q = 1, \dots, K;$$

2. Assign to cluster with highest score (break ties arbitrarily):

$$\hat{y}_i = argmax_q(s_i^q).$$

Optimization problem

Maximize classification accuracy:

$$max_{(x)}\ (1/N)\ \cdot\ \sum_{i=1}^{N} \mathbb{1}\{\ \hat{y}_i\ =\ y_i\ \},$$

subject to the cardinality and disjointness constraints above, where $\mathbb{1}\{\cdot\}$ is the indicator function (1 if true, 0 otherwise).

Input data

Matrix of features A (N=498481 X M=10000) is provided in json format using sparse representation; True labels $y_i \in \{1,\ldots,K\}$, K=61 for each row i are also provided (column "class" in a given matrix).

Minimum success criteria

1. Minimum final classification accuracy: 0.58
2. Cardinality per cluster: $F_{min} = 50\ \leq\ \sum_{j=1}^{M} x_{j,q}\ \leq\ F_{max} = 250$, for all q = 1,…,K should be strictly followed
3. Maximum algorithm execution time when executed single-threaded: 24 hours (CPU @ 3.00 GHz, 32 GB RAM), if GPU or NPU utilized, limit VRAM usage to 8 GB

Preferences

Solution with the highest classification accuracy wins. Numerical efficacy would be the second evaluation criteria. We would also value estimate of an upper bound on the optimum for classification accuracy and proof of convergence to optimum.