A Simple Feature Extraction for High Dimensional Image Representations*

Christian Savu-Krohn and Peter Auer

Chair of Information Technology (CIT), University of Leoben, Austria
{csavukrohn,auer}@unileoben.ac.at

Abstract. We investigate a method to find local clusters in low dimensional subspaces of high dimensional data, e.g. in high dimensional image descriptions. Using cluster centers instead of the full set of data will speed up the performance of learning algorithms for object recognition, and might also improve performance because overfitting is avoided. Using the Graz01 database, our method outperforms a current standard method for feature extraction from high dimensional image representations.

1 Introduction

One of the key requirements to a modern Cognitive Vision System is a robust performance upon changes in illumination, scale, pose etc. For this, modern feature extraction methods like Lowe's Scale-Invariant-Feature-Transforms [1] and Mikolajczyk's and Schmid's Scale-Invariant-Harris-Laplace- [2] and Affine-Invariant-Interest-Point-Detectors [3] come to hand when learning objects or object categories [4, 5]. Opelt *et al.* [4] used AdaBoost [6, 7] to generate a combination of weak hypotheses, where each weak hypothesis consists of a feature vector with a distance threshold. Since the number of feature vectors is very large, the search for weak hypotheses becomes computationally very expensive. To reduce the computational burden, we want to reduce the number of candidates for weak hypotheses by clustering. This would reduce learning time significantly. Furthermore, this promises less overfitting and, eventually, more accurate classifiers. Note that both, Opelt *et al.* [4] and Dance *et al.* [5] use k-means for that purpose.

Unfortunately, modern descriptors like Lowe's SIFTs reside in high dimensional space for which common metrics like the euclidean might be unsuitable [8]. Therefore, dimensionality reduction techniques such as PCA are commonly applied before clustering. Nevertheless, if the specific clusters reside in various subspaces such global reduction techniques may be inappropriate. Recently, projective clustering methods address this problem by searching for local subspace clusters [9, 10]. Aggarwal *et al.* [9] determine the local subspaces through the smallest eigenvectors of each clusters covariance matrix, whereat the user has to predefine the minimum number of subspace dimensions. Böhm *et al.* [10] propose a density connected clustering algorithm, which searches for

* This work was presented in a preliminary version at the First Austrian Cognitive Vision Workshop (ACVW 05), Zell an der Pram, January 2005.

C. Saunders et al. (Eds.): SLSFS 2005, LNCS 3940, pp. 163–172, 2006.

variances below a certain threshold along the attributes to identify subspaces within ϵ-neighborhoods of points. Using another parameter, they limit the number of admissible subspace dimensions from above.

Unfortunately, the number of subspace dimensions is commonly unknown beforehand. Furthermore, the metrics employed by Aggarwal *et al.* [9] and Böhm *et al.* [10] may deteriorate upon high dimensional subspaces. Therefore, we propose a fast projective clustering algorithm (FPC), which aims to find axis-parallel subspace-clusters while determining the number of subspace dimensions automatically. Our approach is to search for the interval of highest density along all coordinate axes recursively [Sec. 2]. Since our actual goal is feature extraction for learning from high dimensional image representations the evaluations are twofold. First, we compare our method to k-means in an unsupervised setting using artificial data [Sec. 3.1]. Then, we evaluate our method within a boosting frame work, enabling us to directly compare our results to those of Opelt *et al.* [4] using k-means [Sec. 3.2].

2 The Clustering Algorithm

Assume that the features reside in $\mathbb{R}^n$. Then the densest interval $[a, b]$ along each coordinate $l \in \{1, ..., n\}$ is calculated. (The details of this calculation are given in the next section.) For the coordinate with the overall densest interval, all data points with a corresponding coordinate in this interval are selected and processed by a recursive application of the algorithm [App. A]. The algorithm terminates if no meaningful dense intervals can be found. The final result of one iteration of the algorithm is a subspace cluster defined by the hyper-rectangle of the recursively chosen coordinates and intervals. When such a cluster is found, the data points in this cluster are removed and the algorithm is restarted for the remaining data. The overall algorithm terminates if no more clusters can be found.

2.1 Calculating the Densest Interval Along a Single Coordinate

Let $\mathbf{x} = (x_1 \leq ... \leq x_m) \in \mathbb{R}$ be an ordered dataset with diameter $r = \max \mathbf{x} - \min \mathbf{x}$. To calculate the densest interval we assume that the data are drawn from a probability density function

$$f\left(x \middle| a, b\right) = \frac{1-q}{r} + 1_{[a,b]}(x)\frac{q}{b-a} \tag{1}$$

with unknown $[a, b]$ and q, and the indicator function

$$1_{[a,b]}(x) = \left\{ \begin{array}{l} 1,\, x \in [a,b] \\ 0,\, else \end{array} \right\} \tag{2}$$

Choosing the parameters which maximize the likelihood of the data we find the desired interval. Optimizing the log-likelihood LL for q we find

$$LL\left(\mathbf{x}|a,b\right) = (m-\xi)\cdot\log\left(\frac{1-\frac{\xi}{m}}{r-(b-a)}\right) + \xi\cdot\log\left(\frac{\frac{\xi}{m}}{b-a}\right) \tag{3}$$

with $\xi = |\{x|a \leq x \leq b\}|$. Thus, it remains to select a and b such that $LL(\mathbf{x}|a,b)$ is maximized. Thereby, we restrict ξ as follows.

Viewing clustering as learning an indicator function in an unsupervised setting, we consult the supervised case to infer a reasonable sample bound. Within the agnostic learning framework [11, 12] samples $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$ are drawn randomly from a joint distribution D over $\mathbb{R}^n \times \{1, ..., \#classes\}$, and the learner's goal is to output a hypothesis $\hat{h} \in \mathcal{H}$ such that for another $(\mathbf{x}, y)$ the probability $P(\hat{h}(\mathbf{x}) \neq y)$ is almost that of the best $h \in \mathcal{H}$:

$$\mathbb{P}\left[P\left(\hat{h}(\mathbf{x}) \neq y\right) - \min_{h \in \mathcal{H}} P\left(h(\mathbf{x}) \neq y\right) \leq \epsilon\right] \geq 1 - \delta \tag{4}$$

with $\mathbb{P}$ the probability of drawing $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$. For this setting it has been shown [13] that one has to sample

$$m \sim \frac{1}{\epsilon^2}\left(VCdim(\mathcal{H}) - \log\delta\right) \tag{5}$$

data points. That is, even if labels were given, the bounds $[a, b]$ of the interval remain imprecise by

$$m \cdot \epsilon = m \cdot \sqrt{(VCdim(\mathcal{H}) - \log\delta)/m} \approx \sqrt{m} \tag{6}$$

data points compared to the best interval possible. Therefore, it is reasonable to use

$$s_{min} \leq \xi \leq m - s_{min}\ , \tag{7}$$

with $s_{min} = \sqrt{m}$, as a validation criterion for a solution $[a, b]$.

This choice of s_{min} guarantees a linear runtime at a maximum level of granularity when maximizing the Likelihood [Eq. 3] using the following exhaustive search procedure. Let

$$\begin{aligned} \Phi = &\left\{\varphi | \varphi = \frac{x_i + x_{i+1}}{2}, 1 \leq i \leq m, x_i \neq x_{i+1}\right\} \\ &\cup \{\min \mathbf{x}, \max \mathbf{x}\}\ , \end{aligned} \tag{8}$$

where $\mathbf{x} = (x_1 \leq ... \leq x_m) \in \mathbb{R}$ and $\varphi_1 < ... < \varphi_{|\phi|} \in \Phi$, denote the ordered set of possible interval bounds. Since an exhaustive search over Φ would require $O(m^2)$ steps, we use the following heuristic search (which might return a suboptimal interval). In a first phase, we determine a coarse optimal solution by exhaustively searching among the subset $\hat{\Phi} \subseteq \Phi$ of bounds, that pairwise enclose the minimum number s_{min} of points. Therefore, we denote

$$\hat{\Phi} = \{\hat{\varphi}_1 < ... < \hat{\varphi}_{|\hat{\Phi}|}\} \tag{9}$$

as the set of coarse bounds $\hat{\varphi}$ with

$$\begin{aligned} \hat{\varphi}_1 &= \min\ \Phi = \min(\mathbf{x}), \\ \hat{\varphi}_i &= \min\left\{\varphi \mid \varphi > \hat{\varphi}_{i-1}, \#\varphi \geq \#\hat{\varphi}_{i-1} + s_{min}\right\}, \\ \#\varphi &= |\{x \leq \varphi\}|\ . \end{aligned}$$

That is, we start at the lowest possible bound and iteratively define a new coarse bound after passing s_{min} points. Note, that we allow for the last interval to contain less than s_{min} data points. Using $\hat{\Phi}$, we get a coarse solution by exhaustive search

$$(\hat{\varphi}_i, \hat{\varphi}_j) = \underset{\hat{\varphi}_{i'} < \hat{\varphi}_{j'}}{\arg\max}\ LL(\mathbf{x} \mid \hat{\varphi}_{i'}, \hat{\varphi}_{j'}) \tag{10}$$

Note, that under the initial assumption of ordered data this computation takes $O(m)$. Then, in the second phase, we refine the coarse solution at the granular level. Although the bounds are imprecise according to Equation 6, we are interested in the empirical maximum of the Likelihood [Eq. 3] and, thus, maximize

$$(\varphi_i, \varphi_j) = \underset{(\varphi_{i'}, \varphi_{j'}) \in \Phi^{\hat{\varphi}_i} \times \Phi^{\hat{\varphi}_j}}{\arg\max} LL(\mathbf{x} \mid \varphi_{i'}, \varphi_{j'}) \tag{11}$$

using

$$\begin{aligned} \Phi^{\hat{\varphi}_i} &= \{\varphi \mid \hat{\varphi}_{i-1} \leq \varphi \leq \hat{\varphi}_{i+1}\} \\ \Phi^{\hat{\varphi}_j} &= \{\varphi \mid \hat{\varphi}_{j-1} \leq \varphi \leq \hat{\varphi}_{j+1}\} \end{aligned} \tag{12}$$

as the set of possible bounds around the coarse bounds. Again, the computation takes $O(m)$ time. Note, that the likelihood for a bimodal distribution is also maximized for (φ_i, φ_j) enclosing the valley between the two modes. Thus, using

$$F(a, b, r, m) = \frac{|\{x : a \leq x \leq b\}|}{m} \cdot \frac{r}{b - a} \tag{13}$$

we select the densest interval (a, b) to

$$(a, b) = \underset{(a', b')}{\arg\max}\ F(a', b', r, m) \tag{14}$$

with $(a', b') \in \{(\min \mathbf{x}, \varphi_i), (\varphi_i, \varphi_j), (\varphi_j, \max \mathbf{x})\}$. To further refine the selected interval we rerun the algorithm on the data from it until no further valid subinterval [Eq. 7] can be selected. Consequently, at least s_{min} data points are removed within each iteration, which yields a total run time of at most $O(m^{3/2})$.

2.2 Processing High Dimensional Data

Assume the densest interval (a, b) along each coordinate has been calculated [Sec. 2.1] and let Λ denote the set of all coordinates along which there exists a valid refinement of the data [Eq. 7]. Thus, we determine the clusters $\mathbf{C} = \{\mathbf{x} : \mathbf{a}_l \leq \mathbf{x}_l \leq \mathbf{b}_l\}$ and $\bar{\mathbf{C}} = \mathbf{X} \setminus \mathbf{C}$ by selecting that coordinate $l \in \Lambda$ which holds the densest interval among all coordinates

$$l = \underset{l' \in \Lambda}{\arg\max}\ F(\mathbf{a}_{l'}, \mathbf{b}_{l'}, \rho_{l'}, m) \tag{15}$$

with ρ the diameter of the full data set. Then, we recurse upon the data in $\mathbf{C}$ by recalculating $(\mathbf{a}, \mathbf{b})$ and Λ until $\Lambda = \{\}$, i.e. there are no more valid coordinates along which cluster $\mathbf{C}$ can be bounded. Consequently, $\mathbf{C}$ is stored, and the algorithm [App. A] restarts upon the remaining data. Finally, the input data is partitioned into a set of subspace clusters, whereat each cluster denotes its constituting bounds in their particular subspace.

3 Evaluation

3.1 Artificial Data

In a first step, we have evaluated our approach on a artificial 25-dimensional dataset containing $k = 19$ axis-parallel clusters $\mathcal{C}$ located in 2-dimensional subspaces. We build the clusters by sampling an increasing number of points $\{100, 150, ..., 1000\}$ from 25-dimensional gaussians $\mathcal{N}(\mathbf{0}, \boldsymbol{\sigma})$ with $\boldsymbol{\sigma} = \mathbf{1}$ except for 2 randomly chosen dimensions with $\sigma = 0.1$. Furthermore, we added 25% of uniformly distributed noise within the range of the data. Let $\hat{\mathcal{C}}$ denote the clustering obtained from a particular clustering method.

We evaluate the quality of a clustering obtained in terms of how well the known clusters are covered. Particularly, we assume that every cluster $\hat{\mathbf{C}}_j$ belongs exactly to one cluster $\mathbf{C}_i$ and, thus, calculate the coverage

$$p = \frac{\sum\limits_{j} |\mathbf{C}_{\tau(j)} \cap \hat{\mathbf{C}}_j|}{\sum\limits_{j} |\hat{\mathbf{C}}_j|}$$

$$\tau(j) = \arg\max_{i} |\mathbf{C}_i \cap \hat{\mathbf{C}}_j|$$

with $i = 1, ..., k$, $j = 1, ..., \hat{k}$ and $\tau = \{1, ..., k\}^{\hat{k}}$.

Unlike k-means, FPC produces a deterministic output upon a certain input. Hence, we compare our clustering of the data to multiple rounds of k-means, each varying in $\hat{k}$ and the seeds sampled from the data at random. Figure 1 shows that FPC outperforms k-means.

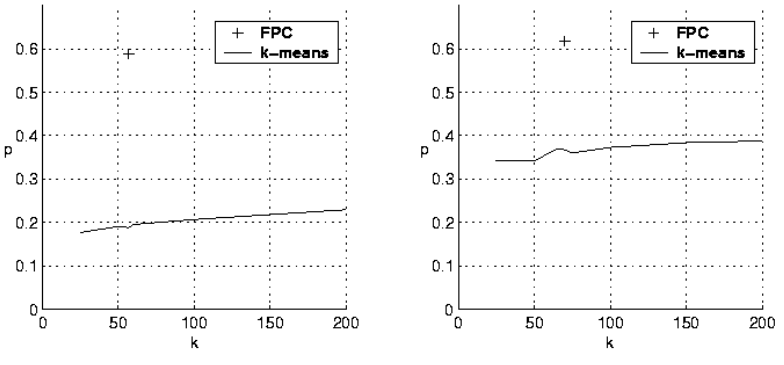

Fig. 1. Coverage of real clusters by the clustering found. Left: at zero noise. Right: at 25% noise level.

3.2 Graz01 Database

Furthermore, we tested our method for image categorization using the Graz01 database[1] using LPBoost [14] as learning method. Particularly, we used the 128-dimensional

[1] Available at http://www.emt.tugraz.at/~pinz/data/

SIFTs extracted by Opelt *et al.* [4] from 300 images of the categories 'bike' and 'background', and retained the SIFTs from 50 images per class for testing. Applying FPC on the training set, we obtained 459 subspace clusters from about 400000 SIFTs. It turned out that all clusters were bounded in at most 18 coordinates, indicating the typically

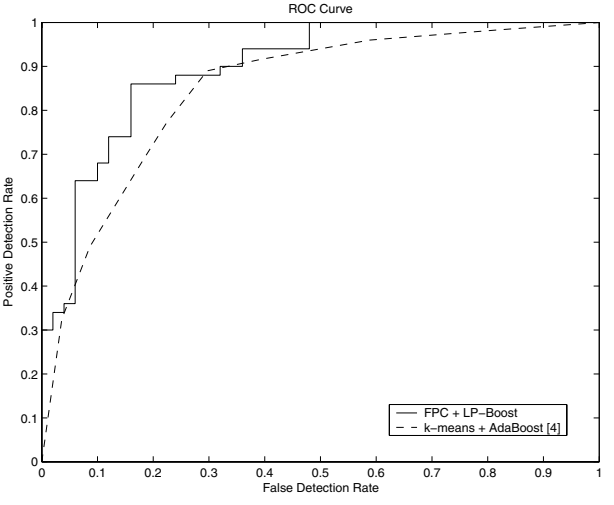

Fig. 2. ROC-Curve of our method compared to Opelt *et al.* [4]

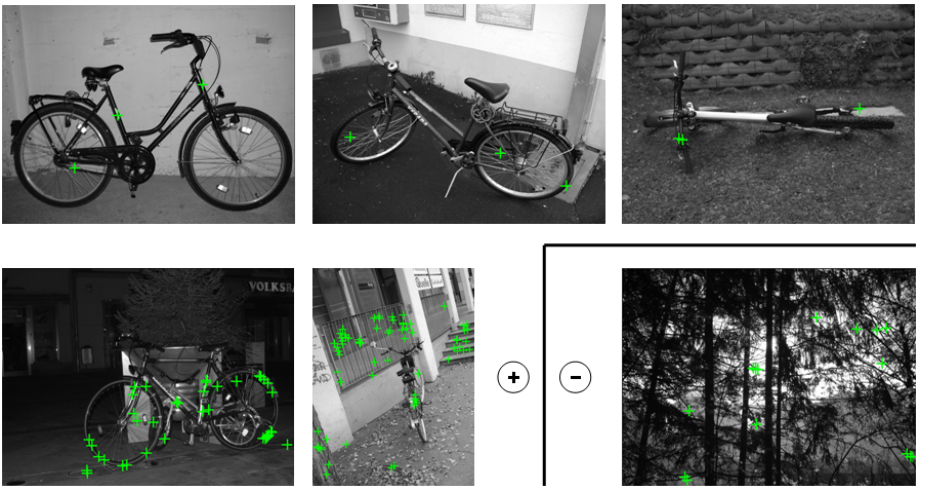

Fig. 3. Detected Features. Upper row: The features with minimum distance to the three weak hypotheses, which have the highest weight (within the ensemble) among all those that triggered for the particular image. Classification is robust to the objects pose. Lower row: **All** features with a distance below the threshold to one of the three weak hypotheses, which have the highest weight (within the ensemble) among all those that triggered for the particular image.

low subspace dimensionality observed. Similar to Opelt *et al.*, we calculate the distance matrix $\mathbf{D}^{k\times\#images}$ of clusters to images before boosting, where each entry denotes the minimum distance between a cluster's center $\mathbf{c}_j$ and the SIFTs from an image. Unlike Opelt *et al.*, we do not consider all SIFTs from an image during the calculation of the distance matrix, but only those that fall into the particular cluster's subspace bounds, setting the distance to infinity if there are no such SIFTs.

Using $\mathbf{D}$, LPBoost calls a weak learner within each boosting round to obtain a optimal weak hypothesis with respect to the current boosting weights. Particularly, a weak hypotheses is derived from a cluster by sorting the distances $\mathbf{D}_{j,\cdot}$ to its center and selecting an optimal threshold θ_j thereupon, such that the sum of the weighted labels from those images with distance below θ_j is maximized. See Opelt *et al.* [4] for details. Using LPBoost [14] we achieve an 86% ROC-equal error rate (with an area of 0.8968 below the curve) [Fig. 2] on the test set, which outperforms Opelt *et al.* [4]. Furthermore, only 28 weak hypotheses, having weights greater zero in the ensemble, contribute to the final hypothesis. Examination of the contributing weak hypotheses showed that our feature extraction focuses on typical structures like bars, tyres, spokes and wires. See figure 3 for some detected features. Particularly, the examples show that classification is robust to variations of the objects pose. Though similar structures may also be detected in the background of images from the positive class or the negative class respectively, the final classification remains correct.

4 Discussion

We have presented a new method for feature extraction from high dimensional image representations. The evaluations approve the viability of our work. Furthermore, generating weak hypotheses from out **FPC** is straightforward since each cluster denotes thresholds in various subspaces of the data. The final ensemble is sparse and outperforms results from earlier work. Hence, overfitting is limited and generalisation performance is improved. Therefore, feature extraction using FPC should be tried out on other applications involving high dimensional data to check if the results translate.

Acknowledgments

This work was supported by the European project LAVA (IST-2001-34405), by the FSP/JRP Cognitive Vision of the Austrian Science Funds (FWF-JRP S9104-N04 SP4) and in part by the IST program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

1. Lowe, D.: Object recognition from local scale-invariant features. In: Seventh International Conference on Computer Vision. (1999) 1150–1157
2. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada. (2001) 525–531

3. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: European Conference on Computer Vision. (2002) 128–141
4. Opelt, A., Fussenegger, M., Pinz, A., Auer, P.: Weak hypotheses and boosting for generic object detection and recognition. In: ECCV (2). (2004) 71–84
5. Dance, C., Willamowski, J., Csurka, G., Bray, C.: Categorizing nine visual classes with bags of keypoints. (2004)
6. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning. (1996) 148–156
7. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55** (1997) 119–139
8. Hinneburg, A., Aggarwal, C.C., Keim, D.A.: What is the nearest neighbor in high dimensional spaces? In: The VLDB Journal. (2000) 506–515
9. Aggarwal, C.C., Yu, S.P.: Finding generalized projected clusters in high dimensional spaces. In: SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM Press (2000) 70–81
10. Böhm, C., Kailing, K., Kriegel, H.P., Kröger, P.: Density connected clustering with local subspace preferences. In: Proceedings of the 4th IEEE International Conference on Data Mining. (2004) 27–34
11. Haussler, D.: Decision theoretic generalizations of the pac model for neural net and other learning applications. Inf. Comput. **100** (1992) 78–150
12. Kearns, M.J., Schapire, R.E., Sellie, L.: Toward efficient agnostic learning. In: Computational Learing Theory. (1992) 341–352
13. Long, P.M.: The complexity of learning according to two models of a drifting environment. Machine Learning **37** (1999) 337–354
14. Demiriz, A., Bennett, K.P., Shawe-Taylor, J.: Linear programming boosting via column generation. Machine Learning **46** (2002) 225–254

A The Algorithm

Algorithm 1 - Fast Projective Clustering (FPC)

procedure fpc($\mathbf{C}$, s_{min})
% Input: data set $\mathbf{C} \in \mathbb{R}^{m \times n}$, minimum support s_{min}
% Output: clustering $\mathcal{C}$

$\rho = \max \mathbf{C} - \min \mathbf{C}$
while $\mathbf{C}$ is not empty **do**
  repeat
    $[\mathbf{C}, \bar{\mathbf{C}}]$ = addBounds($\mathbf{C}$, s_{min}, ρ)
  until $\mathbf{C}$ is not refined
  $\mathcal{C} = \mathcal{C} \cup \{\mathbf{C}\}$
  $\mathbf{C} = \bar{\mathbf{C}}$
end while

---

procedure selectDensestInterval($\mathbf{C}, \varphi_i, \varphi_j, l$)
% Input: cluster $\mathbf{C} \in \mathbb{R}^{m \times n}$, bounds (φ_i, φ_j), coordinate l
% Output: data $\mathbf{C}$ from densest interval with bounds (a, b), separated data $\bar{\mathbf{C}}$

$r = \max \mathbf{C}_l - \min \mathbf{C}_l$
$\Phi_i \times \Phi_j = \{(\min \mathbf{C}_l, \varphi_i), (\varphi_i, \varphi_j), (\varphi_j, \max \mathbf{C}_l)\}$
$(a, b) = \underset{(a',b') \in \Phi_i \times \Phi_j}{\arg\max} F(a', b', r, m)$
$\mathbf{C}, \bar{\mathbf{C}} \overset{(a,b)}{\leftarrow} \mathbf{C}$

---

procedure addBounds($\mathbf{C}, s_{min}, \rho$)
% Input: initial cluster $\mathbf{C} \in \mathbb{R}^{m \times n}$, minimum support s_{min}, diameter ρ of the data set
% Output: refined cluster $\mathbf{C}$ with new bounds (a, b) along coordinate l, separated data $\bar{\mathbf{C}}$

$\Lambda = \{\}$
for $l = 1$ to n **do**
  $\mathbf{C}' = \mathbf{C}$
  $valid = true$
  while $valid$ **do**
    $[\varphi_i, \varphi_j]$ = optimalBounds($\mathbf{C}', s_{min}, l$)
    $[\mathbf{C}', \bar{\mathbf{C}}', a, b]$ = selectDensestInterval($\mathbf{C}', \varphi_i, \varphi_j, l$)
    $valid = (|\mathbf{C}'| \geq s_{min}) \wedge \left(|\bar{\mathbf{C}}'| \geq s_{min}\right)$
    if $valid$ **then**
      $\Lambda = \Lambda \cup l$
      $\mathbf{a}_l = a$
      $\mathbf{b}_l = b$
    end if
  end while
end for
if Λ is not empty **then**
  $l = \underset{l' \in \Lambda}{\arg\max}\ F(\mathbf{a}_{l'}, \mathbf{b}_{l'}, \rho_{l'}, m)$
  $\mathbf{C}, \bar{\mathbf{C}} \overset{(\mathbf{a}_l, \mathbf{b}_l)}{\leftarrow} \mathbf{C}$
end if

---

procedure optimalBounds($\mathbf{C}, s_{min}, l$)
% Input: cluster $\mathbf{C} \in \mathbb{R}^{m \times n}$, minimum support s_{min}, coordinate l
% Output: optimal bounds φ_i, φ_j maximizing the Likelihood

$\mathbf{x} = sort(\mathbf{C}_l)$
$r = \max \mathbf{x} - \min \mathbf{x}$
if r = 0 **then**
  $\varphi_i = \min \mathbf{x}$
  $\varphi_j = \max \mathbf{x}$
  return
end if
$\Phi = \{\varphi | \varphi = \frac{x_i + x_{i+1}}{2}, 1 \leq i \leq m, x_i \neq x_{i+1}\} \cup \{\min \mathbf{x}, \max \mathbf{x}\}$
$\hat{\Phi} = \{\hat{\varphi}_1 < ... < \hat{\varphi}_{|\hat{\Phi}|}\}$ **with**
  $\hat{\varphi}_1 = \min(\mathbf{x})$,
  $\hat{\varphi}_i = \min \{\varphi \mid \varphi > \hat{\varphi}_{i-1}, \#\varphi \geq \#\hat{\varphi}_{i-1} + s_{min}\}$,
  $\#\varphi = |\{x \leq \varphi\}|$
$(\hat{\varphi}_i, \hat{\varphi}_j) = \underset{\hat{\varphi}_{i'} < \hat{\varphi}_{j'}}{\arg\max}\ LL(\mathbf{x} \mid \hat{\varphi}_{i'}, \hat{\varphi}_{j'})$
$\Phi^{\hat{\varphi}_i} = \{\varphi \mid \hat{\varphi}_{i-1} \leq \varphi \leq \hat{\varphi}_{i+1}\}$
$\Phi^{\hat{\varphi}_j} = \{\varphi \mid \hat{\varphi}_{j-1} \leq \varphi \leq \hat{\varphi}_{j+1}\}$
$(\varphi_i, \varphi_j) = \underset{(\varphi_{i'}, \varphi_{j'}) \in \Phi^{\hat{\varphi}_i} \times \Phi^{\hat{\varphi}_j}}{\arg\max}\ LL(\mathbf{x} \mid \varphi_{i'}, \varphi_{j'})$
