11 Appendix

11.1 Qualitative Analysis of Regularization in BM-MDL Algorithm

While the original MDL definition aims at finding an optimal grouping of bipartite data that balances the description lengths of the summary and the corrections, the regularization terms we introduce in Equation 1 can be used to contract the bipartite graph data into a more compact form. In general, when bipartite data becomes sparse, we can set βs to a greater value to encourage merging the node clusters further. Fig. 9 shows the partition results of our vehicle case study dataset with different values of β. In practice, our system provides different choices of β to cope with different needs of co-clustering. In the user interface we map different values of β to different levels of clustering strength. For ease of user interaction and also for illustration here we use the same value of β for both β_P and β_Q.

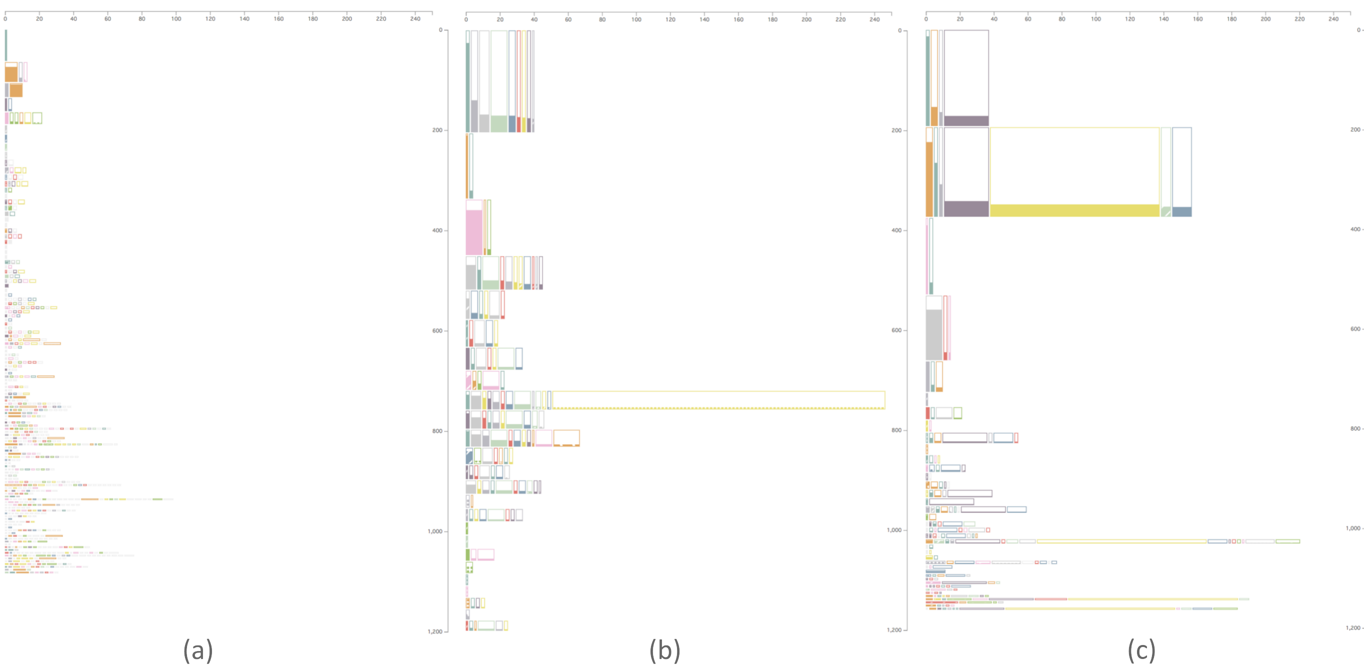

Fig. 9. Summarization results of our vehicle dataset consisting of 2967 vehicle logs with around 250 fault codes using different values of β. (a)$\beta = 1$; (b)$\beta = 10$; (c)$\beta = 100$; Only co-clusters with more than 10% density are included.

11.2 Subroutines in Algorithm 1 and Algorithm 2

Algorithm 3: $cost\_reduction\_for\_bundling((p,q),(p',q),R,\mathbb{S})$

Input: $R \subseteq U \times V$, $\mathbb{S} \subseteq P \times Q$, $p, p' \in P$ and $q \in Q$
Output: Cost reduction Δ for bundling (p,q), (p',q)
$\Delta = 0$
`/* Calculate description length for` (p,q) `*/`
if $(p,q) \in \mathbb{S}$ **then**
  $\Delta + = 1$
  $\Delta + = \alpha \cdot (\|p \times q\| - \|(p \times q) \cap R\|)$ `/* Corrections: remove non-existing edges */`
else
  $\Delta + = \alpha \cdot \|(p \times q) \cap R\|$ `/* Corrections: add existing edges */`
end
`/* Calculate description length for` (p',q) `*/`
if $(p',q) \in \mathbb{S}$ **then**
  $\Delta + = 1$
  $\Delta + = \alpha \cdot (\|p' \times q\| - \|(p' \times q) \cap R\|)$
else
  $\Delta + = \alpha \cdot \|(p' \times q) \cap R\|$
end
`/* Calculate max description length reduction by bundling` (p,q) `and` (p,q') `*/`
$\Delta - = min(1 + \alpha \cdot (\|(p \cup p') \times q\| - \|((p \cup p') \times q) \cap R\|), \alpha \cdot \|((p \cup p') \times q) \cap R\|)$
`/* Output results */`
return Δ

Algorithm 4: $merge(p, p', R, \mathbb{S})$

Input: $R \subseteq U \times V$, $\mathbb{S} \subseteq P \times Q$ and $p, p' \in P$
Output: Updated P, Q, $\mathbb{S} \subseteq P \times Q$
remove p, p' from P, $\mathbb{S}$
remove all edges adjacent to p or p' from $\mathbb{S}$
add $p_{new} = p \cup p'$ to P, $\mathbb{S}$
for $q \in Q$ **do**
  if $1 + \alpha \cdot (\|p_{new} \times q\| - \|(p_{new} \times q) \cap R\|) < \alpha \cdot \|(p_{new} \times q) \cap R\|)\|$ **then**
    add edge (p_{new}, q) to $\mathbb{S}$
  end
end