CRAFT: ClusteR-specific Assorted Feature selecTion

Vikas K. Garg
CSAIL, MIT

Cynthia Rudin
CSAIL and Sloan, MIT

Tommi Jaakkola
CSAIL, MIT

Abstract

We present a hierarchical Bayesian framework for clustering with cluster-specific feature selection. We derive a simplified model, CRAFT, by analyzing the asymptotic behavior of the log posterior formulations in a nonparametric MAP-based clustering setting in this framework. CRAFT handles assorted data, i.e., both numeric and categorical data, and the underlying objective functions are intuitively appealing. The resulting algorithm is simple to implement and scales nicely, requires minimal parameter tuning, obviates the need to specify the number of clusters a priori, and compares favorably with other state-of-the-art methods on several datasets. We provide empirical evidence on carefully designed synthetic data sets to highlight the robustness of the algorithm to recover the underlying feature subspaces, even when the average dimensionality of the features across clusters is misspecified. Besides, the framework seamlessly allows for multiple views of clustering by interpolating between the two extremes of cluster-specific feature selection and global selection, and recovers the DP-means objective [14] under the degenerate setting of clustering without feature selection.

1 Introduction

Feature or variable selection remains a key aspect of modern high-dimensional regression, classification, and structured prediction tasks. Beyond statistical gains from overt dimensionality reduction, isolating a small number of relevant features cuts down test-time computation and storage, provides easily interpretable models, and facilitates data visualization (e.g., [9, 16]). The role of feature selection in clustering is, however, more nuanced. Specifying a reasonable clustering metric in high dimensions is challenging. Indeed, dimensionality reduction methods such as PCA, Laplacian eigenmaps [4], or random projections [27] are often used prior to K-means or other clustering algorithms. Feature selection, as a dimensionality reduction method, entertains strictly axis aligned projections. The main argument for this restriction over oblique projections is interpretability as the original coordinates tend to be well-grounded on the application. The restriction does not necessarily imply a computational overhead. For example, typical similarity measures decompose coordinate-wise and the relevant subset could be obtained via ℓ_1 regularization [28]. Some methods, instead, use a pairwise fusion [8] or a group lasso penalty [25].

By changing the clustering metric via feature selection (or dimensionality reduction), we may also alter what the resulting clusters are. For example, we can cluster apples and grapes based on their size, color, or other features, obtaining different clusterings as a result. While the ambiguity is inherent and present at the outset (e.g., due to different scaling of coordinates), the issue highlights the need to properly setup the overall clustering objective.

Generative models provide a natural basis for specifying clustering objectives, especially with feature selection. For example, we can define a coherent objective for *global* feature selection by adaptively assigning a broad distribution over irrelevant features (effectively excluding them) while concentrating around on others in order for them to influence clustering decisions. Moreover, in many real applications, it makes sense to perform *local* or *cluster-specific* feature selection where clusters can adjust their own metric. For instance, when clustering news articles, similarity between political articles should be assessed based on the language (features) about politics, discounting references to other topics. Making the selection cluster specific does not introduce any major conceptual challenges; indeed, even a simple mixture of multi-variate Gaussians already involves cluster-specific metrics. We open up the selection of which features are modeled with broad or specific distributions at the cluster level, and balance the tendency for clusters to agree on the relevant features (global selection) as opposed to selecting them anew (local selection).

In this paper, we specify a generative hierarchical Bayesian

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 41. Copyright 2016 by the authors.

model for clustering with cluster-specific feature selection. While unsupervised *global* feature selection is widely considered hard [15], *cluster-specific* unsupervised feature selection is somewhat harder (computationally) since separate, possibly overlapping, subsets of features need to be inferred along with the clusters. To address this computational challenge, we study our hierarchical model under asymptotic conditions. The resulting simplified model, CRAFT, dispenses with the need to model unselected features, leaving only a regularizer for the selected features.

CRAFT retains essential benefits of the full Bayesian model, while additionally lending simplicity and scalability to the model. For example, CRAFT provides multiple views to clustering - it contains a single prior parameter that can be adjusted for a desired balance between global and local feature selection. Moreover, CRAFT can handle both numeric and categorical features (assorted data). Many real datasets contain categorical variables or are processed to contain categorical variables; for instance, in web-based clustering applications, it is standard to represent each webpage as a binary (categorical) feature vector. A vast majority of clustering methods, like K-means [17, 18], that were designed for numeric data, do not work well on categorical variables due to absence of ordinal relationships among the categorical labels. This explains why despite several attempts (see, e.g., [1, 2, 11, 12, 21]), variations of K-means have largely proved ineffective in handling mixed data.

The derivation of the CRAFT algorithm follows from asymptotics on the log posterior of its generative model. The model is based on a Dirichlet process mixture [7, 22, 23],[1] and thus the number of clusters can be chosen nonparametrically by the algorithm. Our asymptotic calculations build on the recent advancements in approximate learning due to the works of Kulis and Jordan [14], who derived the DP-means objective by considering approximations to the log-likelihood, and Broderick et al. [5], who instead approximated the posterior log likelihood to derive other nonparametric variations of K-means. These works do not consider feature selection, and as a result, our generative model is entirely different, and the calculations differ considerably from previous works. When the data are only numeric, we recover the DP-means objective with an additional term arising due to feature selection. CRAFT's asymptotics yield interpretable objective functions, and suggest K-means style algorithms that recovered subspaces on synthetic data, and outperformed several state-of-the-art benchmarks on many real datasets.

We introduce our framework in Section 2, and discuss some degenerate cases and some possible extensions in Section 3. We provide a detailed experimental analysis in Section 4, and conclude in Section 5. All the derivations are given in the Supplementary Section 6 for improved readability.

2 The Proposed Model

The main intuition behind our formalism is that the points in a cluster should agree closely on the features selected for that cluster. As it turns out, the objective is closely related to the cluster's entropy for discrete data and variance for numeric data. For instance, consider a parametric setting where the features are all binary categorical, taking values only in $\{0, 1\}$, and we select all the features. Assume that the features are drawn from independent Bernoulli distributions. Let the cluster assignment vector be z, i.e., $z_{n,k} = 1$ if point x_n is assigned to cluster k. Then, we obtain the following objective using a straightforward maximum likelihood estimation (MLE) procedure:

$$\underset{z}{\operatorname{argmin}} \sum_{k} \sum_{n: z_{n,k}=1} \sum_{d} \mathbb{H}(\mu^*_{kd})$$

where μ^*_{kd} denotes the mean of feature d computed by using points belonging to cluster k, and the entropy function $\mathbb{H}(p) = -p \log p - (1-p) \log(1-p)$ for $p \in [0, 1]$ characterizes the uncertainty. Thus the objective tries to minimize the overall uncertainty across clusters and thus forces similar points to be close together, which makes sense from a clustering perspective.

It is not immediately clear how to extend this insight about clustering to cluster-specific feature selection. Our model combines assorted data by enforcing a common Bernoulli prior that selects features, regardless of whether they are categorical or numerical. We derive an asymptotic approximation for the joint log likelihood of the observed data, cluster indicators, cluster means, and feature means. Modeling assumptions are then made for categorical and numerical data separately; this is why our model can handle multiple data types. Unlike the more sophisticated Variational Bayes procedures, such as [7], the CRAFT asymptotics lead to an elegant K-means style algorithm that has the following simple steps repeated in each iteration: (a) compute the "distances" to the cluster centers using the selected features for each cluster, (b) choose which cluster each point should be assigned to (and create new clusters if needed), and (c) recompute the centers and select the appropriate cluster-specific features for the next iteration.

Formally, the data x consists of N i.i.d. D-dimensional binary vectors $x_1, x_2, \ldots, x_N$. We assume a Dirichlet process (DP) mixture model to avoid having to specify a priori the number of clusters K^+, and use the hyper-parameter θ, in the underlying exchangeable probability partition function (EFPF) [20], to tune the probability of starting a new cluster. We use z to denote cluster indicators: $z_{n,k} = 1$ if x_n is assigned to cluster k. Since K^+ depends on z, we will often make the connection explicit by writing $K^+(z)$. Let Cat and Num denote respectively the set of categorical and the set of numeric features respectively.

We also introduce the variables $v_{kd} \in \{0, 1\}$ to indicate whether feature $d \in [D]$ is selected in cluster $k \in [K]$. We assume v_{kd} is generated from a Bernoulli distribution with

[1] See [13] for a prototype model with feature selection.

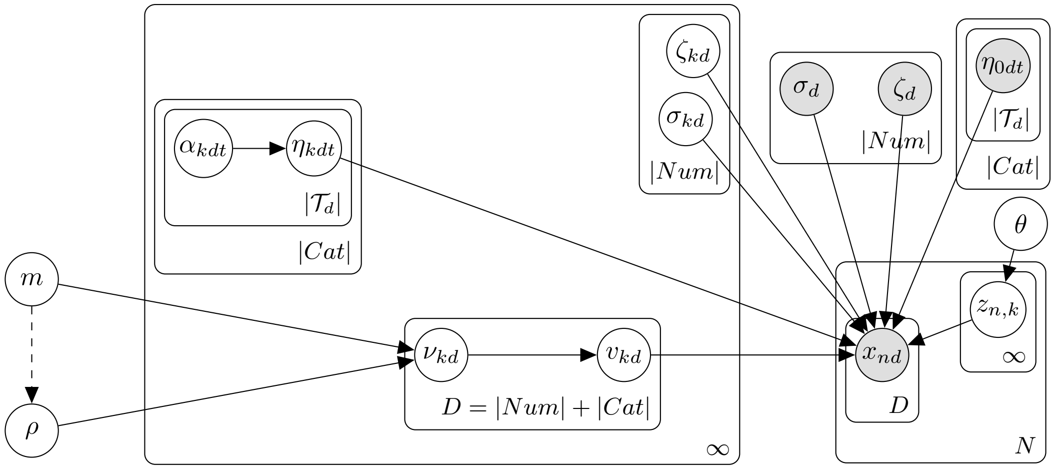


Figure 1: CRAFT- Graphical model. For cluster-specific feature selection ρ is set to a high value determined by m, whereas for global feature selection ρ is set close to 0. The dashed arrow emphasizes this unification of cluster-specific and global feature selection using a single parameter ρ.

parameter ν_{kd}. Further, we assume ν_{kd} is generated from a Beta prior having variance ρ and mean m.

For categorical features, the features d selected in a cluster k have values drawn from a discrete distribution with parameters η_{kdt}, $d \in Cat$, where $t \in \mathcal{T}_d$ indexes the different values taken by the categorical feature d. The parameters η_{kdt} are drawn from a Beta distribution with parameters α_{kdt}/K^+ and 1. On the other hand, we assume the values for the features not selected to be drawn from a discrete distribution with cluster-independent mean parameters η_{0dt}.

For numeric features, we formalize the intuition that the features selected to represent clusters should exhibit small variance relative to unselected features by assuming a conditional density of the form:

$$
\begin{aligned}
f(x_{nd}|v_{kd}) &= \frac{e^{-\left[v_{kd}\frac{(x_{nd}-\zeta_{kd})^2}{2\sigma_{kd}^2}+(1-v_{kd})\frac{(x_{nd}-\zeta_d)^2}{2\sigma_d^2}\right]}}{Z_{kd}}, \\
Z_{kd} &= \frac{\sqrt{2\pi}\sigma_d\sigma_{kd}}{\sigma_{kd}\sqrt{1-v_{kd}}+\sigma_d\sqrt{v_{kd}}},
\end{aligned}
$$

where $x_{nd} \in \mathbb{R}$, $v_{kd} \in \{0,1\}$, and Z_{kd} ensures f integrates to 1, and σ_{kd} guides the allowed variance of a selected feature d over points in cluster k by asserting feature d concentrate around its cluster mean ζ_{kd}. The features not selected are assumed to be drawn from Gaussian distributions that have cluster independent means ζ_d and variances σ_d^2. Fig. 1 shows the corresponding graphical model.

Let $\mathbb{I}(\mathcal{P})$ be 1 if the predicate $\mathcal{P}$ is true, and 0 otherwise. Under asymptotic conditions, minimizing the joint negative log-likelihood yields the following objective

$$
\begin{aligned}
\arg\min_{z,v,\eta,\zeta,\sigma} \underbrace{\sum_{k=1}^{K^+}\sum_{n:z_{n,k}=1}\sum_{d\in Num}\frac{v_{kd}(x_{nd}-\zeta_{kd})^2}{2\sigma_{kd}^2}}_{\text{Numeric Data Discrepancy}}
&+ \underbrace{(\lambda+DF_0)K^+}_{\text{Regularization Term}} + \underbrace{\left(\sum_{k=1}^{K^+}\sum_{d=1}^{D} v_{kd}\right)F_\Delta}_{\text{Feature Control}} \\
&+ \underbrace{\sum_{k=1}^{K^+}\sum_{d\in Cat}\Bigg[v_{kd}\left(\sum_{n:z_{n,k}=1} -\mathbb{I}(x_{nd}=t)\log\eta_{kdt}\right)}_{\text{Categorical Discrepancy Term I}} \\
&\underbrace{+ (1-v_{kd})\sum_{n:z_{n,k}=1}\sum_{t\in\mathcal{T}_d} -\mathbb{I}(x_{nd}=t)\log\eta_{0dt}\Bigg]}_{\text{Categorical Discrepancy Term II}},
\end{aligned}
$$

Algorithm 1 CRAFT

```
Input: $x_1, \ldots, x_N$: $D$-dimensional input data with categorical features $Cat$ and numeric features $Num$, $\lambda > 0$: regularization parameter, and $m \in (0,1)$: fraction of features per cluster, and (optional) $\rho \in (0, m(1-m))$: control parameter that guides global/local feature selection. Each feature $d \in Cat$ takes values from the set $\mathcal{T}_d$, while each feature $d \in Num$ takes values from $\mathbb{R}$.
Output: $K^+$: number of clusters, $l_1, \ldots, l_{K^+}$: clustering, and $v_1, \ldots, v_{K^+}$: selected features.

  1. Initialize $K^+ = 1$, $l_1 = \{x_1, \ldots, x_N\}$, cluster center (sample randomly) with categorical mean $\eta_1$ and numeric mean $\zeta_1$, and draw $v_1 \sim [Bernoulli\,(m)]^D$. If $\rho$ is not specified as an input, initialize $\rho = \max\{0.01, m(1-m) - 0.01\}$. Compute the global categorical mean $\eta_0$. Initialize the cluster indicators $z_n = 1$ for all $n \in [N]$, and $\sigma_{1d} = 1$ for all $d \in Num$.
  2. Compute $F_\Delta$ and $F_0$ using (1).
  3. Execute the CAF routine shown in Algorithm 2.
```

where $F_\triangle$ and F_0 depend only on the (m, ρ) pair: $F_\triangle = F_1 - F_0$, with

$$\begin{aligned} F_0 &= (a_0 + b_0)\log(a_0 + b_0) - a_0 \log a_0 - b_0 \log b_0, \\ F_1 &= (a_1 + b_1)\log(a_1 + b_1) - a_1 \log a_1 - b_1 \log b_1, \\ a_0 &= \frac{m^2(1-m)}{\rho} - m, b_0 = \frac{m(1-m)^2}{\rho} + m, \\ a_1 &= a_0 + 1, \text{ and } b_1 = b_0 - 1. \end{aligned} \tag{1}$$

This objective has an elegant interpretation. The categorical and numerical discrepancy terms show how selected features (with $v_{kd} = 1$) are treated differently than unselected features. The regularization term controls the number of clusters, and modulates the effect of feature selection. The feature control term contains the adjustable parameters: m controls the number of features that would be turned on per cluster, whereas ρ guides the extent of cluster-specific feature selection. A detailed derivation of the objective is provided in the Supplementary.

A K-means style alternating minimization procedure for clustering assorted data as well as selecting features is outlined in Algorithm 1. The algorithm repeats the following steps until convergence: (a) compute the “distances” to the cluster centers using the selected features for each cluster, (b) choose which cluster each point should be assigned to (and create new clusters if needed), and (c) recompute the cluster centers and select the appropriate features for each cluster using the criteria that follow directly from the model objective and variance asymptotics. In particular, the algorithm starts a new cluster if the cost of assigning a point to its closest cluster center exceeds $(\lambda + DF_0)$, the cost it would incur to initialize an additional cluster. The information available from the already selected features is leveraged to guide the initial selection of features in the new cluster. Finally, the updates on cluster means and feature selection are performed at the end of each iteration.

Approximate Budget Setting for a Variable Number of Features: Algorithm 1 selects a fraction m of features per cluster, uniformly across clusters. A slight modification would allow Algorithm 1 to have a variable number of features across clusters, as follows: specify a tuning parameter $\epsilon_c \in (0, 1)$ and choose all the features d in cluster k for which $G_d - G_{kd} > \epsilon_c G_d$. Likewise for numeric features, we may choose features that have variance less than some positive constant ϵ_v. As we show later, this slightly modified algorithm recovers the exact subspace on synthetic data in the approximate budget setting for a wide range of m.

3 Discussion

We now discuss some conceptual underpinnings underlying CRAFT, and also describe briefly some special cases and straightforward extensions to the model.

Algorithm 2 (Auxiliary Module) Cluster Assignment and Feature Selection (CAF)

```
Repeat until cluster assignments do not change
  • For each point $x_n$
      – Compute $\forall k \in [K^+]$,
          $d_{nk} = \sum_{d \in Cat: v_{kd}=0} \sum_{t \in \mathcal{T}_d} -\mathbb{I}(x_{nd} = t) \log \eta_{0dt}$
          $+ \sum_{d \in Cat: v_{kd}=1} \sum_{t \in \mathcal{T}_d} -\mathbb{I}(x_{nd} = t) \log \eta_{kdt}$
          $+ \sum_{d \in Num} v_{kd} \frac{(x_{nd} - \zeta_{kd})^2}{2\sigma_{kd}^2} + \left(\sum_{d=1}^{D} v_{kd}\right) F_\triangle.$
      – If $\min_k d_{nk} > (\lambda + DF_0)$, set $K^+ = K^+ + 1$,
        $z_n = K^+$, and draw $\forall\, d \in [D]$,
          $v_{K^+d} \sim Bernoulli\left(\frac{\sum_{j=1}^{K^+-1} a_{v_{jd}}}{\sum_{j=1}^{K^+-1}(a_{v_{jd}} + b_{v_{jd}})}\right)$,
        where $a$ and $b$ are as defined in (1). Set $\eta_{K^+}$
        and $\zeta_{K^+}$ using $x_n$. Set $\sigma_{K^+d} = 1$ for all $d \in Num$.
      – Otherwise, set $z_n = \operatorname{argmin}_k d_{nk}$.
  • Generate clusters $l_k = \{x_n \mid z_n = k\}$, $\forall k \in \{1, 2, \ldots, K^+\}$, using $z_1, \ldots, z_{K^+}$.
  • Update the means $\eta$ and $\zeta$, and variances $\sigma^2$, for all clusters.
  • For each cluster $l_k$, $k \in [K^+]$, update $v_k$: choose
    the $m|Num|$ numeric features $d'$ with lowest $\sigma_{kd'}$
    in $l_k$, and choose $m|Cat|$ categorical features $d$
    with maximum value of $G_d - G_{kd}$, where $G_d = -\sum_{n:z_{n,k}=1} \sum_{t \in \mathcal{T}_d} \mathbb{I}(x_{nd} = t) \log \eta_{0dt}$ and
    $G_{kd} = -\sum_{n:z_{n,k}=1} \sum_{t \in \mathcal{T}_d} \mathbb{I}(x_{nd} = t) \log \eta_{kdt}$.
```

Recovering DP-means objective on Numeric Data

CRAFT recovers the DP-means objective [14] in a degenerate setting (see Supplementary):

$$\operatorname*{argmin}_z \sum_{k=1}^{K^+(z)} \sum_{n:z_{n,k}=1} \sum_d (x_{nd} - \zeta_{kd}^*)^2 + \lambda K^+(z), \tag{2}$$

where ζ_{kd}^* denotes the (numeric) mean of feature d computed by using points belonging to cluster k.

Unifying Global and Local Feature Selection

The point estimate of ν_{kd} is

$$\frac{a_{kd}}{a_{kd}+b_{kd}} = \frac{\left(\dfrac{m^2(1-m)}{\rho} - m\right) + v_{kd}}{\dfrac{m(1-m)}{\rho}}$$

$$= m + \frac{(v_{kd}-m)\rho}{m(1-m)} \rightarrow \begin{cases} v_{kd}, \text{ as } \rho \rightarrow m(1-m) \\ m, \text{ as } \rho \rightarrow 0. \end{cases}$$

Thus, using a single parameter ρ, we can interpolate between cluster specific selection, $\rho \rightarrow m(1-m)$, and global selection, $\rho \rightarrow 0$. Since we are often interested only in one of these two extreme cases, this also implies that we essentially need to specify only m, which is often governed by the corresponding application requirements. Thus, CRAFT requires minimal tuning for most practical purposes.

Accommodating Statistical-Computational Trade-offs

We can extend the basic CRAFT model of Fig. 1 to have cluster specific means m_k, which may in turn be modulated via Beta priors. The model can also be readily extended to incorporate more informative priors or allow overlapping clusters, e.g., we can do away with the independent distribution assumptions for numeric data, by introducing covariances and taking a suitable prior like the inverse Wishart. The parameters α and σ_d do not appear in the CRAFT objective since they vanish due to the asymptotics and the appropriate setting of the hyperparameter θ. Retaining some of these parameters, in the absence of asymptotics, will lead to additional terms in the objective thereby requiring more computational effort. Depending on the available computational resource, one might also like to achieve feature selection with the exact posterior instead of a point estimate. CRAFT's basic framework can gracefully accommodate all such statistical-computational trade-offs.

4 Experimental Results

We first provide empirical evidence on synthetic data about CRAFT's ability to recover the feature subspaces. We then show how CRAFT outperforms an enhanced version of DP-means that includes feature selection on a real binary dataset. This experiment underscores the significance of having different measures for categorical data and numeric data. Finally, we compare CRAFT with other recently proposed feature selection methods on real world benchmarks. In what follows, the *fixed budget* setting is where the number of features selected per cluster is held constant, and the *approximate budget* setting is where the number of features selected per cluster is allowed to vary across the clusters. We set $\rho = m(1-m) - 0.01$ in all our experiments to facilitate cluster specific feature selection. In our experiments, CRAFT continued to perform very well relative to the other algorithms when the number of clusters was increased. Therefore, we made the subjective choice to include results only up to 10 clusters since similar behavior was observed for higher number of clusters.

Exact Subspace Recovery on Synthetic Data

We now show the results of our experiments on synthetic data, in both the fixed and the approximate budget settings, that suggest CRAFT has the ability to recover subspaces on both categorical and numeric data with minimum parameter tuning, amidst noise, under different scenarios: (a) disjoint subspaces, (b) overlapping subspaces including the extreme case of containment of a subspace wholly within the other, (c) extraneous features, and (d) non-uniform distribution of examples and features across clusters.

Fixed Budget Setting: Fig. 2(a) shows a binary dataset comprising 300 24-feature points, evenly split between 3 clusters that have disjoint subspaces of 8 features each. We sampled the remaining features independently from a Bernoulli distribution with parameter 0.1. Fig. 2(b) shows that CRAFT recovered the subspaces with $m = 1/3$, as we would expect. In Fig. 2(c) we modified the dataset to have (a) an unequal number of examples across the different clusters, (b) a fragmented feature space each for clusters 1 and 3, (c) a completely noisy feature, and (d) an overlap between second and third clusters. As shown in Fig. 2(d), CRAFT again identified the subspaces accurately.

Fig. 3(a) shows the second dataset comprising 300 36-feature points, evenly split across 3 clusters, drawn from independent Gaussians having unit variance and means 1, 5 and 10 respectively. We designed clusters to comprise features 1-12, 13-24, and 22-34 respectively so that the first two clusters were disjoint, whereas the last two had some overlapping features. We added isotropic noise by sampling the remaining features from a Gaussian distribution having mean 0 and standard deviation 3. Fig. 3(b) shows that CRAFT recovered the subspaces with $m = 1/3$. We then modified this dataset in Fig. 3(c) to have cluster 2 span a non-contiguous feature subspace. Additionally, cluster 2 was designed to have one partition of its features overlap partially with cluster 1, while the other was subsumed completely within the subspace of cluster 3. Several extraneous features were not contained within any cluster. CRAFT recovered the subspaces on these data too (Fig. 3(d)).

Approximate Budget Setting: We now show CRAFT may recover the subspaces even when we allow a different number of features to be selected across the different clusters. We modified the original categorical synthetic dataset to have cluster 3 (a) overlap with cluster 1, and more importantly, (b) significantly overlap with cluster 2. We obtained the configuration, shown in Fig. 4(a), by splitting cluster 3 (8 features) evenly in two parts, and increasing the number of features in cluster 2 (16 features) considerably relative to cluster 1 (9 features), thereby making the distribution of features across the clusters non-uniform. We observed (Fig. 4(b)) that for $\epsilon_c \in [0.76, 1)$, the CRAFT algorithm for the approximate budget setting recovered the subspace exactly for a wide range of m, more specifically for all values, when m was varied in increments of 0.1 from 0.2 to 0.9. This implies the procedure essentially requires tuning only ϵ_c. We easily found the appropriate range by searching in decrements of 0.01 starting from 1. Fig. 4(d) shows the recovered subspaces for a similar set-up for the numeric data shown in Fig. 4(c). We observed that for $\epsilon_v \in [4, 6]$, the recovery was robust to selection of $m \in [0.1, 0.9]$, similar to the case of categorical data. Specifically, we searched for ϵ_v in increments of 0.5 from 1 to 9, since empirically the global variance was found to be close to 9. Thus, with minimal tuning, we recovered subspaces in all cases.

Experimental Setup for Real Datasets

In order to compare the non-parametric CRAFT algorithm with other methods (where the number of clusters K is not defined in advance), we followed the farthest-first heuristic used by the authors of DP-means [14], which is reminiscent of the seeding proposed in methods such as K-means++ [3] and Hochbaum-Shmoys initialization [10]: for an approximate number of desired clusters k, a suitable λ is found in the following manner. First a singleton set T is initialized, and then iteratively at each of the k rounds, the point in the dataset that is farthest from T is added to T. The distance of a point x from T is taken to be the smallest distance between x and any point in T, for evaluating the corresponding objective function. At the end of the k rounds, we set λ as the distance of the last point that was included in T. Thus, for both DP-means and CRAFT, we determined their respective λ by following the farthest first heuristic evaluated on their objectives: K-means objective for DP-means and entropy based objective for CRAFT.

Kulis and Jordan [14] initialized T to the global mean for DP-means. We instead chose a point randomly from the input to initialize T for CRAFT. In our experiments, we often found this strategy to be more effective than using the global mean since the cluster centers tend to be better separated. To highlight the failure of the squared Euclidean distance based objective to deal with categorical data sets, we also conducted experiments on DP-means with a random selection of the initial cluster center from the data points. We call this method DP-means(R) where R indicates randomness in selecting the initial center.

Evaluation Criteria For Real Datasets

To evaluate the quality of clustering, we used datasets with known true labels. We employed two standard metrics, *purity* and *normalized mutual information* (NMI), to measure the clustering performance [19, 24]. Assessing the quality of unsupervised methods by comparing with underlying labels using NMI and purity, whenever available, is a standard and well-established practice, see for instance, DP-means [14], NDFS [15], and MCFS [6]. To compute purity, each full cluster is assigned to the class label that is most frequent in the cluster. Purity is the proportion of examples that we assigned to the correct label. Normalized mutual information is the mutual information between the cluster labeling and the true labels, divided by the square root of the true label entropy times the clustering assignment entropy. Both purity and NMI lie between 0 and 1 – the closer they are to 1, the better the quality of the clustering.

Henceforth, we use Algorithm 1 with the fixed budget setting in our experiments to ensure a fair comparison with the other methods, since they presume a fixed m.

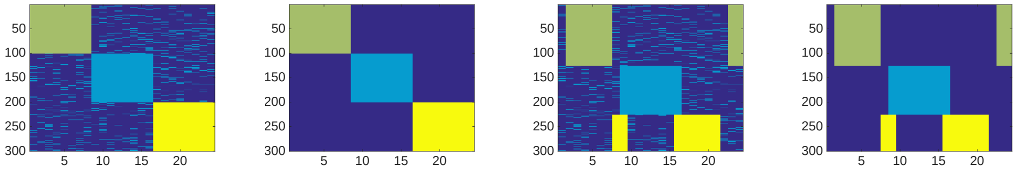
Comparison with DP-means

We now provide evidence that CRAFT outperforms DP-means on categorical data, using the Splice junction determination dataset [26] that has all categorical features. We treated feature values 1 and 2 with 0, and 3 and 4 with 1 to create a binary dataset. We borrowed the feature selection term from CRAFT to extend DP-means(R) to include feature selection, and retained its squared Euclidean distance measure. Recall that, in a special case, the CRAFT objective degenerates to DP-means(R) on numeric data when all features are retained, and cluster variances are all the same (see the Supplementary). Fig. 5 shows the comparison results on the Splice data for different values of m. CRAFT outperforms extended DP-means(R) in terms of both purity and NMI, showing the importance of the entropy term in the context of clustering with feature selection.

Comparison with Feature Selection Methods

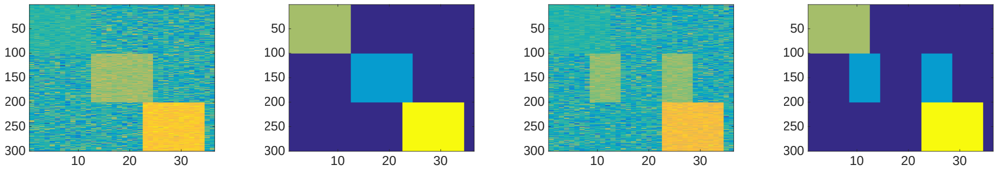
We now demonstrate the benefits of cluster specific feature selection accomplished by CRAFT. Table 1 and Table 2 show how CRAFT compares with two state-of-the-art unsupervised feature selection methods – *Multi-Cluster/Class Feature Selection* (MCFS) [6] and *Nonnegative Discriminative Feature Selection* (NDFS) [15] – besides DP-means and DP-means(R) on several datasets [26], namely Bank, Spam, Wine, Splice (described above), and Monk-3, when m was set to 0.5 and 0.8 respectively. MCFS derives its inspiration from manifold learning and L1-regularized models, and solves an optimization problem involving a sparse eigenproblem and a L1-regularized least squares problem. NDFS exploits the discriminative information in unsupervised scenarios by performing spectral clustering to learn the cluster labels of the input samples, while simultaneously performing the feature selection step.

Our experiments clearly highlight that CRAFT (a) works well for both numeric and categorical data, and (b) compares favorably with both the global feature selection algorithms and clustering methods, such as DP-means, that



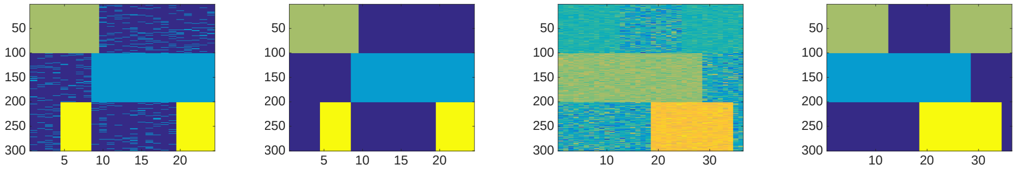
(a) Dataset (b) CRAFT (c) Dataset (d) CRAFT

Figure 2: **(Fixed budget)** CRAFT recovered the subspaces on categorical datasets.



(a) Dataset (b) CRAFT (c) Dataset (d) CRAFT

Figure 3: **(Fixed budget)** CRAFT recovered the subspaces on numeric datasets.



(a) Dataset (b) CRAFT (c) Dataset (d) CRAFT

Figure 4: **(Approximate budget)** CRAFT recovered the subspaces on both the categorical data shown in (a) and the numeric data shown in (c). The subspaces were recovered with minimal tuning even when m was incorrectly specified.

do not select features. Besides, the CRAFT algorithm exhibited low variance across runs, thereby suggesting that the method is robust to initialization. Similar results were obtained for other values of m - we omit the details here.

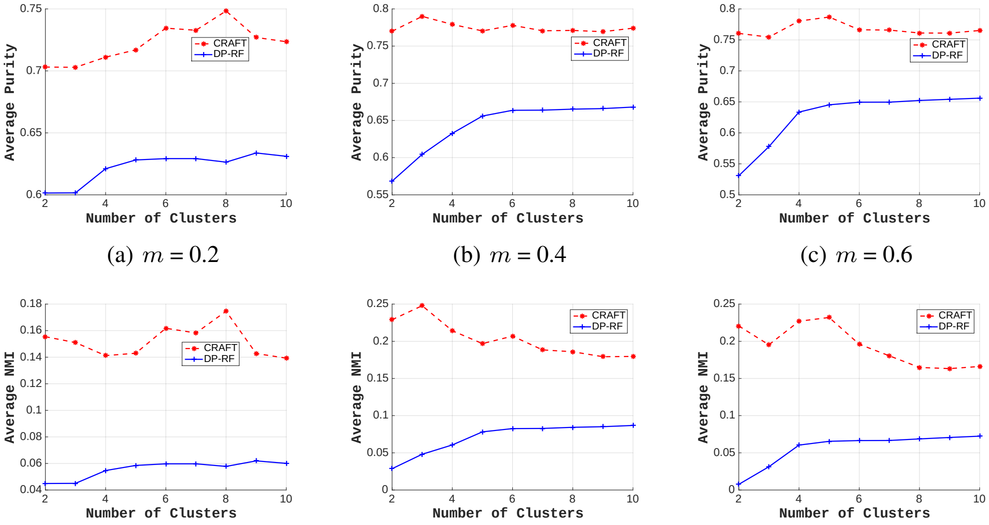
Arguably, setting threshold to control the number of clusters for nonparametric methods (CRAFT, DP-means) in order to compare them to parametric models (MCFS, NDFS) is not exactly fair, since they can better model the data by generating more clusters than the number of true clusters (e.g. when the clusters are not unimodal). Interestingly, despite being at a slight disadvantage due to this heuristic, we observe that CRAFT performs very well on several datasets. We found CRAFT to perform well in terms of the macro F1-score too. For instance, on the Wine dataset, we obtained the following macro F1-scores for $m = 0.5$ - CRAFT: 0.55, MCFS: 0.51, NDFS: 0.50, DP-means(R): 0.29, and DP-means: 0.19. Similarly, on Spam, we observed the following scores - CRAFT: 0.65, MCFS: 0.19, NDFS: 0.19, DP-means: 0.01, and DP-means(R): 0.01.

Finally, we found that besides these criteria, CRAFT also showed good performance in terms of time. For instance, on the Spam dataset for $m = 0.5$, CRAFT required an average execution time of only 0.4 seconds, compared to 1.8 and 61.4 seconds by MCFS and NDFS respectively. This behavior can be attributed primarily to the benefits of the scalable K-means style algorithm employed by CRAFT, as opposed to MCFS and NDFS that require solving computationally expensive spectral problems. This scalability aspect of CRAFT becomes even more prominent with a larger number of data points. For instance, MCFS and NDFS failed to scale to the Adult dataset (a.k.a Census Income Data), which consists of about 50,000 examples. In contrast, CRAFT converged quickly to a good solution.

5 Conclusion

CRAFT's framework incorporates cluster-specific feature selection and handles both categorical and numeric data. The objective obtained from MAP asymptotics is interpretable, and informs simple algorithms for both the fixed budget setting and the approximate budget setting.

Our work opens up several theoretical and practical directions. One direction would be to investigate the various noise conditions under which the model is able to recover



(a) $m = 0.2$ (b) $m = 0.4$ (c) $m = 0.6$

(d) $m = 0.2$ (e) $m = 0.4$ (f) $m = 0.6$

Figure 5: Purity (a-c) and NMI (d-f) results on Splice. DP-RF is DP-means(R) with feature selection.

Table 1: CRAFT versus DP-means and state-of-the-art feature selection methods when half of the features were selected (i.e. $m = 0.5$). We abbreviate MCFS to M, NDFS to N, DP-means to D, and DP-means(R) to DR to fit the table within margins. DP-means and DP-means(R) do not select any features. The number of clusters was chosen to be same as the number of classes in each dataset. The CRAFT algorithm exhibited low variance across runs (not shown here).

Dataset	**Average Purity**					**Average NMI**				
	CRAFT	**M**	**N**	**DR**	**D**	**CRAFT**	**M**	**N**	**DR**	**D**
Bank	0.67	0.65	0.59	0.61	0.61	0.16	0.06	0.02	0.03	0.03
Spam	0.72	0.64	0.64	0.61	0.61	0.20	0.05	0.05	0.00	0.00
Splice	0.75	0.62	0.63	0.61	0.52	0.20	0.04	0.05	0.05	0.01
Wine	0.71	0.72	0.69	0.66	0.66	0.47	0.35	0.47	0.44	0.44
Monk-3	0.56	0.55	0.53	0.54	0.53	0.03	0.02	0.00	0.00	0.00

Table 2: CRAFT versus DP-means and state-of-the-art feature selection methods ($m = 0.8$).

Dataset	**Average Purity**					**Average NMI**				
	CRAFT	**M**	**N**	**DR**	**D**	**CRAFT**	**M**	**N**	**DR**	**D**
Bank	0.64	0.61	0.61	0.61	0.61	0.08	0.03	0.03	0.03	0.03
Spam	0.72	0.64	0.64	0.61	0.61	0.23	0.05	0.05	0.00	0.00
Splice	0.74	0.68	0.63	0.61	0.52	0.18	0.09	0.05	0.05	0.01
Wine	0.82	0.73	0.69	0.66	0.66	0.54	0.42	0.42	0.44	0.44
Monk-3	0.57	0.54	0.54	0.54	0.53	0.03	0.00	0.00	0.00	0.00

the subspaces. We do not know whether the algorithm we presented is guaranteed to converge to a solution, since the interspersing of clustering and feature selection steps makes the analysis hard. We could plausibly devise alternative algorithms that are guaranteed to monotonically decrease the asymptotic objective. Despite the good time performance of our algorithm, there is further scope for scalability. The procedure samples new Bernoulli feature vectors during each iteration, and this randomness allows it to explore different regions of data. However, it would slow down the procedure for very high dimensional problems. It would be interesting to analyze whether the previously sampled random vectors can be reused once the algorithm has seen "enough" data. Finally, since most steps in our algorithm can be executed in parallel, it would be useful to apply CRAFT to real applications in a distributed setting.

Acknowledgments

VG and TJ were partially supported by a MITEI-Shell grant.

References

[1] A. Ahmad and L. Dey. A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63:503–527, 2007.

[2] S. Aranganayagi and K. Thangavel. Improved k-modes for categorical clustering using weighted dissimilarity measure. *World Academy of Science, Engineering and Technology*, 27:992–997, 2009.

[3] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *ACM- SIAM Symp. Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[5] T. Broderick, B. Kulis, and M. I. Jordan. Mad-bayes: Map-based asymptotic derivations from bayes. In *ICML*, 2013.

[6] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *KDD*, 2010.

[7] Y. Guan, J. G. Dy, and M. I. Jordan. A unified probabilistic model for global and local unsupervised feature selection. In *ICML*, 2011.

[8] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Pairwise variable selection for high-dimensional model-based clustering. *Biometrics*, 66(3):793–804, 2010.

[9] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, 2003.

[10] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Math. Operations Research*, 10(2):180–184, 1985.

[11] Z. Huang. Clustering large data sets with mixed numeric and categorical values. In *KDD*, 1997.

[12] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining Knowl. Discov.*, 2(2):283–304, 1998.

[13] B. Kim, C. Rudin, and J. Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *NIPS*, 2014.

[14] B. Kulis and M. I. Jordan. Revisiting k-means: new algorithms via bayesian nonparametrics. In *ICML*, 2012.

[15] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, pages 1026–1032, 2012.

[16] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 17(4):491–502, 2005.

[17] S. P. Lloyd. Least square quantization in pcm. Technical report, Bell Telephone Laboratories Paper, 1957.

[18] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Symp. Mathematical Statistics and Probability, Berkeley, CA*, pages 281–297, 1967.

[19] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[20] J. Pitman. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2):145–158, 1995.

[21] O. M. San, V. Huynh, and Y. Nakamori. An alternative extension of the k-means algorithm for clustering categorical data. *Int. J. Appl. Math. Comput. Sci.*, 14 (2):241–247, 2004.

[22] M. Shaflei and E. Milios. Latent dirichlet co-clustering. In *IEEE Int'l Conf. on Data Mining*, pages 542–551, 2006.

[23] K. Sohn and E. Xing. A hierarchical dirichlet process mixture model for haplotype reconstruction from multi-population data. *Annals of Applied Statistics*, 3 (2):791–821, 2009.

[24] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, Mar 2003. ISSN 1532-4435.

[25] W. Sun, J. Wang, and Y. Fang. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6: 148–167, 2012.

[26] UCI ML Repository. Data sets: (a) banknote authentication (bank), (b) spambase (spam), (c) wine, (d) splice junction determination (splice), and (e) monk-3 (monk), 2013. URL `http://archive.ics.uci.edu/ml`.

[27] S. Vempala. The random projection method. *American Mathematical Society*, 2004.

[28] D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *J Am Stat Assoc.*, 105 (490):713–726, 2010.

CRAFT: ClusteR-specific Assorted Feature selecTion (Supplementary)

6 Supplementary Material

We now derive the various objectives for the CRAFT framework. We first show the derivation for the generic objective that accomplishes feature selection on the assorted data. We then derive the degenerate cases when all features are retained and all data are (a) numeric, and (b) binary categorical. In particular, when the data are all numeric, we recover the DP-means objective [14].

6.1 Main Derivation: Clustering with Assorted Feature Selection

We have the total number of features, $D = |Cat| + |Num|$. We define $S_{N,k}$ to be the number of points assigned to cluster k. First, note that a Beta distribution with mean c_1 and variance c_2 has shape parameters $\dfrac{c_1^2(1-c_1)}{c_2} - c_1$ and $\dfrac{c_1(1-c_1)^2}{c_2} + c_1 - 1$. Therefore, we can find the shape parameters corresponding to m and ρ. Now, recall that for numeric data, we assume the density is of the form $f(x_{nd}|v_{kd})$

$$= \frac{1}{Z_{kd}} e^{-\left[v_{kd}\frac{(x_{nd}-\zeta_{kd})^2}{2\sigma_{kd}^2} + (1-v_{kd})\frac{(x_{nd}-\zeta_d)^2}{2\sigma_d^2}\right]}, \quad (3)$$

where Z_{kd} ensures that the area under the density is 1. Assuming an uninformative conjugate prior on the (numeric) means, i.e. a Gaussian distribution with infinite variance, and using the Iverson bracket notation for discrete (categorical) data, we obtain the joint probability distribution given in Fig. 6 for the underlying graphical model shown in Fig. 1. Note that joint distribution factorizes into a product of posterior distributions (e.g. the beta prior on the features conjugates with the feature likelihood to yield one posterior. We will show that under asymptotic conditions, minimizing the joint negative log-likelihood yields an intuitive objective function via simplification of the log-posteriors.

The total contribution of (3) to the negative joint log-likelihood

$$\begin{aligned} &= \sum_{k=1}^{K^+} \sum_{d \in Num} \sum_{n: z_{n,k}=1} \left[v_{kd} \frac{(x_{nd}-\zeta_{kd})^2}{2\sigma_{kd}^2} \right. \qquad (5) \\ &+ \left. (1-v_{kd}) \frac{(x_{nd}-\zeta_d)^2}{2\sigma_d^2} \right] + \sum_{k=1}^{K^+} \sum_{d \in Num} \log Z_{kd}. \end{aligned}$$

The contribution of the selected categorical features depends on the categorical means of the clusters, and is given by

$$-\log \left(\prod_{k=1}^{K^+} \prod_{n: z_{n,k}=1} \prod_{d \in Cat: v_{kd}=1} \prod_{t \in \mathcal{T}_d} \eta_{kdt}^{\mathbb{I}(x_{nd}=t)} \right).$$

On the other hand, the categorical features not selected are assumed to be drawn from cluster-independent global means, and therefore contribute

$$-\log \left(\prod_{k=1}^{K^+} \prod_{n: z_{n,k}=1} \prod_{d \in Cat: v_{kd}=0} \prod_{t \in \mathcal{T}_d} \eta_{0dt}^{\mathbb{I}(x_{nd}=t)} \right).$$

Thus, the total contribution of the categorical features is

$$\begin{aligned} -\sum_{k=1}^{K^+} \sum_{n: z_{n,k}=1} &\left[\sum_{d \in Cat: v_{kd}=1} \sum_{t \in \mathcal{T}_d} \mathbb{I}(x_{nd}=t) \log \eta_{kdt} \right. \\ &+ \left. \sum_{d \in Cat: v_{kd}=0} \sum_{t \in \mathcal{T}_d} \mathbb{I}(x_{nd}=t) \log \eta_{0dt} \right]. \end{aligned}$$

The Bernoulli likelihood on v_{kd} couples with the conjugate Beta prior on ν_{kd}. To avoid having to provide the value of ν_{kd} as a parameter, we take its point estimate to be the mean of the resulting Beta posterior, i.e., we set

$$\nu_{kd} = \frac{\left(\dfrac{m^2(1-m)}{\rho} - m\right) + v_{kd}}{\dfrac{m(1-m)}{\rho}} = \frac{a_{kd}}{a_{kd}+b_{kd}}, \quad (6)$$

where

$$\begin{aligned} a_{kd} &= \frac{m^2(1-m)}{\rho} - m + v_{kd}, \text{ and} \\ b_{kd} &= \frac{m(1-m)^2}{\rho} + m - v_{kd}. \end{aligned}$$

Then the contribution of the posterior to the negative log likelihood is

$$-\sum_{k=1}^{K^+} \sum_{d=1}^{D} \left[\log \left(\frac{a_{kd}}{a_{kd}+b_{kd}} \right)^{a_{kd}} + \log \left(\frac{b_{kd}}{a_{kd}+b_{kd}} \right)^{b_{kd}} \right],$$

or equivalently,

$$\sum_{k=1}^{K^+} \sum_{d=1}^{D} \underbrace{\left[\log\, (a_{kd}+b_{kd})^{(a_{kd}+b_{kd})} - \log\, a_{kd}^{a_{kd}} - \log\, b_{kd}^{b_{kd}} \right]}_{F(v_{kd})}.$$

$$
\begin{aligned}
&\mathbb{P}(x, z, v, \nu, \eta, \zeta, m) \\
&= \mathbb{P}(x|z, v, \eta, \zeta)\mathbb{P}(v|\nu)\mathbb{P}(z)\mathbb{P}(\eta)\mathbb{P}(\nu; m, \rho) \\
&= \prod_{k=1}^{K^+} \prod_{n:z_{n,k}=1} \Bigg[\Bigg(\prod_{d \in Cat: v_{kd}=1} \prod_{t \in \mathcal{T}_d} \eta_{kdt}^{\mathbb{I}(x_{nd}=t)} \Bigg) \Bigg(\prod_{d \in Cat: v_{kd}=0} \prod_{t \in \mathcal{T}_d} \eta_{0dt}^{\mathbb{I}(x_{nd}=t)} \Bigg) \\
&\quad \Bigg(\prod_{d' \in Num} \frac{1}{Z_{kd'}} e^{-\left[v_{kd'}(x_{nd'}-\zeta_{kd'})^2/(2\sigma_{kd'}^2) + (1-v_{kd'})(x_{nd'}-\zeta_{d'})^2/(2\sigma_{d'}^2))\right]} \Bigg) \Bigg] \\
&\quad \cdot \left[\prod_{k=1}^{K^+} \prod_{d=1}^{D} \nu_{kd}^{v_{kd}} (1-\nu_{kd})^{1-v_{kd}} \right] \cdot \left[\theta^{K^+-1} \frac{\Gamma(\theta+1)}{\Gamma(\theta+N)} \prod_{k=1}^{K^+} (S_{N,k}-1)! \right] \qquad (4) \\
&\quad \cdot \left[\prod_{k=1}^{K^+} \prod_{d \in Cat} \frac{\Gamma\left(\sum_{t \in \mathcal{T}_d} \dfrac{\alpha_{kdt}}{K^+}\right)}{\prod_{t \in \mathcal{T}_d} \Gamma\left(\dfrac{\alpha_{kdt}}{K^+}\right)} \prod_{t' \in \mathcal{T}_d} \eta_{kdt'}^{(\alpha_{kdt'}/K^+)-1} \right] \\
&\quad \cdot \prod_{k=1}^{K^+} \prod_{d=1}^{D} \frac{\Gamma\left(\dfrac{m(1-m)}{\rho} - 1\right) \nu_{kd}^{\left(\frac{m^2(1-m)}{\rho} - m - 1\right)} (1-\nu_{kd})^{\left(\frac{m(1-m)^2}{\rho} - (2-m)\right)}}{\Gamma\left(\dfrac{m^2(1-m)}{\rho} - m\right) \Gamma\left(\dfrac{m(1-m)^2}{\rho} - (1-m)\right)}
\end{aligned}
$$

Figure 6: Joint probability distribution for the generic case (both numeric and categorical features).

Since $v_{kd} \in \{0, 1\}$, this simplifies to

$$
\begin{aligned}
\sum_{k=1}^{K^+} \sum_{d=1}^{D} F(v_{kd}) &= \sum_{k=1}^{K^+} \sum_{d=1}^{D} \left[v_{kd}(F(1) - F(0)) + F(0)\right] \\
&= \left(\sum_{k=1}^{K^+} \sum_{d=1}^{D} v_{kd} \right) \Delta F + K^+ D F(0), \quad (7)
\end{aligned}
$$

where $\Delta F = F(1) - F(0)$ quantifies the change when a feature is selected for a cluster.

The numeric means do not make any contribution since we assumed an uninformative conjugate prior over $\mathbb{R}$. On the other hand, the categorical means contribute

$$
-\log \left[\prod_{k=1}^{K^+} \prod_{d \in Cat} \frac{\Gamma\left(\sum_{t \in \mathcal{T}_d} \dfrac{\alpha_{kdt}}{K^+}\right)}{\prod_{t \in \mathcal{T}_d} \Gamma\left(\dfrac{\alpha_{kdt}}{K^+}\right)} \prod_{t' \in \mathcal{T}_d} \eta_{kdt'}^{(\alpha_{kdt'}/K^+)-1} \right],
$$

which simplifies to

$$
\begin{aligned}
\sum_{k=1}^{K^+} \sum_{d \in Cat} \Bigg[&-\log \frac{\Gamma\left(\sum_{t \in \mathcal{T}_d} \dfrac{\alpha_{kdt}}{K^+}\right)}{\prod_{t \in \mathcal{T}_d} \Gamma\left(\dfrac{\alpha_{kdt}}{K^+}\right)} \\
&- \sum_{t' \in \mathcal{T}_d} \left(\frac{\alpha_{kdt'}}{K^+} - 1 \right) \log \eta_{kdt'} \Bigg]. \qquad (8)
\end{aligned}
$$

Finally, the Dirichlet process specifies a distribution over possible clusterings, while favoring assignments of points to a small number of clusters. The contribution of the corresponding term is

$$
-\log \left[\theta^{K^+-1} \frac{\Gamma(\theta+1)}{\Gamma(\theta+N)} \prod_{k=1}^{K^+} (S_{N,k}-1)! \right],
$$

or equivalently,

$$
-(K^+ - 1)\log\theta - \log \left(\frac{\Gamma(\theta+1)}{\Gamma(\theta+N)} \prod_{k=1}^{K^+} (S_{N,k}-1)! \right). \quad (9)
$$

The total negative log-likelihood is just the sum of terms in (5), (6), (7), (8), and (9). We want to maximize the joint likelihood, or equivalently, minimize the total negative log-likelihood. We would use asymptotics to simplify our objective. In particular, letting $\sigma_d \to \infty,\ \forall k \in [K^+]$ and $d \in Num$, and $\alpha_{kdt} \to K^+,\ \forall t \in \mathcal{T}_d,\ d \in Cat, k \in [K^+]$, setting $\log\theta$ to

$$
-\left(\lambda + \frac{\displaystyle\sum_{k=1}^{K^+} \sum_{d \in Cat} \log |\mathcal{T}_d| - \sum_{k=1}^{K^+} \sum_{d \in Num} \log Z_{kd}}{K^+ - 1} \right),
$$

and ignoring the term containing $S_{N,k}$ that contributes $\mathcal{O}(1)$, we obtain our objective for assorted feature selection:

$$\arg\min_{z,v,\eta,\zeta,\sigma} \underbrace{\sum_{k=1}^{K^+}\sum_{n:z_{n,k}=1}\sum_{d\in Num}\frac{v_{kd}(x_{nd}-\zeta_{kd})^2}{2\sigma_{kd}^2}}_{\text{Numeric Data Discrepancy}}$$

$$+\underbrace{(\lambda+DF_0)K^+}_{\text{Regularization Term}}+\underbrace{\left(\sum_{k=1}^{K^+}\sum_{d=1}^{D}v_{kd}\right)F_\Delta}_{\text{Feature Control}}$$

$$+\underbrace{\sum_{k=1}^{K^+}\sum_{d\in Cat}\Bigg[v_{kd}\left(\sum_{n:z_{n,k}=1}-\mathbb{I}(x_{nd}=t)\log\eta_{kdt}\right)}_{\text{Categorical Discrepancy Term I}}$$

$$\underbrace{+(1-v_{kd})\sum_{n:z_{n,k}=1}\sum_{t\in\mathcal{T}_d}-\mathbb{I}(x_{nd}=t)\log\eta_{0dt}\Bigg]}_{\text{Categorical Discrepancy Term II}},$$

where $\Delta F = F(1) - F(0)$ quantifies the change when a feature is selected for a cluster, and we have renamed the constants $F(0)$ and ΔF as F_0 and F_Δ respectively.

6.1.1 Setting ρ

Reproducing the equation for ν_{kd} from (6), since we want to ensure that $\nu_{kd} \in (0,1)$, we must have

$$0 < \frac{\left(\dfrac{m^2(1-m)}{\rho}-m\right)+v_{kd}}{\dfrac{m(1-m)}{\rho}} < 1.$$

Since $v_{kd} \in \{0,1\}$, this immediately constrains

$$\rho \in (0, m(1-m)).$$

Note that ρ guides the selection of features: a high value of ρ, close to $m(1-m)$, enables local feature selection (v_{kd} becomes important), whereas a low value of ρ, close to 0, reduces the influence of v_{kd} considerably, thereby resulting in global selection.

6.2 Degenerate Case: Clustering Binary Categorical Data without Feature Selection

In this case, the discrete distribution degenerates to Bernoulli, while the numeric discrepancy and the feature control terms do not arise. Therefore, we can replace the Iverson bracket notation by having cluster means μ drawn from Bernoulli distributions. Then, the joint distribution of the observed data x, cluster indicators z and cluster means μ is given by $\mathbb{P}(x,z,\mu)$

$$\begin{aligned}
&= \mathbb{P}(x|z,\mu)\mathbb{P}(z)\mathbb{P}(\mu)\\
&= \underbrace{\left[\prod_{k=1}^{K^+}\prod_{n:z_{n,k}=1}\prod_{d=1}^{D}\mu_{kd}^{x_{nd}}(1-\mu_{kd})^{1-x_{nd}}\right]}_{(A)}\\
&\cdot \underbrace{\left[\theta^{K^+-1}\frac{\Gamma(\theta+1)}{\Gamma(\theta+N)}\prod_{k=1}^{K^+}(S_{N,k}-1)!\right]}_{(B)} \qquad (10)\\
&\cdot \underbrace{\left[\prod_{k=1}^{K^+}\prod_{d=1}^{D}\frac{\Gamma\left(\dfrac{\alpha}{K^+}+1\right)}{\Gamma\left(\dfrac{\alpha}{K^+}\right)\Gamma(1)}\mu_{kd}^{\frac{\alpha}{K^+}-1}(1-\mu_{kd})^0\right]}_{(C)}.
\end{aligned}$$

The joint negative log-likelihood is

$$-\log\mathbb{P}(x,z,\mu) = -[\log(A)+\log(B)+\log(C)].$$

We first note that $\log(A)$

$$\begin{aligned}
&= \sum_{k=1}^{K^+}\sum_{n:z_{n,k}=1}\sum_{d=1}^{D}x_{nd}\log\mu_{kd}+(1-x_{nd})\log(1-\mu_{kd})\\
&= \sum_{k=1}^{K^+}\sum_{n:z_{n,k}=1}\sum_{d=1}^{D}x_{nd}\log\left(\frac{\mu_{kd}}{1-\mu_{kd}}\right)+\log(1-\mu_{kd})\\
&= \sum_{k=1}^{K^+}\sum_{n:z_{n,k}=1}\sum_{d=1}^{D}\Bigg[\log(1-\mu_{kd})+\mu_{kd}\log\left(\frac{\mu_{kd}}{1-\mu_{kd}}\right)\\
&\quad + x_{nd}\log\left(\frac{\mu_{kd}}{1-\mu_{kd}}\right)-\mu_{kd}\log\left(\frac{\mu_{kd}}{1-\mu_{kd}}\right)\Bigg]\\
&= \sum_{k=1}^{K^+}\sum_{n:z_{n,k}=1}\sum_{d=1}^{D}\Bigg[(x_{nd}-\mu_{kd})\log\left(\frac{\mu_{kd}}{1-\mu_{kd}}\right)\\
&\quad + \mu_{kd}\log\mu_{kd}+(1-\mu_{kd})\log(1-\mu_{kd})\Bigg]\\
&= \sum_{k=1}^{K^+}\sum_{n:z_{n,k}=1}\sum_{d=1}^{D}(x_{nd}-\mu_{kd})\log\left(\frac{\mu_{kd}}{1-\mu_{kd}}\right)-\mathbb{H}(\mu_{kd}),
\end{aligned}$$

where

$$\mathbb{H}(p) = -p\log p-(1-p)\log(1-p) \ \text{ for } p\in[0,1].$$

$\log(B)$ and $\log(C)$ can be computed via steps analogous to those used in assorted feature selection. Invoking the asymptotics by letting $\alpha \to K^+$, setting

$$\theta = e^{-\left(\lambda+\frac{K^+D}{K^+-1}\log\left(\frac{\alpha}{K^+}\right)\right)},$$

and ignoring the term containing $S_{N,k}$ that contributes $\mathcal{O}(1)$, we obtain the following objective:

$$\underset{z,\mu}{\operatorname{argmin}} \sum_{k=1}^{K^+} \lambda K^+ + \sum_{n:z_{n,k}=1} \sum_{d} \underbrace{\left[\mathbb{H}(\mu_{kd}) + (\mu_{kd} - x_{nd}) \log \left(\frac{\mu_{kd}}{1-\mu_{kd}} \right) \right]}_{\text{(Binary Discrepancy)}},$$

where the term (Binary Discrepancy) is an objective for binary categorical data, similar to the K-means objective for numeric data. This suggests a very intuitive procedure, which is outlined in Algorithm 3.

Algorithm 3 Clustering binary categorical data

```
Input: $x_1, \ldots, x_N \in \{0,1\}^D$: binary categorical data, and $\lambda > 0$: cluster penalty parameter.
Output: $K^+$: number of clusters and $l_1, \ldots, l_{K^+}$: clustering.
  1. Initialize $K^+ = 1$, $l_1 = \{x_1, \ldots, x_N\}$ and the mean $\mu_1$ (sample randomly from the dataset).
  2. Initialize cluster indicators $z_n = 1$ for all $n \in [N]$.
  3. Repeat until convergence
       • Compute $\forall k \in [K^+], d \in [D]$:
           $\mathbb{H}(\mu_{kd}) = -\mu_{kd} \log \mu_{kd} - (1-\mu_{kd}) \log(1-\mu_{kd})$.
       • For each point $x_n$
           – Compute the following for all $k \in [K^+]$:
               $d_{nk} = \sum_{d=1}^{D} \left[ \mathbb{H}(\mu_{kd}) + (\mu_{kd} - x_{nd}) \log \left( \frac{\mu_{kd}}{1-\mu_{kd}} \right) \right]$.
           – If $\min_k d_{nk} > \lambda$, set $K^+ = K^+ + 1$, $z_n = K^+$, and $\mu_{K^+} = x_n$.
           – Otherwise, set $z_n = \operatorname{argmin}_k d_{nk}$.
       • Generate clusters $l_1, \ldots, l_{K^+}$ based on $z_1, \ldots, z_{K^+}$: $l_k = \{x_n \mid z_n = k\}$.
       • For each cluster $l_k$, update $\mu_k = \frac{1}{|l_k|} \sum_{x \in l_k} x$.
```

In each iteration, the algorithm computes "distances" to the cluster means for each point to the existing cluster centers, and checks if the minimum distance is within λ. If yes, the point is assigned to the nearest cluster, otherwise a new cluster is started with the point as its cluster center. The cluster means are updated at the end of each iteration, and the steps are repeated until there is no change in cluster assignments over successive iterations.

We get a more intuitively appealing objective by noting that the objective (11) can be equivalently written as

$$\underset{z}{\operatorname{argmin}} \sum_{k=1}^{K^+} \sum_{n:z_{n,k}=1} \sum_{d} \mathbb{H}(\mu^*_{kd}) + \lambda K^+, \qquad (11)$$

where μ^*_{kd} denotes the mean of feature d computed by using points belonging to cluster k. characterizes the uncertainty. Thus the objective tries to minimize the overall uncertainty across clusters and thus forces similar points to come together. The regularization term ensures that the points do not form too many clusters, since in the absence of the regularizer each point will form a singleton cluster thereby leading to a trivial clustering.

6.3 Degenerate Case: Clustering Numerical Data without Feature Selection (Recovering DP-means)

In this case, there are no categorical terms. Furthermore, assuming an uninformative conjugate prior on the numeric means, the terms that contribute to the negative joint log-likelihood are

$$\prod_{k=1}^{K^+} \prod_{d'} \frac{1}{Z_{kd'}} e^{-\left[\frac{v_{kd'}(x_{nd'} - \zeta_{kd'})^2}{(2\sigma^2_{kd'})} + (1-v_{kd'}) \frac{(x_{nd'} - \zeta_{d'})^2}{(2\sigma^2_{d'})} \right]},$$

and

$$\theta^{K^+-1} \frac{\Gamma(\theta+1)}{\Gamma(\theta+N)} \prod_{k=1}^{K^+} (S_{N,k} - 1)!.$$

Taking the negative logarithms on both these terms and adding them up, setting $\log \theta$ to

$$-\left(\lambda + \frac{\sum_{k=1}^{K^+} \sum_{d'} \log Z_{kd'}}{K^+ - 1} \right),$$

and $v_{kd'} = 1$ (since all features are retained), letting $\sigma_{d'} \to \infty$ for all d', and ignoring the $\mathcal{O}(1)$ term containing $S_{N,k}$, we obtain

$$\underset{z}{\operatorname{argmin}} \sum_{k=1}^{K^+} \sum_{n:z_{n,k}=1} \sum_{d} \frac{(x_{nd} - \zeta^*_{kd})^2}{2\sigma^{*2}_{kd}} + \lambda K^+, \qquad (12)$$

where ζ^*_{kd} and σ^{*2}_{kd} are, respectively, the mean and variance of the feature d computed using all the points assigned to cluster k. This degenerates to the DP-means objective [14] when $\sigma^*_{kd} = 1/\sqrt{2}$, for all k and d. Thus, using a completely different model and analysis to [14], we recover the DP-means objective as a special case.