6 Full Proofs and Derivations

6.1 Derivation of Main Optimization Objective

The prior probability of a batch containing exactly n_k labels for each cluster $k \in \{1, \ldots, K\}$ is

$$\left(\frac{1}{K}\right)^N \frac{N!}{\prod_{k=1}^K n_k!},$$

and the likelihood is

$$p(Z = z_1, \ldots, z_N | Y = k_1, \ldots, k_N) = \prod_{i=1}^N \frac{\exp(-\frac{1}{2}(z_i - \mu_{k_i})\Sigma_{k_i}^{-1}(z_i - \mu_{k_i}))}{\sqrt{(2\pi)^d |\Sigma_{k_i}|}},$$

where d is the dimension of the feature space, and μ_k and Σ_k are the centroid and covariance matrix of the kth cluster, respectively.

If we further assume each cluster is spherical, with the same isotropic variance across all clusters, i.e., $\Sigma_k = \sigma^2 I, \text{ for } k \in \{1, \ldots, K\}$, then equation 6.1 simplifies to

$$\prod_{i=1}^N \frac{\exp(-\frac{1}{2\sigma^2}||z_i - \mu_{k_i}||^2)}{\sqrt{(2\pi\sigma)^d}},$$

and the full a posteriori is

$$p(Y|Z) \propto P(Y)P(Z|Y) = \left(\frac{1}{K}\right)^N \frac{N!}{\prod_{k=1}^K n_k!} \prod_{i=1}^N \frac{\exp(-\frac{1}{2\sigma^2}||z_i - \mu_{k_i}||^2)}{\sqrt{(2\pi\sigma)^d}}$$

$$\propto \frac{\prod_{i=1}^N \exp(-\frac{1}{2\sigma^2}||z_i - \mu_{k_i}||^2)}{\prod_{k=1}^K n_k!},$$

where we drop the constants that are independent of Y. Then, we obtain an optimization objective by minimizing the corresponding negative log likelihood as follows

$$\begin{aligned}
\arg\max_Y p(Y|Z) &\propto \arg\max_Y p(Y)P(Z|Y) = \\
&= \arg\max_Y \log(\prod_{i=1}^N \exp(-\tfrac{1}{2\sigma^2}||z_i - \mu_{k_i}||^2)) - \log(\prod_{k=1}^K n_k!) = \\
&= \arg\max_Y \sum_{i=1}^N -\tfrac{1}{2\sigma^2}||z_i - \mu_{k_i}||^2 - \sum_{k=1}^K \log(n_k!) = \\
&= \arg\min_Y \sum_{i=1}^N ||z_i - \mu_{k_i}||^2 + 2\sigma \sum_{k=1}^K \log(n_k!)\,.
\end{aligned}$$

6.2 Complexity of Main Optimization Objective

The main optimization objective is too slow to solve exactly. A common solution would involve interpreting the problem as the rectangular assignment problem, where clusters are works and data points are jobs. Then take the standard representation of the assignment problem as a flow network. Instead of adding one edge from the source vertex for each worker, add m parallel edges for each worker. For $k \in \{0, \ldots, m-1\}$ the , kth edge for a worker has capacity 1 and cost $\log k! - \log(k-1)! = \log k$. However, using standard solutions to the assignment problem would then result in complexity cubic in m, which is the batch size. This would be prohibitively slow for all but very small batch sizes.

6.3 Derivation of Greedy Approximation

We want to maximize the conditional probability of the Nth assignment in a batch, conditioned on the $N-1$ previous assignments:

$$\begin{aligned}
&\arg\max_{k_N=1,\dots,K} p(y_N = k | y_1 = k_1, \dots, y_{N-1} = k_{N-1}; Z) = \\
&\arg\max_{k_N=1,\dots,K} \frac{p(y_1 = k_1, \dots, y_N = k_N | Z)}{p(y_1 = k_1, \dots, y_{N-1} = k_{N-1} | Z)} = \\
&\arg\max_{k_N=1,\dots,K} \log p(y_1 = k_1, \dots, y_N = k_N | Z) - \\
&\quad - \log p(y_1 = k_1, \dots, y_{N-1} = k_{N-1} | Z) = \\
&\arg\max_{k_N=1,\dots,K} -\sum_{i=1}^{N} ||z_i - \mu_{k_i}||^2 + 2\sigma \sum_{k=1}^{K} \log(n'_k!) + \\
&\quad + \sum_{i=1}^{N-1} ||z_i - \mu_{k_i}||^2 - 2\sigma \sum_{k=1}^{K} \log(n_k!) = \\
&\arg\min_{k_N=1,\dots,K} \sum_{i=1}^{N} ||z_i - \mu_{k_i}||^2 - \sum_{i=1}^{N-1} ||z_i - \mu_{k_i}||^2 + \\
&\quad + 2\sigma (\sum_{k=1}^{K} \log(n'_k!) - \sum_{k=1}^{K} \log(n_k!)) = \\
&\arg\min_{k_N=1,\dots,K} ||z_N - \mu_{k_N}||^2 + 2\sigma (\sum_{k=1}^{K} \log(n_k!) - \sum_{k=1}^{K} \log(n_k!')) \,, \qquad (9)
\end{aligned}$$

where n_k is the number of points assigned to cluster k before the Nth assignment, and n'_k is the number assigned to the kth cluster after all assignments have been made. This means that

$$n'_k = \begin{cases} n_k + 1 & k = k_N \\ n_k & \text{otherwise} \,. \end{cases}$$

Thus, equation 9 becomes

$$\begin{aligned}
&\arg\min_{k_N=1,\dots,K} ||z_N - \mu_{k_N}||^2 + 2\sigma \log(n_{k_N} + 1!) - \log(n_{k_N}!) = \\
&\qquad \arg\min_{k_N=1,\dots,K} ||z_N - \mu_{k_N}||^2 + 2\sigma \log(n_{k_N} + 1) \,. \qquad (10)
\end{aligned}$$

6.4 Proof of Equivalence to Mutual Information Maximization

We want to show that the greedy algorithm that iteratively solves equation 6 can be interpreted as (a close approximation to) iteratively making whatever assignment will maximize the mutual information between the batch index i and the cluster labels. First note that, because the proposed model makes hard assignments, the entropy of cluster labels given the batch index is automatically zero, and so recalling that

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \,,$$

we see that the mutual information of the batch index i and the cluster labels equals the entropy of cluster labels. Below, we show that the proposed method, up to small approximation error, maximizes the entropy of cluster labels and, hence, the mutual information of cluster labels and the batch indices in each batch.

Lemma 1. *Let $X \in \mathbb{R}^{N\times K}$ be the matrix of already-made assignments in the current batch, and let $H^{(k)}$ be the marginal entropy after the new hard assignment is made to cluster k. Then*

$$H^{(k)} - H^{(k')} \approx \frac{1}{N+1}\left(\log(x_{k'}+1) - \log(x_{k'}+1)\right). \tag{11}$$

Proof. Let $X \in \mathbb{R}^{N\times K}$ be the matrix of already-made assignments in the current batch after N points have been assigned, so that H, the current marginal entropy of X, is given by:

$$H = -\sum_{j=1}^{K}\left(\frac{1}{N}\sum_{i=1}^{N} x_{ij}\right)\log\left(\frac{1}{N}\sum_{i=1}^{N} x_{ij}\right).$$

To simplify notation, let $x_j = \sum_{i=1}^{N} x_{ij}$. Then $H^{(k)}$, the marginal entropy after the new hard assignment is made to cluster k, is given by

$$\begin{aligned}
H^{(k)} =& -\frac{x_k+1}{N+1}\log\frac{x_k+1}{N+1} - \sum_{j=1,j\neq k}^{K}\frac{x_j}{N+1}\log\frac{x_j}{N+1}\\
=&\frac{-1}{N+1}\left((x_k+1)\log(x_k+1) + \sum_{j=1,j\neq k}^{K} x_j(\log x_j - \log(N+1))\right)\\
=&\frac{-1}{N+1}\left((x_k+1)\log(x_k+1) + \sum_{j=1,j\neq k}^{K} x_j\log x_j - \sum_{j=1,j\neq k}^{K} x_j\log(N+1)\right)\\
=&\frac{-1}{N+1}\left((x_k+1)\log(x_k+1) + \sum_{j=1,j\neq k}^{K} x_j\log x_j - N\log(N+1)\right)\\
=&\frac{-1}{N+1}\left((x_k+1)\log(x_k+1) + \sum_{j=1,j\neq k}^{K} x_j\log x_j\right) + \frac{N}{N+1}\log(N+1)
\end{aligned}$$

Now, consider the difference $H^{(k)} - H^{(k')}$ between the entropy after making assignment k vs. after making a different assignment k'.

$$\begin{aligned}
=&\frac{-1}{N+1}\left((x_k+1)\log(x_k+1) + \sum_{j=1,j\neq k}^{K} x_j\log x_j\right) - \left((x_{k'}+1)\log(x_{k'}+1) + \sum_{j=1,j\neq k'}^{K} x_j\log x_j\right) =\\
=&\frac{-1}{N+1}\left((x_k+1)\log(x_k+1) - (x_{k'}+1)\log(x_{k'}+1)\right) + \left(\sum_{j=1,j\neq k}^{K} x_j\log x_j - \sum_{j=1,j\neq k'}^{K} x_j\log x_j\right) =\\
=&\frac{-1}{N+1}\big((x_k+1)\log(x_k+1) - (x_{k'}+1)\log(x_{k'}+1)\big) + (x_{k'}\log x_{k'} - x_k\log x_k) =\\
=&\frac{-1}{N+1}\left(((x_k+1)\log(x_k+1) - x_k\log x_k) - ((x_{k'}+1)\log(x_{k'}+1) - x_{k'}\log x_{k'})\right)\\
\approx&\frac{-1}{N+1}\left((\log(x_k+1) + \frac{x_k}{x_k+1}) - (\log(x_{k'}+1) + \frac{x_{k'}}{x_{k'}+1})\right)\\
=&\frac{-1}{N+1}\left(\log(x_k+1) - \log(x_{k'}+1) - \frac{x_k - x_{k'}}{(x_k+1)(x_{k'}+1)}\right)\\
=&\frac{1}{N+1}\left(\log(x_{k'}+1) - \log(x_k+1) - \frac{x_{k'} - x_k}{(x_k+1)(x_{k'}+1)}\right)\\
=&\frac{1}{N+1}\left(\log(x_{k'}+1) - \log(x_k+1)\right) - \frac{1}{N+1}\left(\frac{x_{k'} - x_k}{(x_k+1)(x_{k'}+1)}\right)
\end{aligned}$$

where the fourth last line uses the fact that $\log n \approx H_n$ to make the substitution

$$x_k \log x_k \approx x_k(\log(x_k+1) - \frac{x_k}{x_k+1}),$$

and similarly for $x_{k'}$. Note that the term $\frac{1}{N+1}\frac{x_k - x_{k'}}{(x_k+1)(x_{k'}+1)}$ is 0 in expectation and has absolute value $\leq \frac{N}{(N+1)^2}$. If we drop this small error term, then we get

$$H^{(k)} - H^{(k')} \approx \frac{1}{N+1}\left(\log(x_{k'}+1) - \log(x_k+1)\right), \tag{12}$$

as desired. □

Lemma 2. *Assume that N data points in a batch have already been assigned. Let $\mathcal{L}(k)$ be the batch likelihood, under the a K-component Gaussian mixture model with isotropic variance σ^2 (as described in Section 3.2), after the $(N+1)$th data point is assigned to cluster k. Let H^k be, as above, the entropy of cluster sizes after the $(N+1)$th data point has been assigned to cluster k. Then maximizing the objective $\log \mathcal{L}(k) + \lambda H^k$ with respect to the $(N+1)$th cluster assignment gives the following optimization problem*

$$\underset{k\in\{1,\ldots,K\}}{\arg\min}\ ||z-\mu_k||^2 + \frac{2\lambda\sigma^2}{N+1}\log(x_k+1)$$

Proof. Maximizing $\log \mathcal{L}(k) + \lambda H^k$ with respect to the $(N+1)$th cluster assignment means we prefer to assign to cluster k over cluster k' if and only if

$$\log \mathcal{L}(k) + \lambda H^k > \log \mathcal{L}(k') + \lambda H^{k'} \iff \tag{13}$$

$$\log \mathcal{L}(k) - \log \mathcal{L}(k') > \lambda H^{k'} - \lambda H^k \iff \tag{14}$$

$$-\log \mathcal{L}(k) - (-\log \mathcal{L}(k')) < \lambda H^k - \lambda H^{k'} \iff \tag{15}$$

$$\log \mathcal{L}(k) - \log \mathcal{L}(k') < \frac{\lambda}{N+1}\left(\log(x_{k'}+1) - \log(x_k+1)\right), \tag{16}$$

where the last line uses lemma 1. Let z be the encoding of the $(N+1)$th point, as in Section 3.2. Then

$$\begin{aligned} -\log \mathcal{L}(k) &= -\log\left(\frac{\exp(-\frac{1}{2\sigma^2}||z-\mu_k||^2)}{\sqrt{2\pi\sigma^d}}\right) \\ &= \tfrac{1}{2}\log\left(2\pi\sigma^d\right) + \frac{1}{2\sigma^2}||z-\mu_k||^2. \end{aligned}$$

Subbing this into equation 16, we get

$$\left(\tfrac{1}{2}\log\left(2\pi\sigma^d\right) + \frac{1}{2\sigma^2}||z-\mu_k||^2\right) - \left(\tfrac{1}{2}\log\left(2\pi\sigma^d\right) + \frac{1}{2\sigma^2}||z-\mu_{k'}||^2\right) <$$

$$\frac{1}{N+1}\left(\log(x_{k'}+1) - \log(x_k+1)\right) \iff$$

$$\frac{1}{2\sigma^2}(||z-\mu_k||^2 - ||z-\mu_{k'}||^2) < \frac{\lambda}{N+1}\left(\log(x_{k'}+1) - \log(x_k+1)\right) \iff$$

$$\frac{1}{2\sigma^2}||z-\mu_k||^2 + \frac{\lambda}{N+1}\log\left(x_k+1\right) < \frac{1}{2\sigma^2}||z-\mu_k||^2 + \frac{\lambda}{N+1}\log\left(x_k+1\right) \iff$$

$$||z-\mu_k||^2 + \frac{2\lambda\sigma^2}{N+1}\log\left(x_k+1\right) < ||z-\mu_{k'}||^2 + \frac{2\lambda\sigma^2}{N+1}\log\left(x_{k'}+1\right).$$

Choosing pairwise between all k, k' as per equation 6.4 is equivalent to choosing k so as to minimize

$$||z-\mu_k||^2 + \frac{2\lambda\sigma^2}{N+1}\log\left(x_k+1\right).$$

□

Remark 3. This shows that our proposed method closely approximates a maximization of the entropy of cluster labels. There is some similarity to those methods, discussed in Section 2, that use an additional loss term to encourage greater entropy of soft assignments in each batch, but the important difference here is that we are maximizing the entropy of hard assignments.

Theorem 4. *Assume that N data points in a batch have already been assigned. Let $\mathcal{L}(k)$ be the batch likelihood, under the a K-component Gaussian mixture model with isotropic variance σ^2 (as described in Section 3.2), after the $(N+1)$th data point is assigned to cluster k. Let C and B be, respectively, random variables indicating the cluster assignments in the batch and the batch indices. Then, the method presented in Section 3.3 is equivalent, up to a small error term, to maximizing*

$$\log \mathcal{L}(k) + \lambda I(C; B)\,,$$

for some $\lambda \in \mathbb{R}$ that does not depend on the cluster assignments in the batch.

Proof. By Lemma 1, the method in equation 4 is equivalent to maximizing

$$\log \mathcal{L}(k) + \lambda H^k\,, \tag{17}$$

for $\lambda = \frac{1}{N+1}$. The mutual information $I(C; B)$ can be expressed in terms of entropy as

$$I(C; B) = H(C) - H(C|B)\,.$$

Moreoever, we are making hard assignments so, given the cluster index, the distribution over cluster labels has all the probability on one cluster and has zero entropy. This means

$$I(C; B) = H(C) - H(C|B) = H(C) - 0 = H(C)\,.$$

Subbing this into equation 17, the result follows. □

7 Modified Variance Maximization

As discussed in Section 4, one of the methods we compare to is that proposed by Zhong et al. (2020), which minimizes the sum of squares of the marginal soft assignments across a batch. The expectation of the square (and hence the sum of squares) can be decomposed as the square of the expectation plus the variance. Minimizing the sum of squares can then help to combat partition collapse as it involves minimizing the variance. However, empirically we find this method not to perform well, see Table 1. Here, we show that a simply modified version of this method performs better than the original, though still less well than our method. Results are presented in Table 3.

The modification is to just minimize variance directly, rather than via sum of squares. Note that this may be equivalent in some formulations, if the probability of membership across clusters for a single data point is normalized to sum to 1. Then the expectation of the sum of memberships is 1, because it is 1 deterministically, so the square of the expectation is also 1 deterministically and, in particular, is independent of the cluster assignments. This means that minimizing the sum of squares with respect to cluster assignments, is identical to minimizing variance with respect to cluster assignments. In our model this is true. The probability of membership depends only on the distance to the cluster centroids, and is conditionally independent across clusters, given the cluster centroids. Details are not given in Zhong et al. (2020) as to whether this holds in their method.

8 Calcuation of Entropies of Matrices

Let $h, s : \mathbb{R}^{4\times 3} \to \mathbb{R}^3$ be the functions that compute the marginal hard and soft cluster distributions for a given matrix of batch assignment probabilities. Then, for matrices

$$D_1 = \begin{bmatrix} .98 & .01 & .01 \\ .98 & .01 & .01 \\ .49 & .50 & .01 \\ .49 & .01 & .50 \end{bmatrix} \quad D_2 = \begin{bmatrix} .34 & .33 & .33 \\ .34 & .33 & .33 \\ .34 & .33 & .33 \\ .34 & .33 & .33 \end{bmatrix}\,,$$

we have

$$\begin{aligned} s(D_1) &= [.74, .13, .13] & H(h(D_1)) &= 1.10 \\ h(D_1) &= [.5, .25, .25] & H(s(D_1)) &= 1.50 \\ s(D_2) &= [.34., .33, .33] & H(s(D_2)) &= 1.58 \\ h(D_2) &= [1, 0, 0] & H(h(D_2)) &= 0\,. \end{aligned}$$

		CA	Var	VarM
Cifar10	Acc	**22.7 (2.07)**	11.8 (1.72)	20.8 (0.67)
	NMI	10.1 (1.68)	1.1 (1.15)	**11.2 (0.65)**
	ARI	5.8 (0.93)	0.3 (0.38)	**7.3 (0.50)**
	HVar	**425 (136)**	43542 (11424)	11515 (1524)
	SVar	407 (135)	**339 (92)**	10028 (1407)
	HEnt	**3.3 (0.01)**	0.9 (1.22)	1.3 (0.09)
	SEnt	**3.3 (0.01)**	2.6 (0.97)	1.5 (0.08)
Cifar100	Acc	**6.4 (0.22)**	1.2 (0.22)	1.0 (0.00)
	NMI	**13.2 (0.37)**	0.6 (1.05)	0.0 (0.00)
	ARI	**1.7 (0.14)**	0.0 (0.04)	0.0 (0.00)
	HVar	**1280 (156)**	55405 (3743)	59400 (0)
	SVar	190 (21)	**121 (62)**	12580 (391)
	HEnt	**5.4 (0.11)**	0.2 (0.17)	0.0 (0.00)
	SEnt	6.5 (0.17)	**6.5 (0.06)**	3.1 (0.03)
FashionMNIST	Acc	**54.5 (6.96)**	10.0 (0.04)	37.4 (2.53)
	NMI	**53.2 (4.23)**	0.0 (0.04)	42.8 (2.08)
	ARI	**39.1 (6.29)**	0.0 (0.00)	27.1 (1.78)
	HVar	**386 (51)**	53950 (98)	8550 (1001)
	SVar	**368 (40)**	376 (172)	6072 (3066)
	HEnt	**3.3 (0.01)**	0.0 (0.01)	1.5 (0.07)
	SEnt	**3.3 (0.00)**	3.1 (0.40)	1.6 (0.06)
STL	Acc	**23.5 (1.42)**	10.1 (0.20)	22.6 (1.52)
	NMI	**13.7 (1.33)**	0.0 (0.08)	11.4 (1.70)
	ARI	7.1 (0.70)	0.0 (0.00)	**7.1 (1.16)**
	HVar	**217 (21)**	11317 (765)	1321 (384)
	SVar	**194 (17)**	524 (247)	949 (303)
	HEnt	**3.2 (0.02)**	0.1 (0.16)	1.7 (0.13)
	SEnt	**3.2 (0.01)**	3.1 (0.12)	1.9 (0.10)

Table 3: Comparison between the modified variance minimization method, denoted 'VarM', the original variance minimization method from Zhong et al. (2020), denoted 'Var', and our method, denoted 'CA'.