B Proofs

We will use subscripts to denote rounds of k-means, and $B(x, r)$ to denote the closed ball centered on x of radius r.

B.1 Proof of correctness of Elkan's algorithm update

By the definition of the lower bound update,

$$l_{t_0+1}(i, j) = l_{t_0}(i, j) - p_{t_0}(j).$$

Using that l_{t_0} is a valid bound, the definition of p_{t_0}, and the triangle inequality,

$$\begin{aligned}
&\leq \|x(i) - c_{t_0}(j)\| - p_{t_0}(j), \\
&\leq \|x(i) - c_{t_0}(j)\| - \|c_{t_0}(j) - c_{t_0+1}(j)\| \\
&\leq \|x(i) - c_{t_0+1}(j)\|.
\end{aligned}$$

Thus the lower bound update is valid. Similarly for the upper bound,

$$\begin{aligned}
u_{t_0+1}(i, j) &= u_{t_0}(i) + p_{t_0}(a(i)), \\
&\geq \|x(i) - c_{t_0}(a(i))\| + p_{t_0}(a(i)), \\
&\geq \|x(i) - c_{t_0}(a(i))\| + \\
&\quad \|c_{t_0}(a(i)) - c_{t_0+1}(a(i))\|, \\
&\geq \|x(i) - c_{t_0+1}(a(i))\|.
\end{aligned}$$

This proves that the upper bound update is valid.

B.2 Proof of correctness of Elkan's algorithm intercentroid test

Suppose that,

$$\frac{cc(a(i), j)}{2} > u(i).$$

Then, by the triangle inequality and previous definitions,

$$\begin{aligned}
\|c(j) - x(i)\| &\geq \|c(j) - c(a(i))\| - \\
&\qquad \|c(a(i)) - x(i)\|, \\
&\geq cc(a(i), j) - u(i), \\
&\geq 2u(i) - u(i), \\
&\geq u(i).
\end{aligned}$$

Thus $c(a(i))$ is nearer to $x(i)$ than $c(j)$ is, and so $j \neq n_1(i)$.

B.3 Proof of correctness of Annular algorithm test

Recall the definition of $R(i)$,

$$R(i) = \max\left(u(i), \|x(i) - c(b(i))\|\right).$$

Following directly from this definition and the definition of $u(i)$, we have $c(a(i)), c(b(i)) \in B(x(i), R(i))$. Therefore by the definitions of $n_1(i)$ and $n_2(i)$, we have that $c(n_1(i)), c(n_2(i)) \in B(x(i), R(i))$. The triangle inequality now provides

$$|\|c(j)\| - \|x(i)\|| > R(i) \implies \|c(j) - x(i)\| > R(i), \tag{1}$$

Thus by the definition of $\mathcal{J}(i)$,

$$\mathcal{J}(i) = \{j : |\|c(j)\| - \|x(i)\|| \leq R(i)\},$$

we can say,

$$j \notin \mathcal{J}(i) \implies j \notin \{n_1(i), n_2(i)\}.$$

B.4 Proof of correctness of Exponion algorithm test

Let $nn(j) \in \{1, \ldots, k\} \setminus \{j\}$ denote the index the cluster whose centroid is nearest to the centroid of cluster j other than j, that is the centroid at distance $s(j)$ from centroid j.

By definitions we have

$$c(a(i)) \in B(x(i), u(i)),$$
$$c(nn(a(i))) \in B(c(a(i)), s(a(i))).$$

Combining these we have

$$c(a(i)), c(nn(a(i))) \in B(x(i), u(i) + s(a(i))), \tag{2}$$

Basic geometric arguments provide

$$B\left(x(i), u(i) + s(a(i))\right) \subseteq B(c(a(i)), 2u(i) + s(a(i))). \tag{3}$$

From (2) we deduce that

$$c(n_1(i)), c(n_2(i)) \in B(x(i), u(i) + s(a(i))),$$

and hence by (3) we have

$$c(n_1(i)), c(n_2(i)) \in (c(a(i)), 2u(i) + s(a(i))),$$

completing the proof.

B.5 Proof that ns upper bound is tighter than sn upper bound

$$\begin{aligned}
u^{ns}_{t_0+\delta t}(i) &= u_{t_0}(i) + \left\| \sum_{t'=t_0}^{t_0+\delta t-1} c_{t'+1}(i) - c_{t'}(i) \right\|, \\
&\leq u_{t_0}(i) + \sum_{t'=t_0}^{t_0+\delta t-1} \|c_{t'+1}(i) - c_{t'}(i)\|, \\
&\leq u^{sn}_{t_0+\delta t}(i).
\end{aligned}$$

C Detailed descriptions

C.1 The inner Yinyang test

We need some temporary notation to present the test which the Yinyang algorithm employs,

$$\begin{aligned} j_1(i,f) &= \underset{j \in \mathcal{G}(f)}{\arg\min} \|x(i) - c(j)\|, \\ j_2(i,f) &= \underset{j \in \mathcal{G}(f) \setminus \{j_1(f)\}}{\arg\min} \|x(i) - c(j)\|, \\ r_2(i,f) &= \|x(i) - c(j_2(f))\|. \end{aligned}$$

The Yinyang test hinges on the fact that centroids in $\mathcal{G}(f)$ which lie beyond radius $r_2(i,f)$ of $x(i)$ do not affect the variable updates and can thus be ignored. Extending this, suppose we have bounds $\tilde{r}_2(i,f)$ and $\tilde{l}(i,j)$ for $j \in \mathcal{G}$ such that $\tilde{r}_2(i,f) > r_2(f)$ and $\tilde{l}(i,j) < \|x(i) - c(j)\|$. Then $\tilde{r}_2(i,f) < \tilde{l}(i,j)$ means that centroid j can be ignored. It remains to define relevant bounds $\tilde{r}_2(i,f)$ and $\tilde{l}(i,j)$.

For $\tilde{r}_2(i,f)$, one keeps track of the second nearest centroid found thus far while looping over the centroids in $\mathcal{G}(f)$. Then for $\tilde{l}(i,j)$ we could take $l(i,f)$, but a better choice is $\tilde{l}(i,j) - q(f) + p(j)$, which replaces the maximum group displacement in the last round with the exact displacement of centroid j.

The Yinyang test to determine whether centroid j needs be considered is thus finally,

$$\begin{aligned} l(i,f) - q(f) + p(j) > \tilde{r}_2(i,f) \implies \\ \text{centroid } j \text{ lies beyond radius } r_2, \text{ can be ignored.} \end{aligned} \tag{4}$$

C.2 SMN, MSN, MNS

A lower bound to at time $t_0 + \delta_t$ on the distance from $x(i)$ to a group of centroids with group index f can be computed in three different ways. Letting Δ_{t_0,δ_t} denote the update term in

$$l_{t_0+\delta_t}(i,f) = \min_{j \in \mathcal{G}(i)} (\|x(i) - c_{t_0}(j)\|) - \Delta_{t_0,\delta_t},$$

the three possibilities are

$$\begin{aligned} \Delta^{SMN}_{t_0,\delta_t} &= \sum_{t'=t_0}^{t_0+\delta t-1} \max_{j \in \mathcal{G}(i)} (\|c_{t'+1}(j) - c_{t'}(j)\|), \\ \Delta^{MSN}_{t_0,\delta_t} &= \max_{j \in \mathcal{G}(i)} \left(\sum_{t'=t_0}^{t_0+\delta t-1} \|c_{t'+1}(j) - c_{t'}(j)\| \right), \\ \Delta^{MNS}_{t_0,\delta_t} &= \max_{j \in \mathcal{G}(i)} (\|c_{t_0+\delta_t}(j) - c_{t_0}(j)\|). \end{aligned}$$

The term $\Delta^{SMN}_{t_0,\delta_t}$ corresponds to the classic approach used in all previous works. The term $\Delta^{MSN}_{t_0,\delta_t}$ corresponds to an intermendiate where improved bounds can be obtained without storing centroids. The term $\Delta^{MNS}_{t_0,\delta_t}$ corresponds to the approach providing the tightest bounds, and is the one we use throughout.

	Data set	d	N
i	birch	2	100,000
ii	europe	2	169,300
iii	urand2	2	1,000,000
iv	ldfpads	3	164,850
v	conflongdemo	3	164,860
vi	skinseg	4	200,000
vii	tsn	4	200,000
viii	colormoments	9	68,040
ix	mv	11	40,760
x	wcomp	15	165,630
xi	house16h	17	22,780
xii	keggnet	28	65,550
xiii	urand30	30	1,000,000
xiv	mnist50	50	60,000
xv	miniboone	50	130,060
xvi	covtype	55	581,012
xvii	uscensus	68	2,458,285
xviii	kddcup04	74	145,750
xix	stl10	108	1,000,000
xx	gassensor	128	13,910
xxi	kddcup98	310	95,000
xxii	mnist784	784	60,000

Table 1: Fullnames of the 22 datasets used. All datasets are preprocessed such that features have mean zero and variance 1.

D Full Results Tables

	mean iterations	SD iterations	mean fastest [s]	SD fastest	bay-sta	mlp-sta	pow-sta	vlf-sta	own-sta	bay-ham	mlp-ham	own-ham	bay-ann	own-ann	own-exp	own-exp-ns	own-syin	own-syin-ns	pow-yin	own-yin	own-selk	own-selk-ns	bay-elk	mlp-elk	vlf-elk	own-elk	own-elk-ns
i	139	35.4	0.26	0.04	62.2	44.7	105	33.7	19.2	5.75	4.87	4.67	1.72	2.19	1.05	**1.00**	1.91	2.01	14.9	1.98	9.37	8.42	14.8	11.1	21.3	7.53	5.55
ii	335	132	1.69	0.25	38.2	28.1	60.3	21.5	10.3	6.44	5.00	5.05	1.71	1.99	1.07	**1.00**	3.05	3.13	14.6	2.96	6.08	5.60	10.8	8.70	15.9	6.94	6.70
iii	455	187	5.23	1.47	100	76.4	153	56.4	29.9	4.00	4.25	3.35	1.75	1.99	1.05	**1.00**	2.04	1.95	23.0	2.33	15.1	13.6	26.6	19.1	34.1	12.3	8.67
iv	270	96.3	2.17	0.48	29.0	17.9	37.7	15.2	6.79	3.99	2.90	2.51	2.03	1.64	1.04	**1.00**	1.02	1.06	6.78	1.09	3.66	3.30	7.09	6.68	11.0	4.54	3.80
v	277	73.6	2.35	0.48	27.5	18.1	35.7	14.4	6.80	4.07	3.06	2.55	2.05	1.66	1.04	**1.00**	1.01	1.05	6.48	1.08	3.47	3.14	6.72	6.57	10.5	4.31	3.63
vi	235	107	1.76	0.61	44.3	25.7	55.6	17.7	9.96	4.68	3.34	2.63	2.23	1.69	1.05	**1.00**	1.20	1.21	8.45	1.32	4.76	4.29	8.79	8.22	13.4	5.16	4.20
vii	117	25.2	1.32	0.18	28.9	15.9	34.2	11.7	5.85	4.29	2.83	2.42	2.61	1.76	1.25	1.19	1.02	**1.00**	5.95	1.06	3.24	2.89	5.72	4.90	8.82	3.19	2.60
viii	203	48.4	1.19	0.20	31.4	14.1	26.3	10.3	5.61	6.47	3.33	2.42	5.08	2.14	2.39	2.24	1.02	**1.00**	5.11	1.28	2.25	2.02	6.22	6.38	10.0	4.10	3.09
ix	127	35.4	0.49	0.07	32.6	14.7	25.5	12.0	5.02	7.80	4.02	2.66	7.99	2.75	2.63	2.51	1.02	**1.00**	4.83	1.32	2.12	1.89	6.12	6.12	9.77	3.43	2.37
x	130	27.2	2.16	0.46	38.7	17.2	29.2	12.0	5.11	10.7	5.21	2.97	4.90	1.83	1.23	1.21	**1.00**	1.03	4.56	1.39	1.89	1.67	3.29	3.28	5.46	2.00	1.67
xi	111	16.6	0.52	0.07	20.8	9.62	14.0	5.36	2.82	9.34	4.89	2.40	5.96	1.88	2.33	2.22	1.11	1.07	3.75	1.73	1.08	**1.00**	2.90	3.30	4.34	2.07	1.90
xii	113	33.4	0.92	0.14	53.3	23.3	30.2	10.3	4.86	12.0	4.21	2.11	5.58	1.47	1.82	1.75	**1.00**	1.00	4.36	1.74	1.66	1.46	5.26	3.64	4.84	1.95	1.60
xiii	2576	110	156	1.86	t	t	t	t	t	9.15	4.61	1.85	9.26	1.97	1.82	1.70	1.04	**1.00**	6.06	1.58	3.37	3.04	4.70	5.40	t	3.46	3.16
xiv	152	58.3	1.72	0.33	67.4	21.6	29.3	10.7	5.60	17.8	5.76	2.47	18.2	2.61	2.44	2.34	1.01	**1.00**	4.00	2.02	1.47	1.25	3.04	2.66	4.72	1.74	1.39
xv	376	90.1	7.18	1.13	87.5	27.2	37.9	13.7	6.97	25.4	7.84	3.40	15.3	2.44	3.27	3.09	**1.00**	1.06	4.53	2.05	1.54	1.38	4.20	4.22	5.72	2.84	2.76
xvi	127	35.0	8.45	0.94	120	38.9	51.4	20.3	10.8	18.1	6.34	2.67	15.2	2.51	1.92	1.85	**1.00**	1.04	4.72	2.05	1.85	1.60	3.38	4.01	5.73	2.01	1.58
xvii	102	38.6	43.9	6.73	t	28.3	m	14.0	6.09	19.5	6.11	2.55	17.5	2.47	2.34	2.26	1.00	**1.00**	m	2.29	1.37	m	3.27	3.44	4.77	1.83	m
xviii	334	118	13.5	3.46	63.9	18.7	26.0	9.58	4.85	23.8	6.90	2.89	22.3	2.90	2.86	2.81	1.14	1.12	4.63	2.75	1.13	**1.00**	2.51	1.91	3.51	1.31	1.16
xix	408	145	92.5	27.0	t	t	t	t	7.18	t	5.77	2.29	t	2.38	2.16	2.10	1.05	1.10	4.35	2.92	1.14	**1.00**	2.59	2.74	3.85	1.58	1.60
xx	57.3	15.9	0.23	0.05	101	28.2	40.0	14.6	6.73	33.7	9.58	3.65	15.8	2.27	3.37	3.25	1.72	1.78	6.14	4.56	1.12	1.13	2.53	2.21	2.41	**1.00**	1.02
xxi	178	58.8	9.96	1.97	121	30.1	44.0	16.7	7.98	41.4	10.5	3.72	42.1	3.94	3.66	3.57	1.47	1.47	5.76	4.64	1.18	**1.00**	3.10	2.07	2.39	1.32	1.14
xxii	131	25.9	9.89	1.23	143	34.8	50.1	20.0	10.0	40.1	10.2	3.97	40.6	4.10	3.87	3.75	1.29	1.34	4.23	3.64	1.25	**1.00**	3.63	1.82	1.71	1.27	1.03

Table 2: Results with $k = 100$ by dataset (rows). Columns 6 to the end contain mean times over the 11 initialisations, relative to the fastest algorithm, that is the algorithm with the lowest mean time, corresponding to the entry **1.00**. The mean and standard deviation of the number of iterations to convergence are given in columns 2 and 3. The mean and standard deviation of the time of the fastest algorithm are given in columns 4 and 5. 't' and 'm' correspond to timeout (40 minutes) and memory (4 GB) failures respectively. The fastest implementation for all data sets is always `own`. The fastest non-`own` implementation for each data set is underlined, where non-`own` implementations correspond to white columns.

	mean iterations	SD iterations	mean fastest [s]	SD fastest	bay-sta	mlp-sta	pow-sta	vlf-sta	own-sta	bay-ham	mlp-ham	own-ham	bay-ann	own-ann	own-exp	own-exp-ns	own-syin	own-syin-ns	pow-yin	own-yin	own-selk	own-selk-ns	bay-elk	mlp-elk	vlf-elk	own-elk	own-elk-ns
i	120	20.5	2.80	0.32	48.0	32.0	104	25.4	12.0	12.5	8.50	8.29	1.49	1.41	1.01	**1.00**	1.01	1.27	8.53	1.13	6.83	6.32	11.5	17.5	16.3	5.54	4.32
ii	533	66.7	12.1	1.22	82.6	56.0	t	43.7	21.6	19.0	13.0	12.5	2.05	1.76	1.01	**1.00**	1.62	2.10	15.2	1.90	11.6	11.1	19.9	36.2	26.6	13.0	11.9
iii	406	86.5	19.9	2.10	t	t	t	t	56.5	15.6	11.1	10.6	2.22	2.13	1.02	**1.00**	3.05	2.48	28.5	3.33	m	m	m	m	m	m	m
iv	193	46.5	7.22	1.07	60.4	34.2	95.4	30.2	13.5	17.8	11.0	10.0	4.49	2.83	1.01	**1.00**	1.15	1.34	9.19	1.32	6.93	6.43	12.0	24.9	18.6	7.51	6.32
v	197	25.2	7.16	0.55	62.1	35.1	98.1	31.1	13.7	17.3	10.7	9.79	4.41	2.77	1.02	**1.00**	1.15	1.34	9.17	1.31	7.10	6.61	12.4	25.3	19.3	7.65	6.39
vi	287	87.0	10.6	2.04	87.0	43.8	132	33.4	15.1	21.7	12.3	10.5	4.39	2.46	1.03	**1.00**	1.24	1.39	10.4	1.43	8.42	7.78	14.7	30.7	23.2	9.13	7.45
vii	94.0	21.8	4.17	0.41	71.7	36.2	105	28.0	12.6	15.6	9.18	7.83	4.85	2.80	1.02	**1.00**	1.25	1.37	9.10	1.39	7.23	6.71	11.9	19.2	18.3	6.08	4.86
viii	87.9	17.1	3.10	0.33	50.8	20.9	50.3	16.3	6.43	25.0	11.5	8.33	15.1	5.10	5.18	4.90	1.16	**1.00**	6.69	1.45	3.24	3.09	7.17	16.5	11.8	5.40	4.23
ix	51.1	5.74	1.28	0.07	48.7	20.8	44.6	17.6	6.10	24.3	11.7	7.40	23.0	6.90	5.01	4.80	1.21	**1.00**	6.13	1.51	2.92	2.73	6.39	15.2	9.73	4.30	3.51
x	201	41.6	12.5	1.30	102	42.0	90.3	30.7	11.3	50.6	22.7	12.9	12.6	3.54	1.96	1.91	**1.00**	1.11	6.93	1.36	4.28	3.95	7.80	16.9	13.1	5.41	4.64
xi	46.6	8.49	1.17	0.07	37.7	15.7	29.1	9.82	3.79	29.2	14.2	7.16	17.8	4.42	5.75	5.61	1.23	**1.00**	4.97	1.72	1.73	1.66	5.48	9.12	6.47	3.32	2.77
xii	32.8	3.81	1.87	0.07	73.4	31.5	47.5	15.0	6.86	28.9	8.55	4.36	9.20	2.09	1.99	1.95	1.04	**1.00**	4.60	1.37	2.17	2.03	12.8	6.60	5.04	2.07	1.57
xiii	738	108	382	31.1	t	t	t	t	t	t	t	t	t	t	t	t	1.14	**1.00**	t	1.87	m	m	m	m	m	m	m
xiv	58.9	7.76	5.05	0.21	88.2	26.8	43.5	14.0	5.87	55.6	16.9	8.14	57.4	8.18	7.97	7.67	1.36	**1.00**	5.15	2.47	1.75	1.46	5.77	5.68	6.33	3.05	1.96
xv	181	41.5	15.8	1.93	t	58.0	92.7	29.6	14.4	109	33.1	15.5	59.1	8.08	11.0	10.6	1.07	**1.00**	5.77	1.84	2.60	2.49	7.56	14.6	9.56	5.30	4.14
xvi	224	55.8	46.6	4.86	t	t	t	t	t	t	29.0	13.9	t	9.61	2.98	2.88	1.03	**1.00**	6.18	1.55	m	m	m	m	m	m	m
xvii	145	32.9	249	11.2	t	t	m	t	t	t	t	t	t	t	t	7.87	**1.00**	m	m	1.73	m	m	m	m	m	m	m
xviii	114	11.9	33.7	1.18	t	24.7	38.8	13.1	6.47	59.0	16.6	8.16	58.6	7.89	8.20	8.00	1.33	**1.00**	5.11	2.80	1.47	1.27	3.99	3.13	4.30	2.24	1.71
xix	612	160	587	76.3	t	t	t	t	t	t	t	t	t	t	t	t	1.09	**1.00**	t	2.63	m	m	m	m	m	m	m
xx	18.0	1.00	0.71	0.03	98.1	27.1	42.4	15.6	10.2	61.6	16.7	7.37	20.8	2.65	2.13	2.17	1.29	1.24	4.47	2.17	**1.00**	1.06	8.31	3.50	2.32	1.35	1.38
xxi	76.1	14.4	31.7	2.73	t	39.9	t	23.1	7.92	t	26.0	10.0	t	10.1	9.98	9.63	1.75	1.38	6.41	4.85	1.39	**1.00**	5.51	3.44	2.90	1.92	1.37
xxii	54.8	11.0	23.0	1.90	t	64.5	t	38.1	13.8	t	42.4	15.3	t	14.8	14.6	14.2	2.10	1.52	6.49	5.53	1.82	**1.00**	9.85	3.56	2.61	2.04	1.22

Table 3: As per Table 2, but with $k = 1000$