Discriminative Metric Learning by Neighborhood Gerrymandering (Supplementary Material)

Shubhendu Trivedi, David McAllester, Gregory Shakhnarovich
Toyota Technological Institute
Chicago, IL - 60637
{shubhendu,mcallester,greg}@ttic.edu

1 Proof of correctness of Algorithm 2

First of all it is easy to see that Algorithm 2 terminates. There are $k - n^*$ iterations after initialization (of the first n^* points) and this amounts to at most a linear scan of X. We need $O(N \log N)$ time to sort the data and then finding $\mathbf{h}^*$ involves $O(N)$, thus the algorithm runs in time $O(N \log N)$.

We need to prove that the algorithm returns h^* as defined earlier. First, we establish the correctness of setting n^*:

Proposition 1. *Let R be the number of classes, and let $\#(h, y)$ be the count of neighbors from target class y included in the assignment h. Then, $\Delta(y^*, h) = 0$ only if $\#(h, y^*) \geq n^*$, where*

$$n^* = \begin{cases} \lceil \frac{k+R-1}{R} \rceil & \textit{if ties not allowed}, \\ \lceil \frac{k}{R} \rceil & \textit{if ties allowed}. \end{cases}$$

We prove it below for the case with no ties; the proof when ties are allowed is very similar.

Proof. Suppose by contradiction that $\Delta(y^*, h) = 0$ and $\#(h, y^*) \leq \lceil \frac{k+R-1}{R} \rceil - 1$. Then, since no ties are allowed, for all $y \neq y^*$, we have $\#(h, y^*) \leq \lceil \frac{k+R-1}{R} \rceil - 2$, and

$$\sum_y \#(h, y) \leq (R-1) \left(\left\lceil \frac{k+R-1}{R} \right\rceil - 2 \right) \tag{1}$$

$$+ \left\lceil \frac{k+R-1}{R} \right\rceil - 1 \tag{2}$$

$$< k, \tag{3}$$

a contradiction to $|h| = k$. □

Next, we prove that the algorithm terminates and produces a correct result. For the purposes of complexity analysis, we consider R (but not k) to be constant, and number of examples from each class to be $O(N)$.

Claim 1. *Algorithm 2 terminates after at most $O((N + k) \log N)$ operations and produces an h such that $|h| = k$.*

Proof. The elements of X can be held in R priority queues, keyed by $D_{\mathbf{W}}$ values, one queue per class. Construction of this data structure is an $O(N \log N)$ operation, carried out before the algorithm starts. To initialize h with n^* values, the algorithm retrieves n^* top elements from the priority queue for class y^*. An $O(n^* \log N)$ operation. Then, for each of the iterations over l, the algorithm needs to examine at most one top element from R queues, which costs $O(\log N)$; each such iteration increases $|h|$ by one. Thus after $k - n^*$ iterations $|h| = k$; the total cost is thus $O(k \log N)$. Combined with the complexity of data structure construction mentioned above, this concludes the proof.

□

Note that for typical scenarios in which $N \gg k$, the cost will be dominated by the $N \log N$ data structure setup.

Claim 2. *Let h^* be returned by Algorithm 2. Then,*

$$h^* = \operatorname*{argmax}_{h:|h|=k,\, \Delta(y^*,h)=0)} S_{\mathbf{W}}(\mathbf{x}, h), \tag{4}$$

i.e., the algorithm finds the highest scoring h with total of k neighbors among those h that attain zero loss.

Proof. From Proposition 1 we know that if $\#(h, y) < n^*$, then h does not satisfy the $\Delta(\mathbf{x}, h) = 0$ condition. $|h| \geq n^*$ to (4) without altering the definition.

We will call h "optimal for l" if

$$h = \operatorname*{argmax}_{h:|h|=n^*+l,\, \#(h,y)\geq n^*,\, \Delta(y^*,h))=0} S_{\mathbf{W}}(\mathbf{x}, h).$$

We now prove by induction over l that this property is maintained through the loop over l in the algorithm.

Let $h^{(j)}$ denote choice of h after j iterations of the loop, i.e., $|h| = n^* + j$. Suppose that $h^{(l-1)}$ is optimal for $l - 1$. Now the algorithm selects $\mathbf{x}_a \in X$, such that

$$\mathbf{x}_a = \operatorname*{argmin}_{\mathbf{x}_i:\, y_i=y, \text{ or } \#(y_i)<\#(y)-\tau,\, \mathbf{x}_i \notin h} D_{\mathbf{W}}\left(\mathbf{x}, \mathbf{x}_i\right). \tag{5}$$

Suppose that $h^{(l)}$ is not optimal for l. Then there exists an $\mathbf{x}_b \in X$ for which $D_{\mathbf{W}}\left(\mathbf{x}, \mathbf{x}_b\right) < D_{\mathbf{W}}\left(\mathbf{x}, \mathbf{x}_a\right)$ such that picking $\mathbf{x}_b$ instead of $\mathbf{x}_a$ would produce h optimal for l. But $\mathbf{x}_b$ is not picked by the algorithm; this can only happen if conditions on the argmin in (5) are violated, namely, if $\#(y_b) = \#(y) - \tau$; therefore picking $\mathbf{x}_b$ would violate conditions of optimality of $h^{(l)}$, and we get a contradiction.

It is also clear that after initialization with k highest scoring neighbors in y^*, h is optimal for $l = 0$, which forms the base of induction. We conclude that $h^{(k-n^*)}$, i.e. the result of the algorithm, is optimal for $k - n^*$, which is equivalent to definition in (4).

□

2 Runtimes using different methods

Here we include the training times in seconds for one fold of each dataset. These timings are for a single partition, for optimal parameters for $k = 7$. These experiments were run on a 12-core Intel Xeon E5-2630 v2 @ 2.60GHz.

Dataset	DSLR	Caltech	Amazon	Webcam	Letters	USPS	Isolet
LMNN	358.11	1812.1	1545.1	518.7	179.77	782.66	1762.1
GB-LMNN	410.13	1976.4	1680.9	591.29	272.87	3672.9	2882.6
MLR	4.93	124.42	88.96	85.02	838.13	1281	33.20
MLNG	413.36	1027.6	2157.2	578.74	6657.3	3891.7	3668.9

3 Experimental results using different feature normalizations

k = 3

Dataset	Isolet	USPS	letters	DSLR	Amazon	Webcam	Caltech
d	170	256	16	800	800	800	800
N	7797	9298	20000	157	958	295	1123
C	26	10	26	10	10	10	10
Euclidean	-	-	-	26.71 $\pm$11	37.26 $\pm$2.3	23.39 $\pm$5.3	58.42 $\pm$3.7
LMNN	-	-	-	23.53 $\pm$7.6	**26.30 $\pm$1.6**	**11.53 $\pm$6.7**	43.72 $\pm$3.5
GB-LMNN	-	-	-	23.53 $\pm$7.6	**26.30 $\pm$1.6**	**11.53 $\pm$6.7**	**43.54 $\pm$3.5**
MLR	-	-	-	24.78 $\pm$14.2	32.35 $\pm$4.5	14.58 $\pm$3.5	52.18 $\pm$2.0
ITML	-	-	-	22.22 $\pm$9.9	32.67 $\pm$3.2	12.88 $\pm$6.1	51.74 $\pm$4.2
NCA	-	-	-	29.84 $\pm$8.1	33.72 $\pm$2.1	21.36 $\pm$4.9	54.50 $\pm$2.0
ours	-	-	-	**21.63 $\pm$6.1**	29.23 $\pm$2.8	14.58 $\pm$5.4	46.46 $\pm$2.4

k = 7

Dataset	Isolet	USPS	letters	DSLR	Amazon	Webcam	Caltech
Euclidean	-	-	-	32.46 $\pm$8.3	38.2 $\pm$1.6	27.46 $\pm$5.9	56.9 $\pm$2.9
LMNN	-	-	-	26.11 $\pm$8.6	25.47 $\pm$1.6	**10.51 $\pm$4.9**	41.77 $\pm$4.0
GB-LMNN	-	-	-	**25.48 $\pm$10.9**	**25.36 $\pm$1.7**	**10.51 $\pm$4.9**	**41.59 $\pm$3.6**
MLR	-	-	-	27.94 $\pm$9.0	30.16 $\pm$3.0	16.95 $\pm$3.4	49.51 $\pm$3.6
ITML	-	-	-	22.28 $\pm$8.8	32.88 $\pm$3.3	13.90 $\pm$6.3	50.59 $\pm$4.7
NCA	-	-	-	37.48 $\pm$8.2	33.09 $\pm$1.9	23.39 $\pm$5.3	51.74 $\pm$2.6
ours	-	-	-	**25.65 $\pm$7.1**	28.65 $\pm$2.3	17.29 $\pm$5.0	48.62 $\pm$1.7

k = 11

Dataset	Isolet	USPS	letters	DSLR	Amazon	Webcam	Caltech
Euclidean	-	-	-	35.02 $\pm$8.9	37.57 $\pm$2.3	30.51 $\pm$4.8	56.55 $\pm$2.4
LMNN	-	-	-	49.64 $\pm$5.7	**24.84 $\pm$2.1**	**10.17 $\pm$3.8**	43.19 $\pm$2.7
GB-LMNN	-	-	-	43.89 $\pm$5.6	25.16 $\pm$2.0	**10.17 $\pm$3.8**	**43.10 $\pm$3.1**
MLR	-	-	-	28.63 $\pm$7.7	30.48 $\pm$2.4	17.63 $\pm$5.3	48.18 $\pm$3.8
ITML	-	-	-	**24.82 $\pm$5.1**	31.10 $\pm$2.6	15.25 $\pm$6.3	50.32 $\pm$3.9
NCA	-	-	-	41.37 $\pm$4.7	32.88 $\pm$1.5	24.07 $\pm$8.4	51.20 $\pm$3.9
ours	-	-	-	32.44 $\pm$6.7	32.15 $\pm$2.8	17.65 $\pm$3.5	49.60 $\pm$2.6

Table 1: kNN error,for k=3, 7 and 11. Mean and standard deviation are shown for data sets on which 5-fold partition was used. These experiments were done after histogram normalization. Best performing methods are shown in bold. Note that the only non-linear metric learning method in the above is GB-LMNN

k = 3

Dataset	Isolet	USPS	letters	DSLR	Amazon	Webcam	Caltech
d	170	256	16	800	800	800	800
N	7797	9298	20000	157	958	295	1123
C	26	10	26	10	10	10	10
Euclidean	8.98	5.03	4.31 ±0.2	58.01 ±5.0	56.89 ±2.4	40.34 ±4.2	74.89 ±3.2
LMNN	4.17	5.38	3.26 ±0.1	23.53 ±5.6	28.08 ±2.2	**11.19 ±5.6**	44.97 ±2.6
GB-LMNN	**3.72**	5.03	2.50 ±0.2	23.53 ±5.6	28.08 ±2.2	11.53 ±5.5	44.70 ±2.4
MLR	17.32	8.42	45.70 ±18.7	35.69 ±7.6	**23.40 ±1.7**	20 ±4.6	47.11 ±1.7
ITML	6.86	**4.78**	4.35 ±0.2	24.82 ±10.9	34.77 ±4.7	12.20 ±4.1	53.97 ±3.2
NCA	5.07	5.18	4.39 ±1.1	24.19 ±5.8	29.54 ±1.4	12.88 ±4.9	46.84 ±2.0
ours	4.11	5.13	2.84 ±0.2	**22.77 ±5.9**	27.84 ±2.4	14.58 ±3.9	**44.54 ±2.9**

k = 7

Dataset	Isolet	USPS	letters	DSLR	Amazon	Webcam	Caltech
Euclidean	6.93	5.08	4.69 ±0.2	60.46 ±5.2	59.07 ±4.5	43.05 ±3.7	72.3 ±3.3
LMNN	4.04	5.28	3.53 ±0.2	24.15 ±9.0	28.19 ±2.8	13.56 ±4.5	43.90 ±2.4
GB-LMNN	**3.72**	5.03	2.32 ±0.2	24.80 ±8.1	28.29 ±3.1	13.14 ±5.8	43.54 ±2.2
MLR	23.28	8.12	33.61 ±16.8	38.17 ±10.9	**23.79 ±3.9**	20.34 ±2.9	45.60 ±4.8
ITML	5.90	5.23	4.93 ±0.5	**23.57 ±9.6**	32.46 ±3.2	**11.19 ±5.7**	52.63 ±3.3
NCA	5.52	4.98	5.06 ±1.1	37.58 ±5.7	31.01 ±2.0	16.81 ±5.9	43.90 ±2.4
ours	4.07	**4.93**	3.13 ±0.2	29.94 ±7.6	27.62 ±3.4	13.24 ±3.1	**42.83 ±3.1**

k = 11

Dataset	Isolet	USPS	letters	DSLR	Amazon	Webcam	Caltech
Euclidean	7.95	5.68	5.26 ±0.2	61.71 ±6.4	61.48 ±3.7	49.15 ±3.9	73.1 ±3.6
LMNN	**3.85**	5.73	4.09 ±0.2	49.6 ±5.5	27.04 ±1.8	14.58 ±4.6	44.61 ±1.3
GB-LMNN	3.98	6.33	**2.96 ±0.1**	45.18 ±10.5	27.25 ±2.2	14.58 ±4.6	45.55 ±6.9
MLR	33.61	10.26	35.50 ±16.5	34.40 ±8.2	**24.21 ±3.4**	18.31 ±5.3	46.04 ±1.9
ITML	7.18	5.88	5.35 ±0.3	28.04 ±7.7	33.09 ±2.1	**12.54 ±5.4**	51.91 ±3.3
NCA	5.52	5.03	5.8 ±1.3	45.18 ±6.5	32.47 ±1.7	19.32 ±7.5	**44.17 ±2.6**
ours	3.87	**4.98**	3.28 ±0.4	**27.50 ±8.1**	27.91 ±3.5	14.24 ±6.5	45.76 ±2.9

Table 2: kNN error,for k=3, 7 and 11. No feature scaling was applied in these experiments. Mean and standard deviation are shown for data sets on which 5-fold partition was used. Best performing methods are shown in bold. Note that the only non-linear metric learning method in the above is GB-LMNN.