Sample Complexity of Learning Mahalanobis Distance Metrics

Nakul Verma
Janelia, HHMI
Mahalanobis Metric Learning

Comparing observations in feature space:

\[ \rho(x_1, x_2) = \|x_1 - x_2\|^2 \]
\[ = (x_1 - x_2)^T(x_1 - x_2) \quad \text{[sq. Euclidean dist]} \]

(all features are equally weighted)

\[ \rho_M(x_1, x_2) = \|M(x_1 - x_2)\|^2 \quad \text{(using weighting mechanism } M) \]
\[ = (x_1 - x_2)^T(M^TM)(x_1 - x_2) \quad \text{[sq. Mahalanobis dist]} \]

Q: What should be the correct weighting \( M \)?
A: Data-driven.

Given data of interest, learn a metric \( (M) \), which helps in the prediction task.
Learning a Mahalanobis Metric

Suppose we want $M$ s.t.:
- data from **same class** $\leq$ distance $U$
- data from **different classes** $\geq$ distance $L$ \quad [U < L]

Given two labelled samples $(x_i, y_i), (x_j, y_j)$ from a sample $S$. Then
- the distance between the pair $\rho_M^{ij} = \rho_M(x_i, x_j)$
- label agreement between the pair $Y_{ij} = 1[y_i = y_j]

Define a **pairwise penalty** function $\phi(\rho_M^{ij}, Y_{ij}) = \begin{cases} (\rho_M^{ij} - U)_+ & \text{if } Y_{ij} = 1 \\ (L - \rho_M^{ij})_+ & \text{otherwise} \end{cases}$

So total error:
$\text{err}_S(M) = \text{avg}_{(x_i,y_i), (x_j,y_j) \in S} [\phi(\rho_M^{ij}, Y_{ij})]$ \quad $\text{err}(M) = \mathbb{E}[\phi(\rho_M^{ij}, Y_{ij})]$

(empirical error over the sample $S$) \quad (generalization error)
(error of $M$ over the sample $S$) \quad (error of $M$ over the (unseen) population)

\[ \rho_M(x_i, x_j) = (x_i - x_j)^T(M^T M)(x_i - x_j) \]
Statistical consistency of Metric Learning

Best possible metric on the population:
\[ M^* = \arg\min_M \text{ err}(M) \]
\[ \text{err}(M) = \mathbb{E}\left[\phi(\rho_M^{ij}, Y_{ij})\right] \]
\[ \text{err}_S(M) = \text{avg}_S\left[\phi(\rho_M^{ij}, Y_{ij})\right] \]

Best possible metric on the sample \( S \) (of size \( m \)) [drawn independently from the population]
\[ M_{m}^* = \arg\min_M \text{ err}_{S_m}(M) \]

Questions we want to answer:
(i) Does \( \text{err}(M_{m}^*) \rightarrow \text{err}(M^*) \) as \( m \rightarrow \infty \)? (consistency)
(ii) At what rate does \( \text{err}(M_{m}^*) \rightarrow \text{err}(M^*) \)? (finite sample rates)
(iii) What factors affect the rate? (data dim, feature info content)
What we show: Theorem 1

Given a $D$-dimensional feature space. For any $\lambda$-Lipschitz penalty function $\phi$, and any sample size $m$,

$$
err(M_m^*) - err(M^*) \leq O\left(\lambda \sqrt{\frac{D \ln(1/\delta)}{m}}\right)
$$

(with probability at least $1-\delta$ over the draw of the sample)

If we want $err(M_m^*) - err(M^*) \leq \epsilon$,

then we require $m \geq \Omega\left(D \ln(1/\delta) \frac{\lambda^2}{\epsilon^2}\right)$

This gives us consistency as well as a rate!

Question: Is the convergence rate on the data dimension $D$ tight?
What we show: Theorem 2

Given a $D$-dimensional feature space.

For any metric learning algorithm $A$ that (given a sample $S_m$) returns

$$ A(S_m) = \arg\min_M \text{avg}_{S_m} \left[ \phi(\rho_M^{ij}, Y_{ij}) \right] $$

There exists a $\lambda$-Lipschitz penalty function $\phi$, s.t. for all $\varepsilon, \delta$,

if sample size $m \leq O(D/\varepsilon^2)$

then

$$ P_{S_m} \left[ \text{err}(A(S_m)) - \text{err}(M^*) > \varepsilon \right] > \delta $$

Remark: this is the worst case analysis in the absence of any other information about the data distribution.

Can we refine our results if we know about the quality of our feature set?

Dependence on the representation dimension $D$ is tight!
Quantifying feature-set quality

Quantifying the quality of our feature set.

Each feature has a different information content for the prediction task.

Fix a particular prediction task $T$.
Let $M$ be the optimal feature weighting for task $T$.
Define the *metric learning complexity* $d^*$ for task $T$ as:

$$d^* = \|M^T M\|_F^2$$

Observation: not all features are created equal

$d^*$ is unknown a priori

Question: Can we get a sample complexity rate that only depends on $d^*$?
What we show: Theorem 3

Given a $D$-dimensional feature space, and a prediction task $T$ with (unknown) metric learning complexity $d^*$
For any $\lambda$-Lipschitz penalty function $\phi$, and any sample size $m$, 

$$\text{err}(M_{m}^{\text{reg}}) - \text{err}(M^*) \leq O\left(\lambda \sqrt{\frac{d^* \ln(D) \ln(1/\delta)}{m}}\right)$$

(With probability at least $1-\delta$ over the draw of the sample)

$$M_{m}^{\text{reg}} = \arg\min_M \left[ \text{avg}_S [\phi(\rho_{ij}^M, Y_{ij})] + \Lambda \| M^T M \|_F \right]$$

$\Lambda \approx \lambda \sqrt{\ln(D/\delta)/m}$

Take home message:

*regularization can help adapt to the unknown metric learning complexity!*
Empirical Evaluation

Want to study

Given a dataset with small metric learning complexity, but high representation dimension. How do regularized vs. unregularized Metric Learning algs. fare?

Approach

• pick benchmark datasets of low dimensionality ($d$)
• augment each dataset with large ($D$ dim.) corr. noise

\[ \Sigma_D \sim \text{Wishart(unit-scale)} \]

for each orig. sample $x_i$, augmented sample $x_i = [x_i \ x_\sigma]$ $x_\sigma \sim N(0, \Sigma_D)$

(we can now control signal-noise ratio)

• study the prediction accuracy of regularized & unregularized Metric Learning algorithms as a function of noise dimension.

<table>
<thead>
<tr>
<th>UCI dataset</th>
<th>dim ($d$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>4</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>34</td>
</tr>
</tbody>
</table>
Empirical Evaluation
Theorem 1

Given a $D$-dimensional feature space.
For any $\lambda$-Lipschitz penalty function $\phi$ and any sample size $m$,

$$\text{err}(M_m^*) - \text{err}(M^*) \leq O\left(\lambda \sqrt{\frac{D \ln(1/\delta)}{m}}\right)$$

(with probability at least $1-\delta$ over the draw of the sample)

If we want $\text{err}(M_m^*) - \text{err}(M^*) \leq \epsilon$
then we require $m \geq \Omega\left(D \ln(1/\delta) \frac{\lambda^2}{\epsilon^2}\right)$

This gives us consistency as well as a rate!
Proof Idea (Theorem 1)

Want to find a sample size \( m \) such that for any weighting \( M \) the empirical performance of \( M \) is approximately equal to the generalization performance of \( M \):

\[
\text{err}(M) = \mathbb{E}[\phi(\rho^i_M, Y_{ij})] \\
\text{err}_S(M) = \text{avg}_S[\phi(\rho^i_M, Y_{ij})]
\]

Then, choosing the best \( M \) on samples, will have close to best generalization performance!

Try 1 (covering argument)

Fix a weighting metric \( M \), define random variable

\[
Z^M_{i,j} = \phi(\rho^i_M, Y_{ij}) \quad \in [0, 1]
\]

if \( \phi \) is bounded per example pair

By Hoeffding’s bound (since \( Z \) is bounded r.v.)

\[
\left| \text{avg}_{S_m} [Z^M_{i,j}] - \mathbb{E}[Z^M_{i,j}] \right| \leq \sqrt{\frac{\ln(2/\delta)}{2m}}
\]

w.p. \( \geq 1-\delta \) over the draw of \( S_m \)

But, we want to have a similar result for all \( M \)!

For all \( M \in \mathcal{M} \)

\[
\left| \text{avg}_{S_m} [Z^M_{i,j}] - \mathbb{E}[Z^M_{i,j}] \right| \leq O\left(\sqrt{\frac{D^2 \ln(1/\delta)}{m}}\right)
\]

\( \mathcal{M} \) Collection of all \( M \)
Proof Idea (Theorem 1)

Try 2 (VC argument)  
Recall: VC-theory is only for binary classification

If we view metric learning as classification, we can apply VC-style results!

Recall: given a (binary) classification class $F$, for all $f \in F$

$$|\text{avg}_{S_m} [1[f(x_i) \neq y_i]] - \mathbb{E}[1[f(x_i) \neq y_i]]| \leq O\left(\sqrt{\frac{Q_F \ln(1/\delta)}{m}}\right)$$

w.p. $\geq 1 - \delta$ over the draw of $S_m$

$Q_F = \text{maximum sample size that can achieve all possible labels from using } f \in F$

For metric learning: say penalty function $\phi$ is binary threshold on distance.

$$F = \{ \phi_M : M \in \mathcal{M} \}$$

+ labeling $\Rightarrow \rho_M$ for a pair is small

− labeling $\Rightarrow \rho_M$ for a pair is large

$$\text{err}(M) = \mathbb{E}[\phi(\rho_{M}^{i,j}, Y_{i,j})]$$

$$\text{err}_S(M) = \text{avg}_S[\phi(\rho_{M}^{i,j}, Y_{i,j})]$$

what is the maximum number of pairs which can attain all labeling from $F$?

$$Q_F \leq O(D^2)$$

(VC complexity of ellipsoids)

(a) only works for thresholds on $\phi$

(b) cannot adapt to quality of the feature space!
Proof Idea (Theorem 1)

Try 3 (Rademacher Complexity argument)

**Rademacher Complexity**: given a class $F$, how well does some $f \in F$ correlate to binary noise $\sigma \in \{-1, 1\}$.

$$\mathcal{R}_F^m = \mathbb{E}_{x_i} \mathbb{E}_{\sigma_i} \sup_f \left| \frac{1}{m} \sum_{i} \sigma_i f(x_i) \right|$$

Then for all $f$

$$\mathbb{E} [f(x_i)] - \text{avg}_{S_m} [f(x_i)] \leq 2\mathcal{R}_F^m + O\left( \sqrt{\frac{\ln(1/\delta)}{m}} \right) \quad \text{w.p.} \geq 1 - \delta \text{ over the draw of } S_m$$

For metric learning

$$\mathcal{R}_F^m \leq O\left( \sqrt{\frac{\sup_M \| M^T M \|^2}{m}} \right)$$

(a) works for any Lipschitz $\phi$

(b) can adapt to quality of the feature space!

*for scale restricted metrics $M$, $\|M^T M\|^2 \leq D*
Theorem 2

Given a $D$-dimensional feature space.

For any metric learning algorithm $A$ that (given a sample $S_m$) returns

$$A(S_m) = \arg \min_M \text{avg}_{S_m} [\phi(\rho^i_M, Y_{ij})]$$

There exists a $\lambda$-Lipschitz penalty function $\phi$, s.t. for all $\epsilon$, $\delta$, if sample size $m \leq O(D/\epsilon^2)$ then

$$P_{S_m} [\text{err}(A(S_m)) - \text{err}(M^*) > \epsilon] > \delta$$

Dependence on the representation dimension $D$ is tight!

How can we prove this?
Proof Idea (Theorem 2)

Try 1: (VC argument, by treating Metric Learning as classification)
If we can lower bound \( \mathcal{D}_F \geq m \),
then a standard construction gives a **specific** distribution on which we must have \( \Omega(m/\varepsilon^2) \) samples to get accuracy within \( \varepsilon \).

\[ \text{Since, we work with pairs of points, the specific distribution for VC argument doesn’t actually ever occur! (we need this distribution to be a product distribution)} \]

Try 2: (Our approach -- deconstruct the VC argument)
We’ll use the **probabilistic method**.
- Create a **collection of distributions** such that if one of them is chosen at random then the generalization error of \( M \) returned by \( A \) would be large.

So there is **some distribution** in the collection which has large error.

\[ \text{These distributions constructed so that Metric Learning acts as classification.} \]
Proof Idea (Theorem 2)

Construction: (point masses on the vertices regular simplex)

- Collection of distributions:
  each vertex is labeled + or – (randomly) with bias $\frac{1}{2} + \varepsilon$

- Loss function:

  $\phi(\rho^i_M, Y^i) = \begin{cases} 
  (\rho^i_M - U)^+ & \text{if } Y^i = 1 \\
  (L - \rho^i_M)^+ & \text{otherwise}
  \end{cases}$

Key insight: for this collection of distributions and this loss function the problem reduces to binary classification in the product space!

For $m$ i.i.d. samples from a randomly selected dist. from the collection any empirical error minimizing algorithm would require $m \geq \Omega(D/\varepsilon^2)$

How? Calculate minimum number of samples required to distinguish the bias of two coins. Repeat it for $\sim D/2$ pairs.

Other possible approaches:

Use information-theoretic arguments to establish minimum number of samples needed to distinguish good metric from bad ones. (e.g. use Fano’s inequality)
Theorem 3

Given a $D$-dimensional feature space, and a prediction task $T$ with (unknown) metric learning complexity $d^*$.

For any $\lambda$-Lipschitz penalty function $\phi$ and any sample size $m$,

$$\text{err}(M_{m}^{\text{reg}}) - \text{err}(M^*) \leq O\left(\lambda \sqrt{\frac{d^* \ln(D) \ln(1/\delta)}{m}}\right)$$

(with probability at least $1-\delta$ over the draw of the sample)

Take home message:

regularization can help adapt to the unknown metric learning complexity!
Using Rademacher complexity argument, already shown:

\[ \text{err}(M_m^*) - \text{err}(M^*) \leq O \left( \lambda \sqrt{\frac{\sup_M \|M^T M\|_F^2}{m} \cdot \ln\left(\frac{1}{\delta}\right)} \right) \leq D \]

If we know \( M^* \) has small norm (say \( d \ll D \)), then we are done!

*but don’t know the norm of the best metric a priori...*

Will use a refinement trick...

*Observation: we are allowed to fail \( \delta \) fraction of time, we distribute this over each class \( \delta / D \)*

For all \( d \leq D \) and all \( M^d \) (s.t. \( \|M^d^T M^d\|^2 \leq d \))

\[ \text{err}(M^d) - \text{err}(M^d) \leq O \left( \lambda \sqrt{\frac{d \cdot \ln(D/\delta)}{m}} \right) \]

w.p. \( \geq 1 - \delta \) over the draw of sample of size \( m \)
Proof Idea (Theorem 3)

\[ \text{err}(M^d) - \text{err}(M^d) \leq O\left(\lambda \sqrt{d \cdot \ln(D/\delta) / m}\right) \]

So, if the algorithm picks:

\[ M_{m}^{\text{reg}} = \arg\min_{M} \left[ \text{avg}_S[\phi(\rho_M^{ij}, Y_{ij})] + \Lambda\|M^T M\|_F \right] \]

\[ \Lambda \approx \lambda \sqrt{\ln(D/\delta)/m} \]

Then (w.p. \( \geq 1-\delta \)):

\[ \text{err}(M_{m}^{\text{reg}}) - \text{err}(M^*) \leq \text{err}_S(m, M_{m}^{\text{reg}}) + \Lambda\|M_{m}^{\text{reg}}^T M_{m}^{\text{reg}}\|_F - \text{err}(M^*) \]

\[ \leq \text{err}_S(m, M^*) + \Lambda\|(M^*)^T (M^*)\|_F - \text{err}(M^*) \]

\[ = O\left(\lambda \sqrt{d^* \ln(D/\delta) / m}\right) \]
Comparison with previous results

<table>
<thead>
<tr>
<th></th>
<th>Previous results</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Convergence rate</strong></td>
<td>For thresholds on convex $\phi$</td>
<td>For general Lipschitz $\phi$ with ERM</td>
</tr>
<tr>
<td>(upper bound)</td>
<td>$\leq O\left(\sqrt{?/m}\right)$ Stable and regularized algs.</td>
<td>$\leq O\left(\sqrt{D/m}\right)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Theorem 1</a></td>
</tr>
<tr>
<td><strong>Convergence rate</strong></td>
<td>No known results</td>
<td>In absence of any other information, exists Lipschitz $\phi$, with ERM</td>
</tr>
<tr>
<td>(lower bound)</td>
<td></td>
<td>$\geq \Omega\left(\sqrt{D/m}\right)$</td>
</tr>
<tr>
<td><strong>Data complexity</strong></td>
<td>No known results</td>
<td>For gen. Lipschitz $\phi$ with regularized ERM</td>
</tr>
<tr>
<td>$d^*$</td>
<td></td>
<td>$\leq O\left(\sqrt{d^* \ln(D)/m}\right)$</td>
</tr>
</tbody>
</table>
Open problems

- Analysis of Metric Learning in Online and Active Learning framework?

- Non-linear metric learning?

- ‘Structured’ metric learning? (ranking problems, clustering problems, etc)
Thank You!