Introduction

- Problem: Learn a Mahalanobis distance function subject to linear constraints.
Introduction

- Problem: Learn a Mahalanobis distance function subject to linear constraints
- Information-theoretic viewpoint
  - Bijection between Gaussian distributions and Mahalanobis distances
  - Natural entropy-based objective
Introduction

- Problem: Learn a Mahalanobis distance function subject to linear constraints
- Information-theoretic viewpoint
  - Bijection between Gaussian distributions and Mahalanobis distances
  - Natural entropy-based objective
- Connections with kernel learning
Introduction

- Problem: Learn a Mahalanobis distance function subject to linear constraints
- Information-theoretic viewpoint
  - Bijection between Gaussian distributions and Mahalanobis distances
  - Natural entropy-based objective
- Connections with kernel learning
- Fast and simple methods
  - Based on Bregman’s method for convex optimization
  - No eigenvalue computations are needed!
Learning a Mahalanobis Distance

- Given $n$ points $\{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$
- Given inequality constraints relating pairs of points
  - Similarity constraints: $d_A(x_i, x_j) \leq u$
  - Dissimilarity constraints: $d_A(x_i, x_j) \geq \ell$
Learning a Mahalanobis Distance

- Given $n$ points $\{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$
- Given inequality constraints relating pairs of points
  - Similarity constraints: $d_A(x_i, x_j) \leq u$
  - Dissimilarity constraints: $d_A(x_i, x_j) \geq \ell$
- Problem: Learn a Mahalanobis distance that satisfies these constraints:

$$d_A(x_i, x_j) = (x_i - x_j)^T A (x_i - x_j)$$
Learning a Mahalanobis Distance

- **Given** $n$ points $\{x_1, ..., x_n\}$ in $\mathbb{R}^d$
- **Given** inequality constraints relating pairs of points
  - Similarity constraints: $d_A(x_i, x_j) \leq u$
  - Dissimilarity constraints: $d_A(x_i, x_j) \geq \ell$
- **Problem**: Learn a Mahalanobis distance that satisfies these constraints:

$$d_A(x_i, x_j) = (x_i - x_j)^T A(x_i - x_j)$$

- **Applications**
  - $k$-means clustering
  - Nearest neighbor searches

Jason V. Davis, Brian Kulis, Suvrit Sra, and Inderjit Dhillon
Information-Theoretic Metric Learning
Mahalanobis Distance and the Multivariate Gaussian

Problem: How to choose the ‘best’ Mahalanobis distance from the feasible set?

Solution: Regularize by choosing that which is ‘closest’ to Euclidean distance.

Bijection between the multivariate Gaussian and the Mahalanobis Distance

\[ p(x; m, A) = \frac{1}{Z} \exp \left( -\frac{1}{2} (x - m)^T A (x - m) \right) \]

Allows for comparison of two Mahalanobis distances.

Differential relative entropy between the associated Gaussians:

\[ \text{KL}(p(x; m_1, A_1) \| p(x; m_2, A_2)) = \int p(x; m_1, A_1) \log \frac{p(x; m_1, A_1)}{p(x; m_2, A_2)} \, dx \]
Mahalanobis Distance and the Multivariate Gaussian

Problem: How to choose the ‘best’ Mahalanobis distance from the feasible set?

Solution: Regularize by choosing that which is ‘closest’ to Euclidean distance.

Bijection between the multivariate Gaussian and the Mahalanobis Distance:

\[ p(x; m, A) = \frac{1}{Z} \exp \left( -\frac{1}{2} (x - m)^T A (x - m) \right) \]

Allows for comparison of two Mahalanobis distances:

\[ \text{Differential relative entropy between the associated Gaussians:} \]

\[ \text{KL}(p(x; m_1, A_1) \parallel p(x; m_2, A_2)) = \int p(x; m_1, A_1) \log p(x; m_1, A_1) p(x; m_2, A_2) \, dx. \]
Mahalanobis Distance and the Multivariate Gaussian

Problem: How to choose the ‘best’ Mahalanobis distance from the feasible set?

Solution: Regularize by choosing that which is ‘closest’ to Euclidean distance

Bijection between the multivariate Gaussian and the Mahalanobis Distance

\[ p(x; m, A) = \frac{1}{Z} \exp \left( -\frac{1}{2} (x - m)^T A (x - m) \right) \]

Differential relative entropy between the associated Gaussians:

\[ \text{KL}(p(x; m_1, A_1) \parallel p(x; m_2, A_2)) = \int p(x; m_1, A_1) \log p(x; m_1, A_1) p(x; m_2, A_2) \, dx. \]
Mahalanobis Distance and the Multivariate Gaussian

Problem: How to choose the ‘best’ Mahalanobis distance from the feasible set?

Solution: Regularize by choosing that which is ‘closest’ to Euclidean distance

Bijection between the multivariate Gaussian and the Mahalanobis Distance

\[ p(x; m, A) = \frac{1}{Z} \exp \left( -\frac{1}{2} (x - m)^T A (x - m) \right) \]

- Allows for comparison of two Mahalanobis distances
- Differential relative entropy between the associated Gaussians:

\[ \text{KL}(p(x; m_1, A_1) \| p(x; m_2, A_2)) = \int p(x; m_1, A_1) \log \frac{p(x; m_1, A_1)}{p(x; m_2, A_2)} \, dx. \]
Goal: Minimize differential relative entropy subject to pairwise inequality constraints

\[
\begin{align*}
\min & \quad \text{KL}(p(\mathbf{x}; \mathbf{m}, A) \| p(\mathbf{x}; \mathbf{m}, I)) \\
\text{subject to} & \quad d_A(\mathbf{x}_i, \mathbf{x}_j) \leq u \quad (i, j) \in S, \\
& \quad d_A(\mathbf{x}_i, \mathbf{x}_j) \geq \ell \quad (i, j) \in D \\
A & \succ 0
\end{align*}
\]
Overview: Optimizing the Model

- Show an equivalence between our problem and a low-rank kernel learning problem [Kulis, 2006]
  - Yields closed-form solutions to compute the problem objective
  - Shows that the problem is convex
Overview: Optimizing the Model

- Show an equivalence between our problem and a low-rank kernel learning problem [Kulis, 2006]
  - Yields closed-form solutions to compute the problem objective
  - Shows that the problem is convex
- Use this equivalence to solve our problem efficiently
Low-Rank Kernel Learning

- Given $X = [x_1 \ x_2 \ \ldots \ x_n]$, $x_i \in \mathbb{R}^d$, define $K_0 = X^T X$
- Constraints: similarity ($S$) or dissimilarity ($D$) between pairs of points
- Objective: Learn $K$ that minimizes the divergence to $K_0$

Jason V. Davis, Brian Kulis, Suvrit Sra, and Inderjit Dhillon
Low-Rank Kernel Learning

- Given $X = [x_1 \ x_2 \ ... \ x_n]$, $x_i \in \mathbb{R}^d$, define $K_0 = X^T X$
- Constraints: similarity ($S$) or dissimilarity ($D$) between pairs of points
- Objective: Learn $K$ that minimizes the divergence to $K_0$

$$\min_D \ D_{\text{Burg}}(K, K_0)$$
subject to

\[
\begin{align*}
K_{ii} + K_{jj} - 2K_{ij} &\leq u \\
(i, j) &\in S, \\
K_{ii} + K_{jj} - 2K_{ij} &\geq \ell \\
(i, j) &\in D, \\
K &\succeq 0
\end{align*}
\]

$D_{\text{Burg}}$ is the Burg divergence

$$D_{\text{Burg}}(K, K_0) = \text{Tr}(KK_0^{-1}) - \log \det(KK_0^{-1}) - n$$

Jason V. Davis, Brian Kulis, Suvrit Sra, and Inderjit Dhillon
Information-Theoretic Metric Learning
Equivalence to Kernel Learning

*Kulis, 2006* Let $K$ be the optimal solution to the low-rank kernel learning problem.

- Then $K$ has the same range space as $K_0$
- $K = XX^T W^T WX$
[Kulis, 2006] Let $K$ be the optimal solution to the low-rank kernel learning problem.

1. Then $K$ has the same range space as $K_0$
2. $K = X^T W^T W X$

**Theorem:** Let $K = X^T W^T W X$ be an optimal solution to the low-rank kernel learning problem.

Then $A = W^T W$ is an optimal solution to the corresponding metric learning problem.
Lemma 1: $D_{\text{Burg}}(K, K_0) = 2\text{KL}(p(x; m, A) \| p(x; m, I)) + c$

- Establishes that the objectives for the problem are the same
- Builds on a recent connection relating the relative entropy between Gaussians and the Burg divergence [Davis, 2006]
Proof Sketch

**Lemma 1:** $D_{\text{Burg}}(K, K_0) = 2KL(p(x; m, A)\|p(x; m, I)) + c$

- Establishes that the objectives for the problem are the same
- Builds on a recent connection relating the relative entropy between Gaussians and the Burg divergence [Davis, 2006]

**Lemma 2:** Given $K = X^T A X$, $A$ is feasible if and only if $K$ is feasible.
Optimization via Bregman’s Method

- Solve the associated kernel learning problem via Bregman’s method
  - Dual ascent method
  - Iteratively projects onto one constraint at a time
  - Closed-form updates are known for this projection

Jason V. Davis, Brian Kulis, Suvrit Sra, and Inderjit Dhillon
Information-Theoretic Metric Learning
Optimization via Bregman’s Method

- Solve the associated kernel learning problem via Bregman’s method
  - Dual ascent method
  - Iteratively projects onto one constraint at a time
  - Closed-form updates are known for this projection
- Running time per iteration: $O(cd^2)$
  - Works on the kernel in factored form
  - Uses closed-form Bregman projections
Solve the associated kernel learning problem via Bregman’s method
  ▶ Dual ascent method
  ▶ Iteratively projects onto one constraint at a time
  ▶ Closed-form updates are known for this projection

Running time per iteration: $O(cd^2)$
  ▶ Works on the kernel in factored form
  ▶ Uses closed-form Bregman projections

Requires no eigenvalue decomposition
Extensions

- Minimizing KL-divergence to a different Mahalanobis matrix
  - inverse of the sample covariance matrix
Extensions

- Minimizing $KL$-divergence to a different Mahalanobis matrix
  - inverse of the sample covariance matrix
- Slack variables
Extensions

- Minimizing $KL$-divergence to a different Mahalanobis matrix
  - inverse of the sample covariance matrix
- Slack variables
- General linear inequality constraints
  - e.g. Relative distance comparisons [Schutz, 2003]
Experimental Methodology

- Goal: learn a Mahalanobis function for $kNN$ classification

- Approach:
  - Constrain points in the same class to be similar
  - Constrain points in different class to be dissimilar
  - Upper and lower bounds determined empirically
  - Sample 100 such constraints
  - No parameter tuning
  - Evaluate via cross-validation
Experimental Methodology

- Goal: learn a Mahalanobis function for $kNN$ classification
- Approach:
  - Constrain points in the same class to be similar
  - Constrain points in different class to be dissimilar
  - Upper and lower bounds determined empirically

Jason V. Davis, Brian Kulis, Suvrit Sra, and Inderjit Dhillon

Information-Theoretic Metric Learning
Experimental Methodology

- Goal: learn a Mahalanobis function for $kNN$ classification
- Approach:
  - Constrain points in the same class to be similar
  - Constrain points in different class to be dissimilar
  - Upper and lower bounds determined empirically
  - Sample 100 such constraints
  - No parameter tuning
Goal: learn a Mahalanobis function for $kNN$ classification

Approach:
- Constrain points in the same class to be similar
- Constrain points in different class to be dissimilar
- Upper and lower bounds determined empirically
- Sample 100 such constraints
- No parameter tuning

Evaluate via cross-validation
Experimental Results

- ITML: Information-Theoretic Metric Learning
- Sample Cov: parametrize Mahalanobis distance by the inverse of the sample covariance of the data
- LDA: Linear Discriminant Analysis
- MCML: Maximally Collapsing Metric Learning [Globerson, 2005]
Experimental Results

- **ITML**: Information-Theoretic Metric Learning
- **Sample Cov**: parametrize Mahalanobis distance by the inverse of the sample covariance of the data
- **LDA**: Linear Discriminant Analysis
- **MCML**: Maximally Collapsing Metric Learning [Globerson, 2005]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ITML</th>
<th>Sample Cov</th>
<th>Euclidean</th>
<th>LDA</th>
<th>MCML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance-scale</td>
<td>0.9312</td>
<td>0.9072</td>
<td>0.9120</td>
<td>0.9312</td>
<td>.9536</td>
</tr>
<tr>
<td>Wine</td>
<td>0.8315</td>
<td>0.8258</td>
<td>0.8427</td>
<td>0.7303</td>
<td>.8034</td>
</tr>
<tr>
<td>Iris</td>
<td>1.0000</td>
<td>0.9733</td>
<td>0.9667</td>
<td>1.0000</td>
<td>.9600</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>0.9915</td>
<td>0.9858</td>
<td>0.9829</td>
<td>0.5128</td>
<td>.9915</td>
</tr>
<tr>
<td>Soybean</td>
<td>0.9283</td>
<td>0.9429</td>
<td>0.9283</td>
<td>0.9385</td>
<td>.9590</td>
</tr>
</tbody>
</table>
Presented an information-theoretic formulation for metric learning

Given an equivalence between this problem and low-rank kernel learning

Provided efficient algorithms

Experiments are promising, but much more work is needed!