Information-Theoretic Metric Learning

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

The University of Texas at Austin

December 9, 2006

Presenter: Jason V. Davis

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 $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ in $\Re^d$
- Given inequality constraints relating pairs of points
  - Similarity constraints: $d_A(\mathbf{x_i}, \mathbf{x_j}) \leq u$
  - Dissimilarity constraints: $d_A(\mathbf{x_i}, \mathbf{x_j}) \geq \ell$

Learning a Mahalanobis Distance

- Given n points $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ in $\Re^d$
- Given inequality constraints relating pairs of points
  - Similarity constraints: $d_A(\mathbf{x_i}, \mathbf{x_j}) \leq u$
  - Dissimilarity constraints: $d_A(\mathbf{x_i}, \mathbf{x_j}) \geq \ell$
- Problem: Learn a Mahalanobis distance that satisfies these constraints:

$$d_A(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} - \mathbf{x_j})^T A(\mathbf{x_i} - \mathbf{x_j})$$

Learning a Mahalanobis Distance

- Given n points $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ in $\Re^d$
- Given inequality constraints relating pairs of points
  - Similarity constraints: $d_A(\mathbf{x_i}, \mathbf{x_j}) \leq u$
  - Dissimilarity constraints: $d_A(\mathbf{x_i}, \mathbf{x_j}) \geq \ell$
- Problem: Learn a Mahalanobis distance that satisfies these constraints:

$$d_A(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} - \mathbf{x_j})^T A(\mathbf{x_i} - \mathbf{x_j})$$

- Applications
  - k-means clustering
  - Nearest neighbor searches

Mahalanobis Distance and the Multivariate Gaussian

- Problem: How to choose the 'best' Mahalanobis distance from the feasible set?

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

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

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

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

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

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

$$\mathrm{KL}(p(\mathbf{x};\mathbf{m_1},A_1)\|p(\mathbf{x};\mathbf{m_2},A_2)) = \int p(\mathbf{x};\mathbf{m_1},A_1)\log\frac{p(\mathbf{x};\mathbf{m_1},A_1)}{p(\mathbf{x};\mathbf{m_2},A_2)}\,d\mathbf{x}.$$

Problem Formulation

Goal: Minimize differential relative entropy subject to pairwise inequality constraints

$$
\begin{aligned}
\min \quad & \mathrm{KL}(p(\mathbf{x};\mathbf{m},A)\|p(\mathbf{x};\mathbf{m},I)) \\
\text{subject to} \quad & d_A(\mathbf{x}_i,\mathbf{x}_j) \le u \qquad (i,j) \in S, \\
& d_A(\mathbf{x}_i,\mathbf{x}_j) \ge \ell \qquad (i,j) \in D \\
& A \succ 0
\end{aligned}
$$

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 $X = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_n]$, $\mathbf{x}_i \in \Re^d$, 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

Low-Rank Kernel Learning

- Given $X = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_n]$, $\mathbf{x}_i \in \Re^d$, 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

$$\begin{aligned} \min \quad & D_{\text{Burg}}(K, K_0) \\ \text{subject to} \quad & K_{ii} + K_{jj} - 2K_{ij} \le u \qquad (i,j) \in S, \\ & K_{ii} + K_{jj} - 2K_{ij} \ge \ell \qquad (i,j) \in D, \\ & K \succeq 0 \end{aligned}$$

- D_{Burg} is the Burg divergence

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

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
- $K = X^T W^T W X$

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
- $K = X^T W^T W X$

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

Proof Sketch

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

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

- 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 $K = X^T AX$, 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

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

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
- 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

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

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

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 Results

- ITML: Information-Theroetic 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-Theroetic 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]

Dataset	ITML	Sample Cov	Euclidean	LDA	MCML
Balance-scale	0.9312	0.9072	0.9120	0.9312	.9536
Wine	0.8315	0.8258	0.8427	0.7303	.8034
Iris	1.0000	0.9733	0.9667	1.0000	.9600
Ionosphere	0.9915	0.9858	0.9829	0.5128	.9915
Soybean	0.9283	0.9429	0.9283	0.9385	.9590

Conclusion

- 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!