Kernel Learning with Bregman Matrix Divergences

Inderjit S. Dhillon

The University of Texas at Austin

Workshop on Algorithms for Modern Massive Data Sets

Stanford University and Yahoo! Research

June 22, 2006

Joint work with Brian Kulis, Mátyás Sustik and Joel Tropp

Kernel Methods

Kernel methods are important for many problems in machine learning

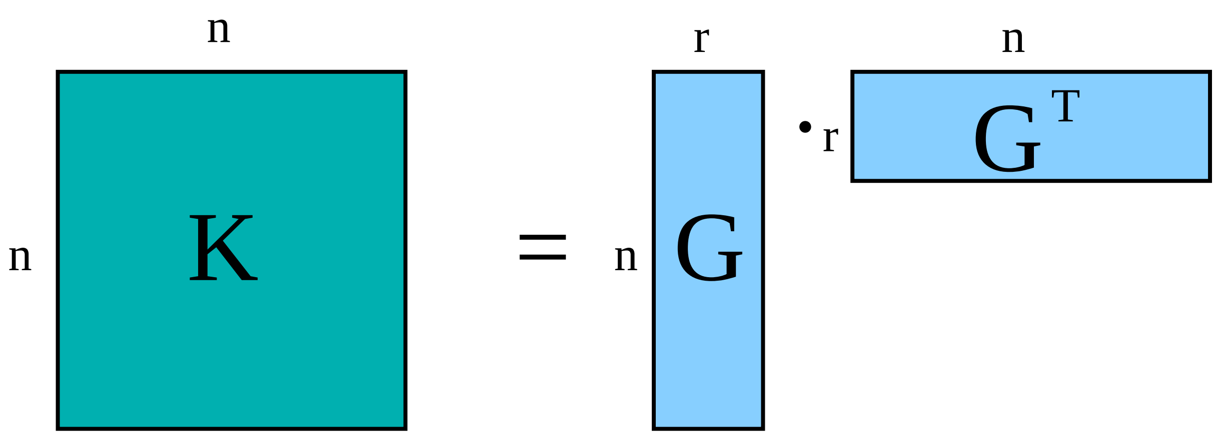
- Embed data into high-dimensional space via a mapping ψ
- Kernel function gives the inner product in the feature space:

$$\kappa(\mathbf{a}_i, \mathbf{a}_j) = \psi(\mathbf{a}_i) \cdot \psi(\mathbf{a}_j)$$

- Kernel algorithms use the kernel matrix for learning in feature space
- Kernels have been defined for various discrete structures
  - trees
  - graphs
  - strings
  - images
  - ...

Low-Rank Kernels

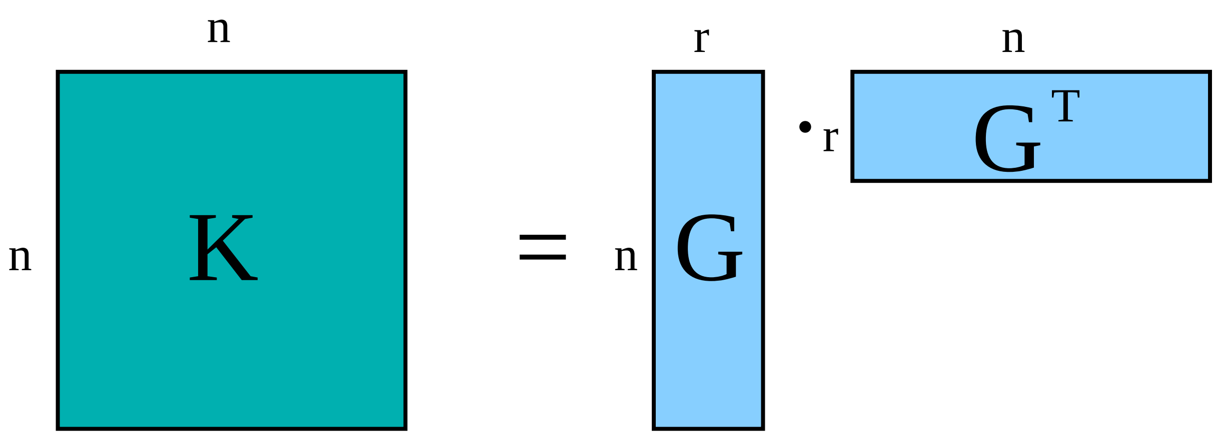
- A kernel matrix may not be of full rank, but rather of rank $r < n$



- Can *decompose* the kernel as $K = GG^T$

Low-Rank Kernels

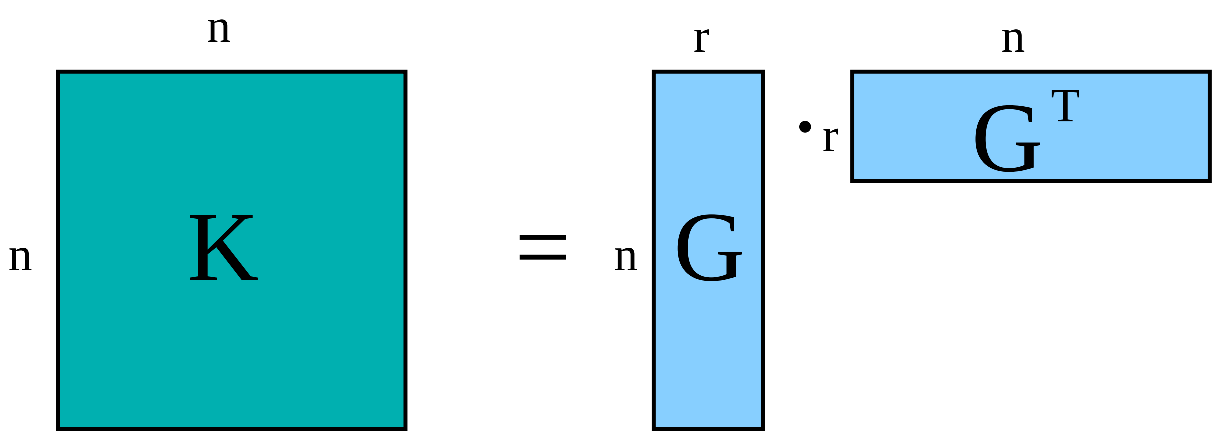
- A kernel matrix may not be of full rank, but rather of rank $r < n$



- Can *decompose* the kernel as $K = GG^T$
- Several advantages to such a decomposition
  - Storage reduces from $O(n^2)$ to $O(nr)$
  - Running time of most kernel algorithms improves to linear in n
  - Example: SVM training goes from $O(n^3)$ to $O(nr^2)$

Learning Low-Rank Kernels

- A kernel matrix may not be of full rank, but rather of rank $r < n$



- Sometimes need to *learn* the kernel matrix
  - No kernel may be given, but other information may be available
  - An existing kernel may be given, but may want to modify the kernel to satisfy additional constraints
  - May want to combine multiple sources of data into a single kernel
- Existing methods for learning a kernel are $O(n^3)$ (or worse)
  - Semi-definite programming methods [Lanckriet et al., JMLR, 2004]
  - Projection-based methods [Tsuda et al., JMLR, 2005]
  - Hyperkernels and other methods [Ong et al, JMLR, 2005]

Bregman Divergences

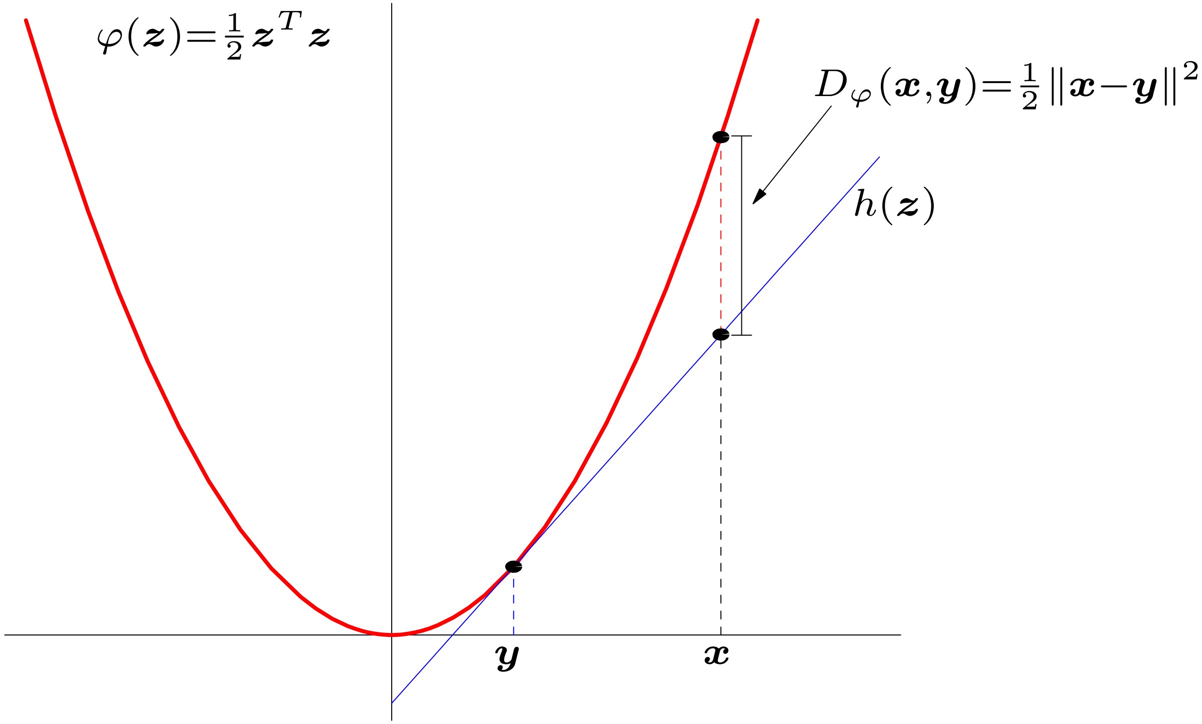
- Let $\varphi : S \to \mathbb{R}$ be a differentiable, strictly convex function of "Legendre type" ($S \subseteq \mathbb{R}^d$)
- The Bregman Divergence $D_\varphi : S \times \text{ri}(S) \to \mathbb{R}$ is defined as

$$D_\varphi(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{y}) - (\boldsymbol{x} - \boldsymbol{y})^T \nabla\varphi(\boldsymbol{y})$$

Bregman Divergences

- Let $\varphi : S \to \mathbb{R}$ be a differentiable, strictly convex function of "Legendre type" ($S \subseteq \mathbb{R}^d$)
- The Bregman Divergence $D_\varphi : S \times \mathrm{ri}(S) \to \mathbb{R}$ is defined as

$$D_\varphi(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{y}) - (\boldsymbol{x} - \boldsymbol{y})^T \nabla\varphi(\boldsymbol{y})$$

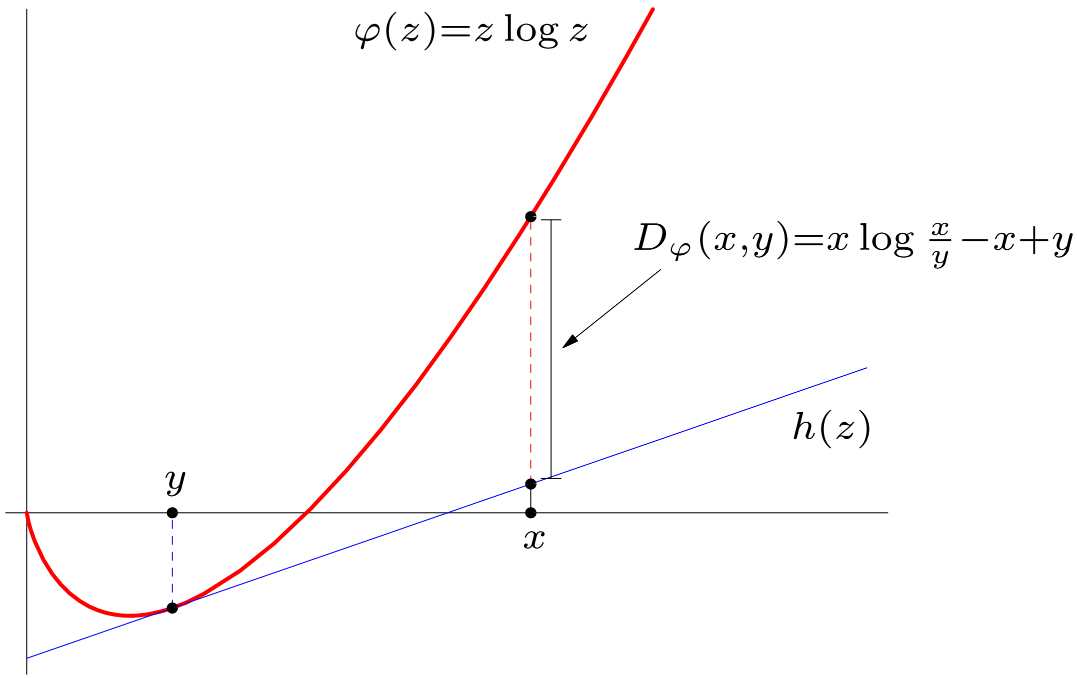


Squared Euclidean distance is a Bregman divergence

Bregman Divergences

- Let $\varphi : S \to \mathbb{R}$ be a differentiable, strictly convex function of "Legendre type" ($S \subseteq \mathbb{R}^d$)
- The Bregman Divergence $D_\varphi : S \times \text{ri}(S) \to \mathbb{R}$ is defined as

$$D_\varphi(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{y}) - (\boldsymbol{x} - \boldsymbol{y})^T \nabla\varphi(\boldsymbol{y})$$

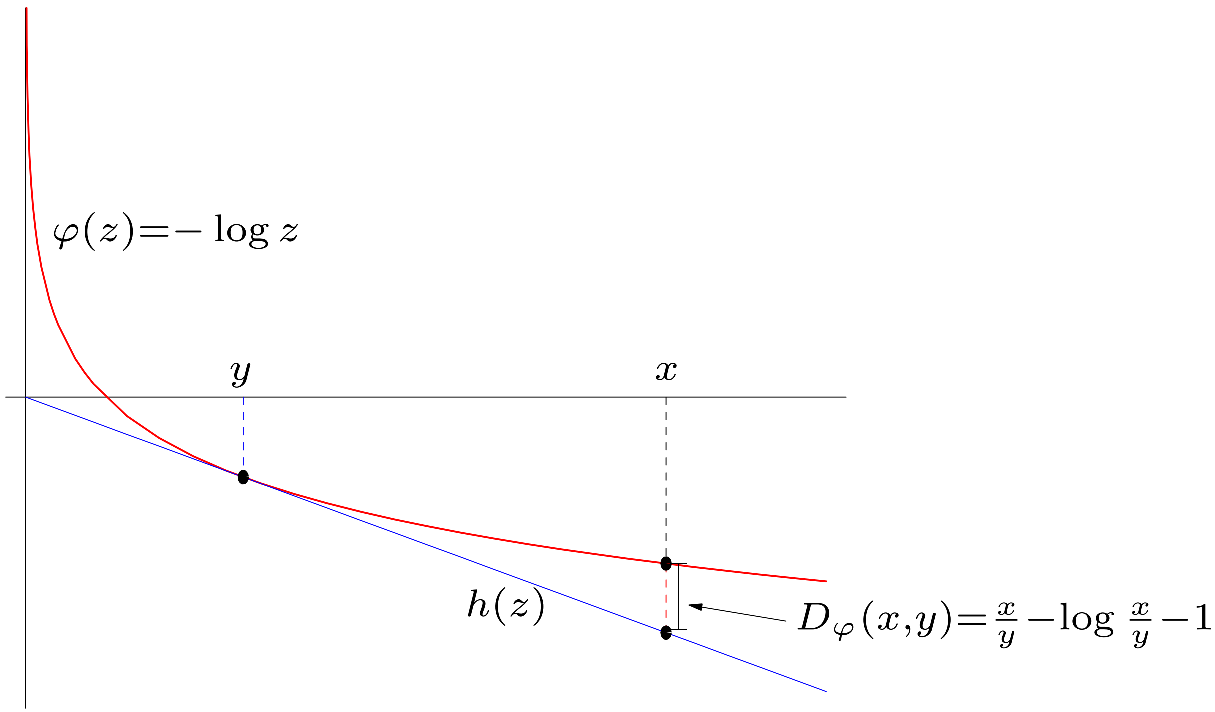


Relative Entropy (also called KL-divergence)

Bregman Divergences

- Let $\varphi : S \to \mathbb{R}$ be a differentiable, strictly convex function of "Legendre type" ($S \subseteq \mathbb{R}^d$)
- The Bregman Divergence $D_\varphi : S \times \text{ri}(S) \to \mathbb{R}$ is defined as

$$D_\varphi(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{y}) - (\boldsymbol{x} - \boldsymbol{y})^T \nabla\varphi(\boldsymbol{y})$$



Itakura-Saito Distance (or Burg divergence)—used in signal processing

Properties of Bregman Divergences

- $D_\varphi(\boldsymbol{x}, \boldsymbol{y}) \geq 0$, and equals 0 iff $\boldsymbol{x} = \boldsymbol{y}$
- Not a metric (symmetry, triangle inequality do not hold)
- Strictly convex in the first argument, but not convex (in general) in the second argument
- Three-point property generalizes the “Law of cosines”:

$$D_\varphi(\boldsymbol{x}, \boldsymbol{y}) = D_\varphi(\boldsymbol{x}, \boldsymbol{z}) + D_\varphi(\boldsymbol{z}, \boldsymbol{y}) - (\boldsymbol{x} - \boldsymbol{z})^T (\nabla\varphi(\boldsymbol{y}) - \nabla\varphi(\boldsymbol{z}))$$

Bregman Projections

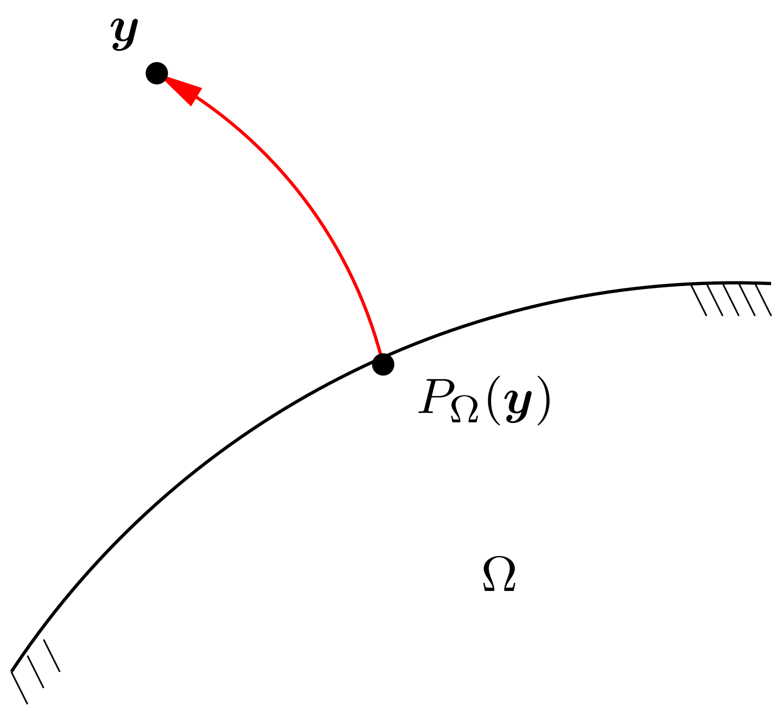
- Nearness in Bregman divergence: the "Bregman" projection of $\boldsymbol{y}$ onto a convex set Ω,

$$P_\Omega(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{\omega} \in \Omega} D_\varphi(\boldsymbol{\omega}, \boldsymbol{y})$$

Bregman Projections

- Nearness in Bregman divergence: the "Bregman" projection of $\boldsymbol{y}$ onto a convex set Ω,

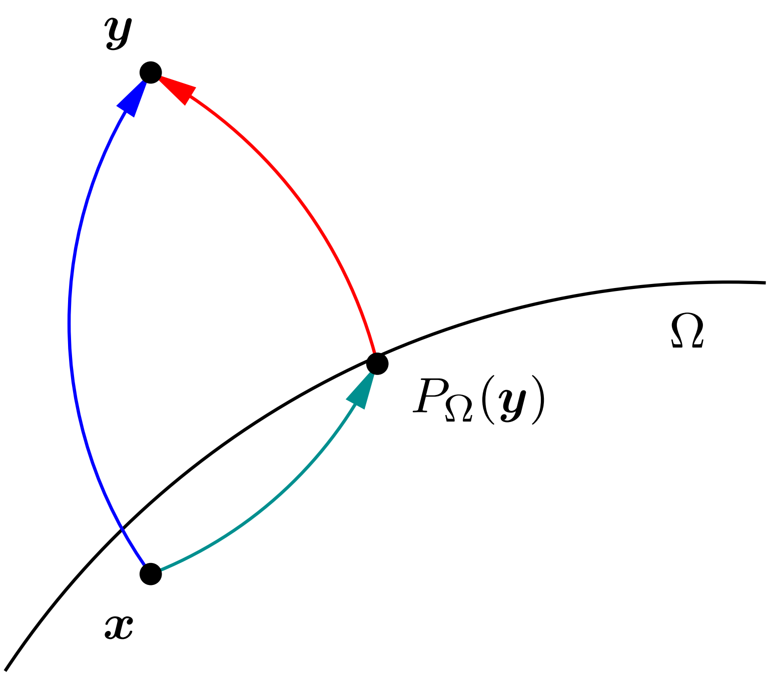
$$P_\Omega(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{\omega}\in\Omega} D_\varphi(\boldsymbol{\omega}, \boldsymbol{y})$$



Bregman Projections

- Nearness in Bregman divergence: the "Bregman" projection of $\boldsymbol{y}$ onto a convex set Ω,

$$P_\Omega(\boldsymbol{y}) = \underset{\boldsymbol{\omega}\in\Omega}{\operatorname{argmin}}\, D_\varphi(\boldsymbol{\omega}, \boldsymbol{y})$$



- Generalized Pythagoras Theorem:

$$D_\varphi(\boldsymbol{x}, \boldsymbol{y}) \geq D_\varphi(\boldsymbol{x}, P_\Omega(\boldsymbol{y})) + D_\varphi(P_\Omega(\boldsymbol{y}), \boldsymbol{y})$$

When Ω is an affine set, the above holds with equality

Bregman Matrix Divergences

- Extend Bregman divergences over vectors to divergences over real, symmetric matrices:

$$D_\varphi(X,Y) = \varphi(X) - \varphi(Y) - \mathrm{trace}((\nabla\varphi(Y))^T(X-Y))$$

- Squared Frobenius norm: $\varphi(X) = \frac{1}{2}\text{tr}(X^TX)$. Then

$$D_{Frob}(X,Y) = \frac{1}{2}\|X-Y\|_F^2$$

- von Neumann Divergence: $\varphi(X) = \text{tr}(X\log X - X)$ (negative entropy of the eigenvalues). Then

$$D_{vN}(X,Y) = \mathrm{trace}(X\log X - X\log Y - X + Y)$$

- Burg Matrix Divergence (LogDet divergence): $\varphi(X) = -\log\text{det}X$ (Burg entropy of the eigenvalues). Then

$$D_{Burg}(X,Y) = \mathrm{trace}(XY^{-1}) - \log\text{det}(XY^{-1}) - n$$

Optimization Framework

- Optimization problem:

$$\begin{array}{rl} \text{minimize} & D_\varphi(K, K_0) \\ \text{subject to} & \mathrm{tr}(KA_i) \leq b_i, \ 1 \leq i \leq c \\ & \mathrm{rank}(K) \leq r \\ & K \succeq 0. \end{array}$$

- Problem is non-convex due to the rank constraint
- Standard convex optimization techniques do not seem to apply here

Rank-Preserving Matrix Divergences

- Rewrite the Von Neumann divergence and Burg divergence using the eigendecompositions
- Given matrices X and Y, let $X = V\Lambda V^T$ and $Y = U\Theta U^T$
- von Neumann:

$$
\begin{aligned}
D_{vN}(X,Y) &= \text{tr}(X\log X - X\log Y - X + Y) \\
&= \sum_i \lambda_i \log\lambda_i - \sum_{i,j}(\mathbf{v}_i^T\mathbf{u}_j)^2\lambda_i\log\theta_j - \sum_i(\lambda_i - \theta_i)
\end{aligned}
$$

- Important Implication: $D_{vN}(X,Y)$ is finite iff range$(X) \subseteq$ range(Y)

Rank-Preserving Matrix Divergences

- Rewrite the Von Neumann divergence and Burg divergence using the eigendecompositions
- Given matrices X and Y, let $X = V\Lambda V^T$ and $Y = U\Theta U^T$
- Burg divergence:

$$\begin{aligned} D_{Burg}(X, Y) &= \mathrm{tr}(XY^{-1}) - \log \mathrm{det}(XY^{-1}) - n \\ &= \sum_{i,j} \frac{\lambda_i}{\theta_j} (\mathbf{v}_i^T \mathbf{u}_j)^2 - \sum_i \log\left(\frac{\lambda_i}{\theta_i}\right) - n \end{aligned}$$

- Important Implication: $D_{Burg}(X, Y)$ is finite iff range$(X) =$ range(Y)

Rank-Preserving Matrix Divergences

- Rewrite the Von Neumann divergence and Burg divergence using the eigendecompositions
- Given matrices X and Y, let $X = V\Lambda V^T$ and $Y = U\Theta U^T$
- Burg divergence:

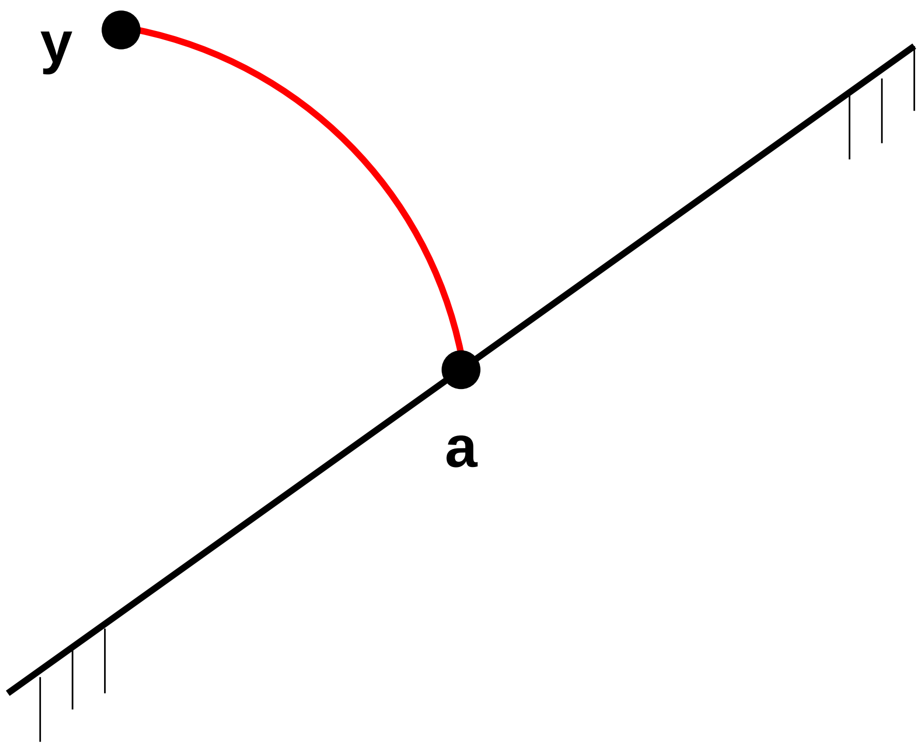
$$\begin{aligned} D_{Burg}(X, Y) &= \text{tr}(XY^{-1}) - \log \det(XY^{-1}) - n \\ &= \sum_{i,j} \frac{\lambda_i}{\theta_j} (\mathbf{v}_i^T \mathbf{u}_j)^2 - \sum_i \log\left(\frac{\lambda_i}{\theta_i}\right) - n \end{aligned}$$

- Important Implication: $D_{Burg}(X, Y)$ is finite iff range$(X) =$ range(Y)
- Optimization problem:
  - Satisfies rank$(K) \leq$ rank(K_0) in vN divergence
  - Satisfies rank$(K) =$ rank(K_0) in Burg divergence
  - Implicitly maintains rank and positive semi-definite constraints
  - Is convex when rank$(K_0) \leq r$

Method of Cyclic Projections

- Use the "Bregman" projection of $\boldsymbol{y}$ onto affine set $\mathcal{H}$,

$$P_{\mathcal{H}}(\boldsymbol{y}) = \underset{\boldsymbol{a}\in\mathcal{H}}{\operatorname{argmin}}\, D_{\varphi}(\boldsymbol{a}, \boldsymbol{y})$$



- Method projects onto each constraint, one at a time, and applies appropriate "correction"—provably converges to the optimal solution

Projections with rank-preserving Bregman divergences

- Solve smaller optimization problem to get projection onto constraint i

$$\begin{array}{rl} \text{minimize} & D_\varphi(K_{t+1}, K_t) \\ \text{subject to} & \text{tr}(K_{t+1}A_i) \le b_i \end{array}$$

- For both divergences, projections can be calculated in $O(r^2)$ time
- Requires all operations to be done on a suitable factored form of the kernel matrices

Projections with rank-preserving Bregman divergences

- Solve smaller optimization problem to get projection onto constraint i

$$\begin{array}{rl} \text{minimize} & D_\varphi(K_{t+1}, K_t) \\ \text{subject to} & \mathrm{tr}(K_{t+1}A_i) \leq b_i \end{array}$$

- For both divergences, projections can be calculated in $O(r^2)$ time
- Requires all operations to be done on a suitable factored form of the kernel matrices
- Burg divergence update:

$$\begin{array}{rcl} K_{t+1} & = & (K_t^\dagger - \alpha A_i^T)^\dagger \\ \text{such that} \quad \mathrm{tr}(K_{t+1}A_i) & \leq & b_i \end{array}$$

  - Distance constraints expressed as $A_i = \boldsymbol{z}_i \boldsymbol{z}_i^T$
  - Projection parameter α *has a closed form solution*
  - $O(r^2)$ projection achieved using factored form of K_t

Projections with rank-preserving Bregman divergences

- Solve smaller optimization problem to get projection onto constraint i

$$\begin{array}{rl} \text{minimize} & D_\varphi(K_{t+1}, K_t) \\ \text{subject to} & \text{tr}(K_{t+1}A_i) \le b_i \end{array}$$

- For both divergences, projections can be calculated in $O(r^2)$ time
- Requires all operations to be done on a suitable factored form of the kernel matrices
- Von Neumann divergence update:

$$\begin{array}{rrcl} & K_{t+1} & = & \exp(\log(K_t) + \alpha \boldsymbol{z}_i \boldsymbol{z}_i^T) \\ \text{such that} & \text{tr}(K_{t+1}A_i) & \le & b_i \end{array}$$

  - Requires the fast multipole method to achieve $O(r^2)$ projection
  - We have designed a custom non-linear solver to compute the projection parameter α

Projections with rank-preserving Bregman divergences

- Solve smaller optimization problem to get projection onto constraint i

$$\begin{array}{rl} \text{minimize} & D_\varphi(K_{t+1}, K_t) \\ \text{subject to} & \mathrm{tr}(K_{t+1}A_i) \leq b_i \end{array}$$

- For both divergences, projections can be calculated in $O(r^2)$ time
- Requires all operations to be done on a suitable factored form of the kernel matrices
- Von Neumann divergence update:

$$\begin{array}{rrcl} & K_{t+1} & = & \exp(\log(K_t) + \alpha \boldsymbol{z}_i \boldsymbol{z}_i^T) \\ \text{such that} & \mathrm{tr}(K_{t+1}A_i) & \leq & b_i \end{array}$$

  - Requires the fast multipole method to achieve $O(r^2)$ projection
  - We have designed a custom non-linear solver to compute the projection parameter α
- End Result: Algorithms are linear in $n + c$, quadratic in r

Special Cases and Related Work

- Full-rank case with von Neumann divergence, $b_i = 0 \forall i$, obtain DefiniteBoost optimization problem [Tsuda et al., JMLR, 2005]
  - Improve the running time from $O(n^3)$ to $O(n^2)$ per projection
  - Allow arbitrary linear constraints, both equality and inequality

- For constraints $K_{ii} = 1 \forall i$, obtain the nearest correlation matrix problem [Higham, IMA J. Numerical Analysis, 2002]
  - Arises in financial applications
  - New efficient methods for finding low-rank correlation matrices

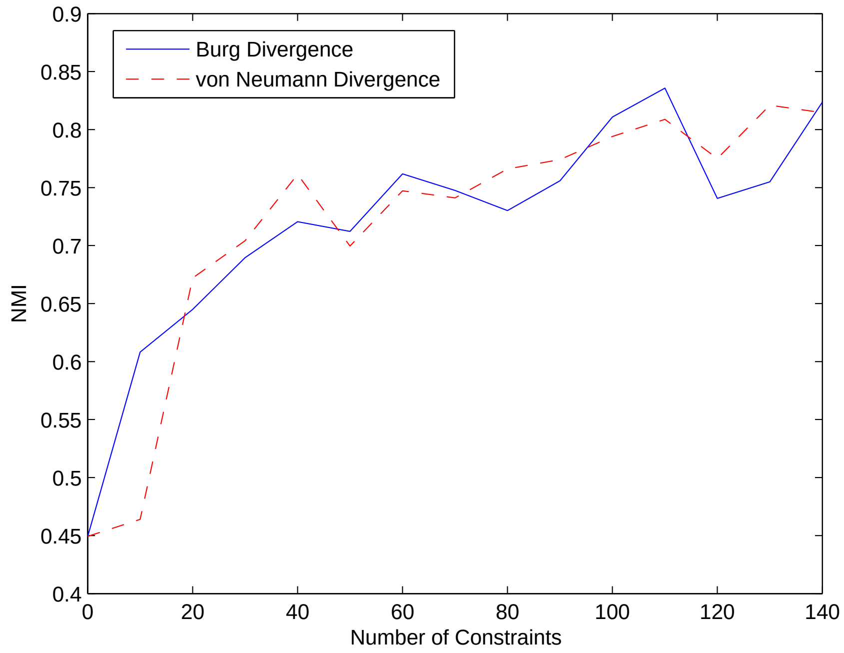
- Can be used for nonlinear dimensionality reduction
  - Isometry constraints are linear
  - [Weinberger et al., ICML, 2004] maximize trace(X) (by semi-definite programming)

Experiments

- Digits data: 317 digits, 3 classes
  - Given a rank-16 kernel for 317 digits
  - Randomly create constraints:

$$d(i_1, i_2) \leq (1 - \epsilon) b_i$$
$$d(i_1, i_2) \geq (1 + \epsilon) b_i$$

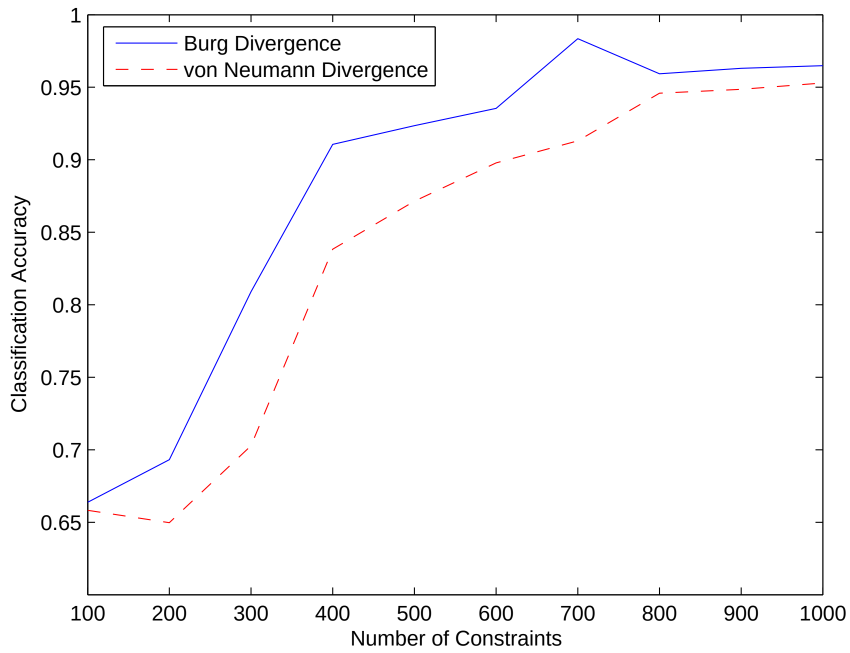
  - Attempt to learn a "better" rank-16 kernel



- Clustering: use kernel k-means with random initialization, compute accuracy using normalized mutual information

Experiments

- GyrB protein data: 52 proteins, 3 classes
  - Given *only* constraints
  - Want to learn a kernel based on constraints
  - Constraints generated from target kernel matrix
  - Attempt to learn a full-rank kernel



- Classification: use k-nearest neighbor, $k = 5$, 50/50 training/test split, 2-fold cross validation averaged over 20 runs

Relevant Papers

- [Dhillon & Tropp] "Matrix Nearness Problems with Bregman Divergences", submitted to the SIAM Journal on Matrix Analysis and Applications, 2006.
- [Kulis, Sustik & Dhillon] "Low-Rank Kernel Learning", International Conference on Machine Learning, 2006 (to appear).

Conclusions

- Low-rank kernel matrices can be learnt efficiently using appropriate Bregman matrix divergences
  - Burg divergence and von Neumann divergence
  - Implicitly maintain rank and positive semi-definiteness constraints
- Algorithms scale linearly with n and quadratically with r
- Future Work
  - Further investigate the gains of preserving range space
  - Apply to varied applications
  - Improvement over cyclic projection methods