Spectral Measures for Nearness Problems

Inderjit S. Dhillon

The University of Texas at Austin

6th International Workshop on Accurate Solution of Eigenvalue Problems

Pennsylvania State University, University Park

May 22, 2006

Joint work with Brian Kulis and Mátyás Sustik

Nearness Problems

- Given an input matrix, find the “nearest” matrix that satisfies user constraints
- How should nearness be measured?
- Typical choices are the Frobenius norm or the spectral 2-norm
- However, these may not be appropriate for the application at hand
- Outline of talk
  - Bregman vector divergences
  - Bregman matrix divergences — offer alternate spectral measures
  - Nearness problems with von Neumann & Burg matrix divergences

Bregman Divergences

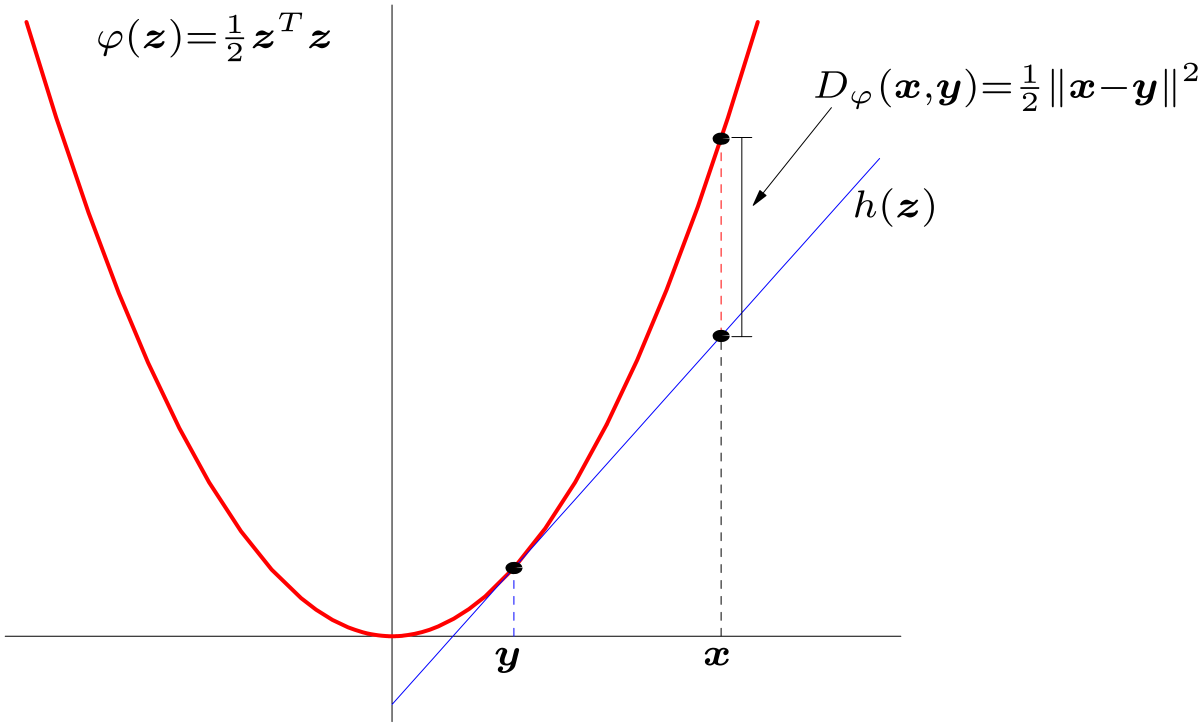
- Let $\varphi : S \to \mathbb{R}$ be a differentiable, strictly convex function of "Legendre type" ($S \subseteq \mathbb{R}^d$)
- The Bregman Divergence $D_\varphi : S \times \text{relint}(S) \to \mathbb{R}$ is defined as

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

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 $D_\varphi : S \times \text{relint}(S) \to \mathbb{R}$ is defined as

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

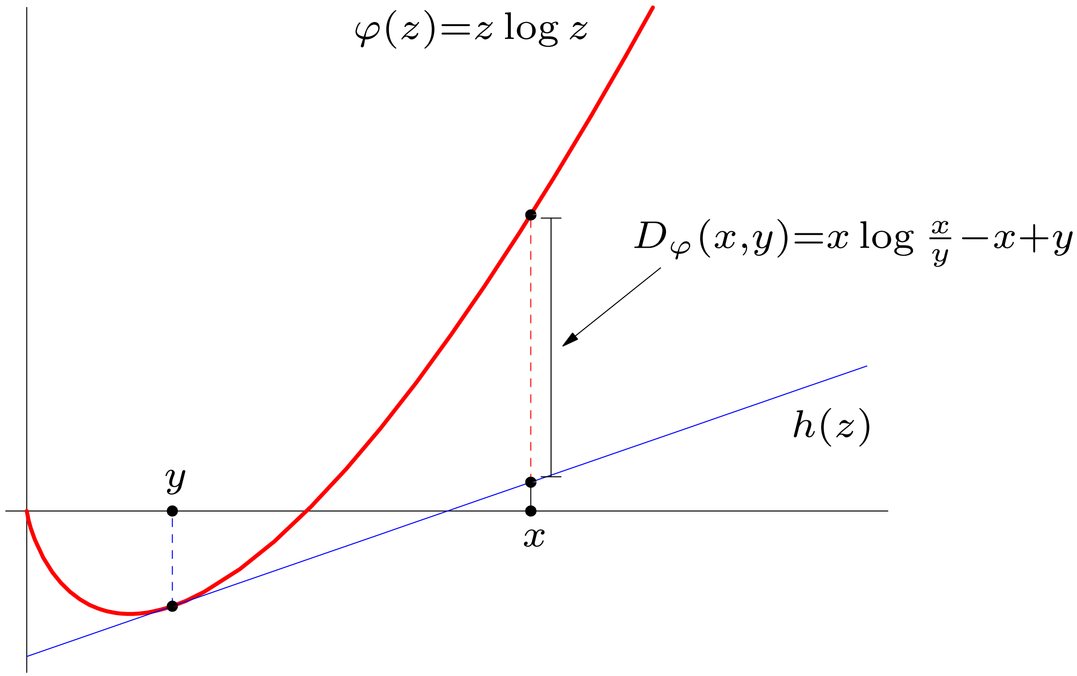


Squared Euclidean distance is a Bregman divergence

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 $D_\varphi : S \times \text{relint}(S) \to \mathbb{R}$ is defined as

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

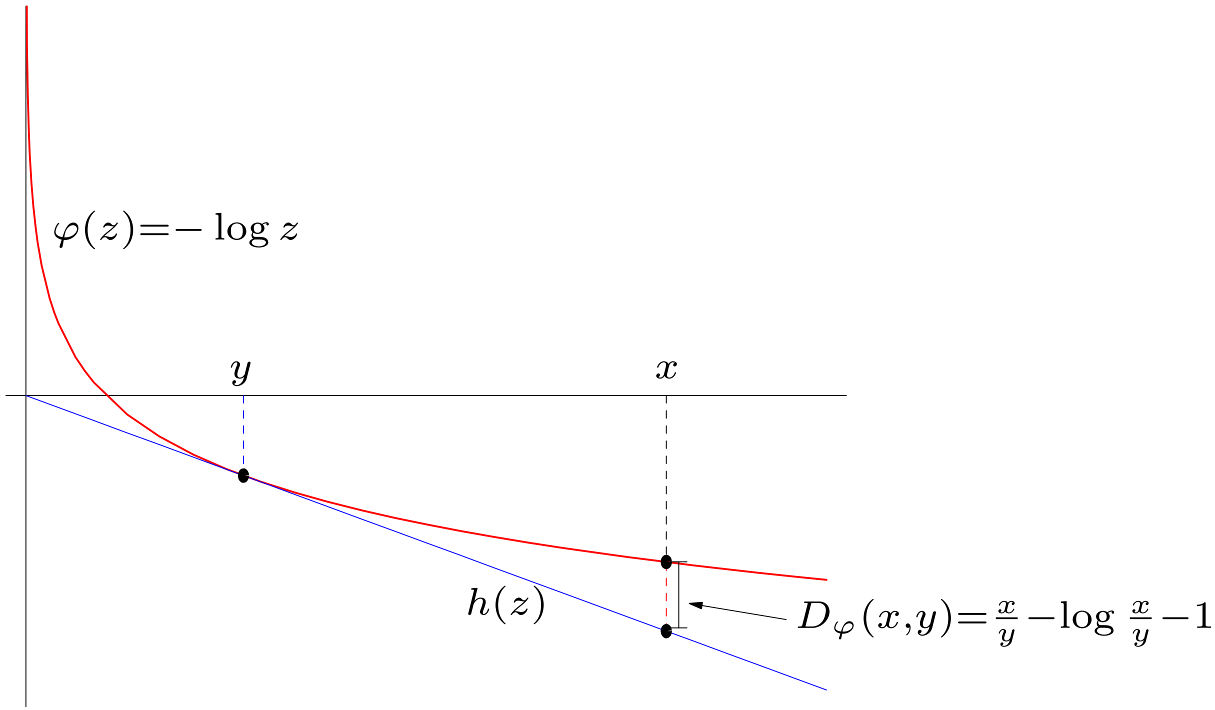


Relative Entropy (also called KL-divergence) is another Bregman divergence

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 $D_\varphi : S \times \text{relint}(S) \to \mathbb{R}$ 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 (used in signal processing) is another Bregman divergence

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

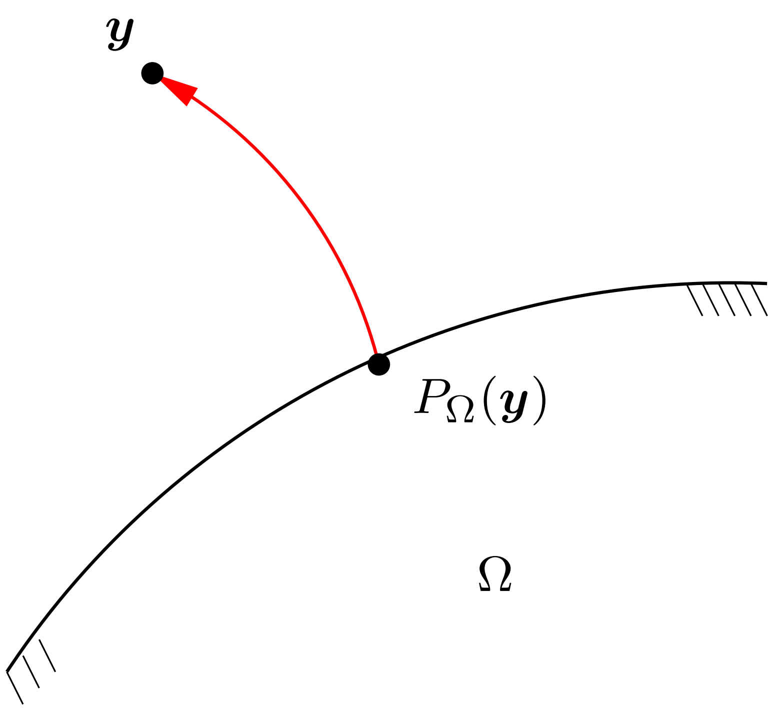
- Nearness in Bregman divergence: the "Bregman" projection of $\boldsymbol{y}$ onto a convex set Ω,

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

Bregman Projections

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

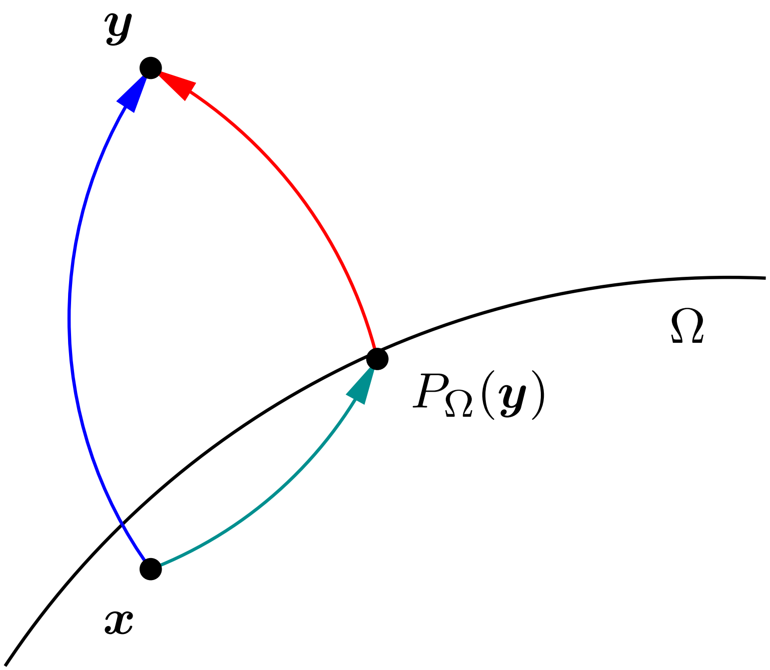
$$P_\Omega(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{\omega}\in\Omega} D_\varphi(\boldsymbol{\omega}, \boldsymbol{y})$$



Bregman Projections

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

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



- 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

- Generalizes the notion of divergence to matrices
- Let φ be a real-valued convex function over matrices
- Leads to Bregman matrix divergences:

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

- For example, $\varphi(\boldsymbol{X}) = \|\boldsymbol{X}\|_F^2$ leads to

$$D_\varphi(\boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2$$

Squared Euclidean Distance	$\longleftrightarrow$	Squared Frobenius Distance
Relative Entropy	$\longleftrightarrow$	von Neumann Divergence (Quantum Relative Entropy)
Itakura-Saito Divergence	$\longleftrightarrow$	Burg Divergence (LogDet Divergence)

Von Neumann Matrix Divergence

- Let $\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ be a positive definite matrix
- Consider negative entropy of the eigenvalues (von Neumann entropy):

$$\varphi(\boldsymbol{X}) = \sum_i (\lambda_i \log \lambda_i - \lambda_i) = \mathrm{tr}(\boldsymbol{X} \log \boldsymbol{X} - \boldsymbol{X})$$

- Yields the von Neumann matrix divergence (quantum relative entropy):

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

- In terms of the spectrum of $\boldsymbol{X}$ and $\boldsymbol{Y}$ ($\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$, $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Theta}\boldsymbol{U}^T$):

$$D_{vN}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_i \lambda_i \log \lambda_i - \sum_i \sum_j (\boldsymbol{v}_i^T \boldsymbol{u}_j)^2 \lambda_i \log \theta_j - \sum_i (\lambda_i - \theta_i)$$

- Definition can be extended to semi-definite matrices
- Divergence is finite iff $\mathrm{range}(\boldsymbol{X}) \subseteq \mathrm{range}(\boldsymbol{Y})$

Burg Matrix Divergence

- Let $\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ be an $N \times N$ positive definite matrix
- Consider Burg entropy of the eigenvalues:

$$\varphi(\boldsymbol{X}) = \sum_i \log \lambda_i = \log\det \boldsymbol{X}$$

- Yields the Burg (or LogDet) matrix divergence:

$$D_{Burg}(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{tr}(\boldsymbol{X}\boldsymbol{Y}^{-1}) - \log\det(\boldsymbol{X}\boldsymbol{Y}^{-1}) - N$$

- In terms of the spectrum of $\boldsymbol{X}$ and $\boldsymbol{Y}$ ($\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$, $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Theta}\boldsymbol{U}^T$):

$$D_{Burg}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_i \sum_j \frac{\lambda_i}{\theta_j}(\boldsymbol{v}_i^T \boldsymbol{u}_j)^2 - \sum_i \log \frac{\lambda_i}{\theta_i} - N$$

- Definition can be extended to semi-definite matrices
- Divergence is finite iff $\mathrm{range}(\boldsymbol{X}) = \mathrm{range}(\boldsymbol{Y})$

Nearness Problem with Matrix Divergences

- Nearness with respect to linear constraints

$$\begin{array}{rl} \min\limits_{\boldsymbol{X} \in S} & d(\boldsymbol{X}, \boldsymbol{X}_0) \\ \text{subject to} & \mathrm{tr}(\boldsymbol{X}\boldsymbol{A}_i) \leq b_i \end{array}$$

Nearness Problem with Matrix Divergences

- Nearness with respect to linear constraints

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & D_{\varphi}(\boldsymbol{X}, \boldsymbol{X}_0) \\ \text{subject to} \quad & \operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_i) \le b_i \\ & \boldsymbol{X} \succeq 0 \end{aligned}$$

Nearness Problem with Matrix Divergences

- Nearness with respect to linear constraints

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & D_{\varphi}(\boldsymbol{X}, \boldsymbol{X}_0) \\ \text{subject to} \quad & \operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_i) \leq b_i \\ & \boldsymbol{X} \succeq 0 \end{aligned}$$

- Arises in various applications:
  - Nearest correlation matrix (Higham, 2002)
  - Kernel learning (Tsuda et al, 2004; Kulis et al 2006)
  - . . .

Rank-Constrained Nearness Problem

- Nearness with respect to linear and rank constraints

$$\begin{array}{rl} \min\limits_{\boldsymbol{X}} & D_{\varphi}(\boldsymbol{X}, \boldsymbol{X}_0) \\ \text{subject to} & \operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_i) \leq b_i \\ & \boldsymbol{X} \succeq 0 \\ & \operatorname{rank}(\boldsymbol{X}) \leq r \end{array}$$

Rank-Constrained Nearness Problem

- Nearness with respect to linear and rank constraints

$$\begin{array}{rl} \min\limits_{\boldsymbol{X}} & D_\varphi(\boldsymbol{X}, \boldsymbol{X}_0) \\ \text{subject to} & \operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_i) \le b_i \\ & \boldsymbol{X} \succeq 0 \\ & \operatorname{rank}(\boldsymbol{X}) \le r \end{array}$$

- In general, the above problem is non-convex

Rank-Constrained Nearness Problem

- Nearness with respect to linear and rank constraints

$$\begin{array}{rl} \min\limits_{\boldsymbol{X}} & D_{\varphi}(\boldsymbol{X}, \boldsymbol{X}_0) \\ \text{subject to} & \operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_i) \leq b_i \\ & \boldsymbol{X} \succeq 0 \\ & \operatorname{rank}(\boldsymbol{X}) \leq r \end{array}$$

- In general, the above problem is non-convex
- Turns out to be convex if:
  - $\operatorname{rank}(\boldsymbol{X}_0) \leq r$, and
  - D_{φ} is the von Neumann or Burg divergence

Rank-Constrained Nearness Problem

- Nearness with respect to linear and rank constraints

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & D_\varphi(\boldsymbol{X}, \boldsymbol{X}_0) \\ \text{subject to} \quad & \operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_i) \leq b_i \\ & \boldsymbol{X} \succeq 0 \\ & \operatorname{rank}(\boldsymbol{X}) \leq r \end{aligned}$$

- In general, the above problem is non-convex
- Turns out to be convex if:
  - $\operatorname{rank}(\boldsymbol{X}_0) \leq r$, and
  - D_φ is the von Neumann or Burg divergence
- Thus, in this case, the last two constraints can be “dropped”:

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & D_\varphi(\boldsymbol{X}, \boldsymbol{X}_0) \\ \text{subject to} \quad & \operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_i) \leq b_i \end{aligned}$$

Method of cyclic projections

- Consider the convex optimization problem:

$$\begin{array}{ll} \min_{\boldsymbol{x}} & \varphi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{a}_i^T \boldsymbol{x} = b_i, \quad i = 0, \ldots, m-1 \end{array}$$

- Bregman's cyclic projection method:
  1. Start with $\boldsymbol{x}^0$ that satisfies $\nabla\varphi(\boldsymbol{x}^0) = -A^T \boldsymbol{\pi}$. Set $t = 0$.
  2. Let $j = t \mod m$. Compute $\boldsymbol{x}^{t+1}$ to be the Bregman projection of $\boldsymbol{x}^t$ onto the j-th hyperplane, i.e., $\boldsymbol{x}^{t+1}$ is the solution of

$$\begin{array}{ll} \min_{\boldsymbol{x}} & D_\varphi(\boldsymbol{x}, \boldsymbol{x}^t) \\ \text{subject to} & \boldsymbol{a}_j^T \boldsymbol{x} = b_j \end{array}$$

  3. Set $t = t + 1$ and repeat.
- Converges to globally optimal solution (Bregman, 1967)
- Can be extended to halfspace and convex constraints — each projection needs to be followed by a correction

Cyclic Projection Step

- At step t of the cyclic projection algorithm, we need to solve:

$$\begin{array}{rl} \min_{\boldsymbol{X}} & D_{\varphi}(\boldsymbol{X}, \boldsymbol{X}_t) \\ \text{subject to} & \operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_i) = b_i \end{array}$$

- Lagrange dual:

$$L(\boldsymbol{X}, \alpha) = \min_{\boldsymbol{X}} \; D_{\varphi}(\boldsymbol{X}, \boldsymbol{X}_t) + \alpha(\operatorname{tr}(\boldsymbol{X}\boldsymbol{A}_{\boldsymbol{i}}) - b_i)$$

- Need to solve for X_{t+1} and α:

$$\begin{aligned} \nabla\varphi(\boldsymbol{X}_{t+1}) &= \nabla\varphi(\boldsymbol{X}_t) + \alpha\boldsymbol{A}_i \\ \operatorname{tr}(\boldsymbol{X}_{t+1}\boldsymbol{A}_i) &= b_i \end{aligned}$$

Burg Update

- Burg Divergence

$$D_{Burg}(\boldsymbol{X}, \boldsymbol{X}_t) = \mathrm{tr}(\boldsymbol{X}\boldsymbol{X}_t^{-1}) - \log\det(\boldsymbol{X}\boldsymbol{X}_t^{-1}) - N$$

- Gradient is

$$\nabla D_{Burg}(\boldsymbol{X}, \boldsymbol{X}_t) = -\boldsymbol{X}^{-1} + \boldsymbol{X}_t^{-1}$$

- The Burg projection update becomes:

$$\begin{aligned} \nabla\varphi(\boldsymbol{X}_{t+1}) &= \nabla\varphi(\boldsymbol{X}_t) + \alpha\boldsymbol{A}_i \\ \Longrightarrow \qquad \boldsymbol{X}_{t+1} &= (\boldsymbol{X}_t^{-1} - \alpha\boldsymbol{A}_i)^{-1} \end{aligned}$$

- The update is often rank-one, $\boldsymbol{A}_i = \boldsymbol{z}_i\boldsymbol{z}_i^T$
  - Correlation matrix
  - Distance constraints in kernel learning

Projection Parameter—Burg Divergence

- Burg update:

$$\begin{aligned} \boldsymbol{X}_{t+1} &= (\boldsymbol{X}_t^{-1} - \alpha \boldsymbol{z}\boldsymbol{z}^T)^{-1} \\ \boldsymbol{z}^T \boldsymbol{X}_{t+1} \boldsymbol{z} &= b \end{aligned}$$

- A closed form solution exists!
- Sherman-Morrison-Woodbury formula leads to:

$$\begin{aligned} p &= \boldsymbol{z}^T \boldsymbol{X}_t \boldsymbol{z} \\ \alpha &= \frac{1}{p} - \frac{1}{b} \\ \beta &= \alpha/(1 - \alpha p) \\ \boldsymbol{X}_{t+1} &= \boldsymbol{X}_t + \beta \boldsymbol{X}_t \boldsymbol{z}\boldsymbol{z}^T \boldsymbol{X}_t \end{aligned}$$

- Allows extension to the rank-deficient case

Burg Update—Efficiency

- Burg update:

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \beta \boldsymbol{X}_t \boldsymbol{z} \boldsymbol{z}^T \boldsymbol{X}_t$$

- Using $\boldsymbol{X}_t = \boldsymbol{G}_t \boldsymbol{G}_t^T$, the Cholesky factor $\boldsymbol{G}_t$ needs to be updated:

$$\begin{aligned} I + \beta(\boldsymbol{G}_t^T \boldsymbol{z})(\boldsymbol{G}_t^T \boldsymbol{z})^T &= \boldsymbol{L}\boldsymbol{L}^T \\ \boldsymbol{G}_{t+1} &= \boldsymbol{G}_t \boldsymbol{L} \end{aligned}$$

- Note that $I + \beta(\boldsymbol{G}_t^T \boldsymbol{z})(\boldsymbol{G}_t^T \boldsymbol{z})^T$ is an $r \times r$ matrix
- Multiplication with $\boldsymbol{L}$ appears to be the most expensive operation
- Special structure of $\boldsymbol{L}$ allows an $O(r^2)$ algorithm

Burg Update using Eigendecomposition

- Burg update:

$$\boldsymbol{X}_{t+1} = (\boldsymbol{X}_t^{-1} - \alpha \boldsymbol{z}\boldsymbol{z}^T)^{-1}$$

- Maintain alternate factored form: $\boldsymbol{X}_t = \boldsymbol{V}_t \boldsymbol{\Lambda}_t \boldsymbol{V}_t^T$

$$\begin{aligned} \boldsymbol{X}_{t+1} &= (\boldsymbol{V}_t \boldsymbol{\Lambda}_t^{-1} \boldsymbol{V}_t^T - \alpha \boldsymbol{z}\boldsymbol{z}^T)^{-1} \\ &= \boldsymbol{V}_t (\boldsymbol{\Lambda}_t^{-1} - \alpha \boldsymbol{V}_t^T \boldsymbol{z}\boldsymbol{z}^T \boldsymbol{V}_t)^{-1} \boldsymbol{V}_t^T \end{aligned}$$

- Eigenvalue problem for a diagonal plus rank-one matrix:

$$\boldsymbol{\Lambda}_t^{-1} - \alpha (\boldsymbol{V}_t^T \boldsymbol{z})(\boldsymbol{V}_t^T \boldsymbol{z})^T = \boldsymbol{U} \boldsymbol{\Theta} \boldsymbol{U}^T$$

- Update the factored form:

$$\boldsymbol{V}_{t+1} = \boldsymbol{V}_t \boldsymbol{U},\ \boldsymbol{\Lambda}_{t+1} = \boldsymbol{\Theta}^{-1}$$

Von Neumann Update

- Von Neumann divergence:

$$D_{vN}(\boldsymbol{X}, \boldsymbol{X}_t) = \mathrm{tr}(\boldsymbol{X} \log \boldsymbol{X} - \boldsymbol{X} \log \boldsymbol{X}_t - \boldsymbol{X} + \boldsymbol{X}_t)$$

- Gradient is:

$$\nabla D_{vN}(\boldsymbol{X}, \boldsymbol{X}_t) = \log \boldsymbol{X} - \log \boldsymbol{X}_t$$

- The von Neumann projection update becomes:

$$\begin{aligned} \nabla\varphi(\boldsymbol{X}_{t+1}) &= \nabla\varphi(\boldsymbol{X}_t) + \alpha \boldsymbol{A}_i \\ \Longrightarrow \qquad \boldsymbol{X}_{t+1} &= \exp(\log(\boldsymbol{X}_t) + \alpha \boldsymbol{A}_i) \end{aligned}$$

- For rank-one updates: $\boldsymbol{A}_i = \boldsymbol{z}_i \boldsymbol{z}_i^T$

Von Neumann Update

- Von Neumann Update:

$$\begin{aligned} \boldsymbol{X}_{t+1} &= \exp(\log(\boldsymbol{X}_t) + \alpha \boldsymbol{z}\boldsymbol{z}^T) \\ \boldsymbol{z}^T \boldsymbol{X}_{t+1} \boldsymbol{z} &= b \end{aligned}$$

- Maintain factored form for efficiency: $\boldsymbol{X}_t = \boldsymbol{V}_t \boldsymbol{\Lambda}_t \boldsymbol{V}_t^T$

$$\begin{aligned} \boldsymbol{X}_{t+1} &= \exp(\boldsymbol{V}_t \log(\boldsymbol{\Lambda}_t) \boldsymbol{V}_t^T + \alpha \boldsymbol{z}\boldsymbol{z}^T) \\ &= \boldsymbol{V}_t \exp(\log(\boldsymbol{\Lambda}_t) + \alpha \boldsymbol{V}_t^T \boldsymbol{z}\boldsymbol{z}^T \boldsymbol{V}_t) \boldsymbol{V}_t^T \end{aligned}$$

- Eigenvalue problem for a diagonal plus rank-one matrix:

$$\log(\boldsymbol{\Lambda}_t) + \alpha(\boldsymbol{V}_t^T \boldsymbol{z})(\boldsymbol{V}_t^T \boldsymbol{z})^T = \boldsymbol{U}\boldsymbol{\Theta}\boldsymbol{U}^T$$

- Update in factored form:

$$\boldsymbol{V}_{t+1} = \boldsymbol{V}_t \boldsymbol{U}, \ \boldsymbol{\Lambda}_{t+1} = \exp(\boldsymbol{\Theta})$$

Von Neumann Update—Efficiency

- Von Neumann update in factored form:

$$\log(\boldsymbol{\Lambda}_t) + \alpha(\boldsymbol{V}_t^T\boldsymbol{z})(\boldsymbol{V}_t^T\boldsymbol{z})^T = \boldsymbol{U}\boldsymbol{\Theta}\boldsymbol{U}^T$$

$$\boldsymbol{V}_{t+1} = \boldsymbol{V}_t\boldsymbol{U},\ \ \boldsymbol{\Lambda}_{t+1} = \exp(\boldsymbol{\Theta})$$

- Note that $\log(\boldsymbol{\Lambda}_t) + \alpha(\boldsymbol{V}_t^T\boldsymbol{z})(\boldsymbol{V}_t^T\boldsymbol{z})^T$ is an $r \times r$ matrix
- The most expensive operation appears to be the $\boldsymbol{V}_t\boldsymbol{U}$ multiplication
- The Fast Multipole Method can exploit the structure of U (Greengard & Rokhlin, 1987)
- The multiplication can be performed in $O(r^2)$ time

Projection Parameter—Von Neumann Update

- Von Neumann Update:

$$\begin{aligned} \boldsymbol{X}_{t+1} &= \exp(\log(\boldsymbol{X}_t) + \alpha \boldsymbol{z}\boldsymbol{z}^T) \\ \boldsymbol{z}^T \boldsymbol{X}_{t+1} \boldsymbol{z} &= b \end{aligned}$$

- Set $\boldsymbol{w} = \boldsymbol{V}_t^T \boldsymbol{z}$. We need to solve the following for α:

$$\boldsymbol{w}^T \exp(\log(\boldsymbol{\Lambda}_t) + \alpha \boldsymbol{w}\boldsymbol{w}^T)\boldsymbol{w} = b$$

- The left hand side is a monotone function of α
- Ordinary bisection converges linearly (>50 iterations)
- Custom non-linear solver rarely needs more than 6 evaluations

Projection Parameter—Von Neumann Divergence

- We exploit the fact that

$$g(\alpha) = \boldsymbol{w}^T \exp(\log(\boldsymbol{\Lambda}_t) + \alpha \boldsymbol{w}\boldsymbol{w}^T)\boldsymbol{w} - b$$

  is similar to an exponential function
- Like Newton's method, but fit exponentials instead of straight lines
- Set $\alpha_0 = 0$, $\alpha_1 = 1$. At the i-th step let $g_1(\alpha) = \exp(p\alpha + q) - b$ such that:

$$g_1(\alpha_{i-1}) = g(\alpha_{i-1}), \;\; g_1(\alpha_i) = g(\alpha_i)$$

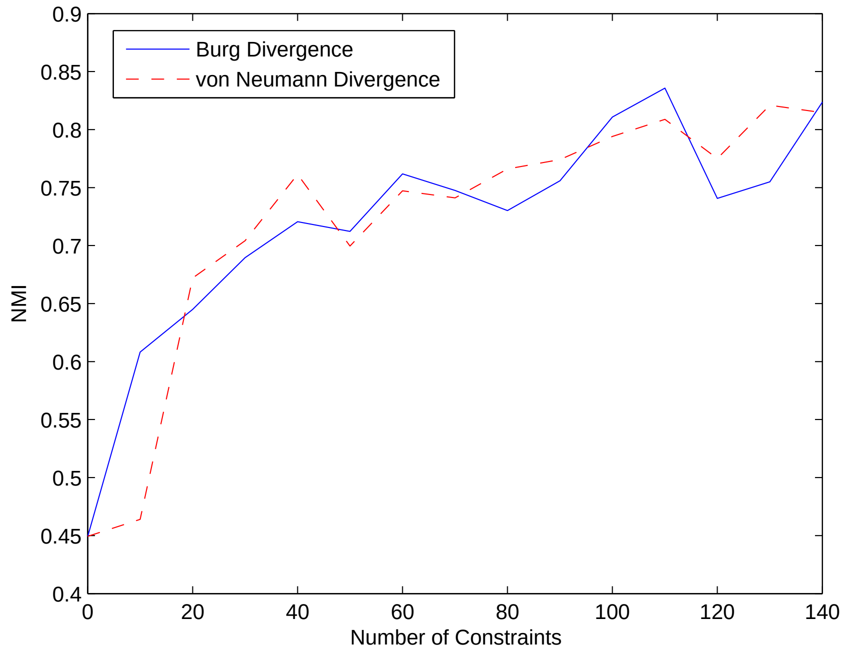
- Set α_{i+1} to be the solution of $g_1(\alpha) = 0$.
- Each iteration involves the solution of a secular equation

Experiments

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

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

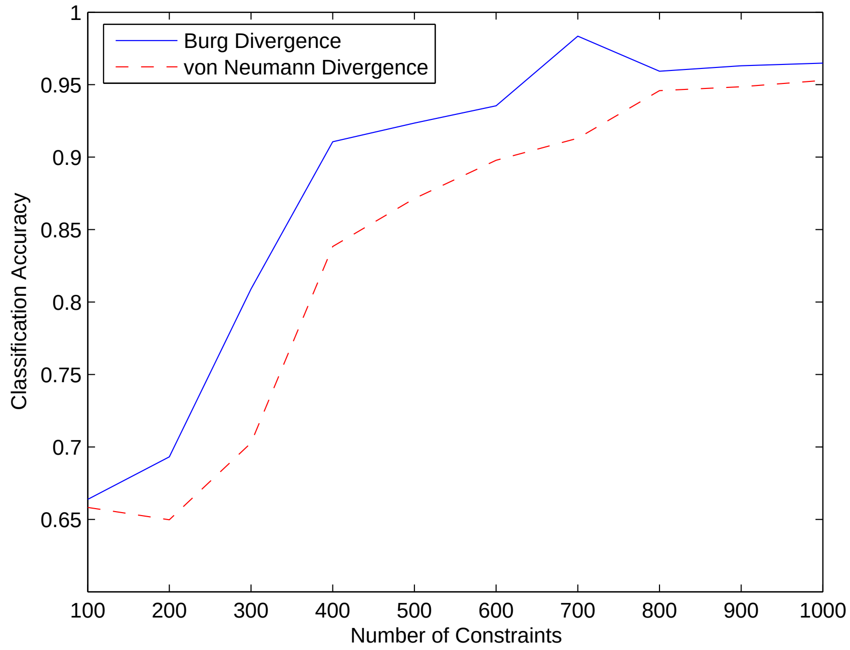
  - Attempt to learn a "better" rank-16 kernel



- 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

Conclusions & Future Work

- Bregman matrix divergences lead to intriguing nearness problems
- Nearness problems with von Neumann & Burg matrix divergences
  - Very useful if rank & null space need to be preserved
- Future Work:
  - Characterize usefulness of preserving null space
  - Detailed investigations into:
    - Nearest correlation matrix problem
    - Kernel learning problem
  - Improvement over cyclic projection methods