Information Theoretic Metric Learning

Instructor: Sham Kakade

1 Metric Learning

In k-nearest neighbors (k-nn) and other classification algorithms, one crucial choice is what metric to use to characterize distances between points. Suppose we are given features $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ where each $x_i \in \mathbb{R}^d$ with associated class labels $\mathcal{Y} = \{y_1, \ldots, y_n\}$, and we seek to learn a k-nn classifier. Recall that if one uses the Euclidean distance in k-nn, typically the first step is to normalize the features x_i such that the sample mean is 0 and the sample standard deviation is 1. I.e, we form new features

$$\tilde{x}_i = \frac{x_i - \bar{x}}{s_x}.$$

Given the test point z we employ this normalization to form a new feature $\tilde{z}$ and then find the k nearest neighbors in $\mathcal{X}$ according to the Euclidean metric, and classify z according to majority vote of the associated labels in $\mathcal{Y}$.

In [DKJ+07], the goal is to learn the metric itself rather than rely on the Euclidean metric and normalization. The authors consider learning the squared Mahalanobis distance given a matrix $A \succ 0$ (i.e., a positive definite matrix), which the authors denote

$$d_A(x, y) = (x - y)^T A(x - y).$$

Additionally, given the training data, one can denote a subset of points as similar (e.g., belong to the same class) and those which are dissimilar (e.g., belong to different classes). Thus, two natural sets of constraints arise,

$$\begin{aligned} (i, j) \in S: &\quad d_A(x_i, x_j) \le u, \\ (i, j) \in D: &\quad d_A(x_i, x_j) \ge \ell, \end{aligned} \tag{1}$$

representing similar and dissimilar points respectively, where the user chooses the parameters u, ℓ.

The authors of [DKJ+07] propose the following optimization problem to learn a metric from the data:

$$\begin{aligned} \min_{A \succeq 0} &\quad D_{\ell d}(A, A_0) \\ \text{s.t.} &\quad \operatorname{tr}(A(x_i - x_j)(x_i - x_j)^T) \le u \text{ for } (i, j) \in S, \\ &\quad \operatorname{tr}(A(x_i - x_j)(x_i - x_j)^T) \ge \ell \text{ for } (i, j) \in D. \end{aligned} \tag{2}$$

Note that the constraints in (2) are precisely those stated (1), which follows from the invariance of the trace to cyclic permutations (i.e., $\operatorname{tr}(ABCD) = \operatorname{tr}(DABC) = \operatorname{tr}(CDAB) = \operatorname{tr}(BCDA)$). The objective function $D_{\ell d}(A, A_0)$ we develop in the sequel.

2 Bregman Divergences

2.1 Definition and Properties

Suppose we have a strictly convex, differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$, defined over a convex set $\Omega = \mathrm{dom}(\phi) \subset \mathbb{R}^d$. Given such a function, one generalized notion of a distance induced by such a function is as follows:

Definition 1 (Bregman Divergence)**.** *The Bregman divergence with respect to ϕ is a map $D_\phi : \Omega \times \mathrm{relint}(\Omega) \to \mathbb{R}$, defined as*

$$D_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla \phi(y),\, x - y \rangle,$$

where $\langle x, y \rangle = x^T y$ denotes the usual inner product in $\mathbb{R}^m$.

Intuitively, it should be clear from the definition that the Bregman divergence measures the error in first order approximation of $\phi(x)$ around y.

The Bregman divergence is not a metric in the usual sense. In particular, $D_\phi(x, y) \neq D_\phi(y, x)$ in general, and the triangle inequality does not hold. We enumerate some of its properties (verify!):

- *Non-negativity*: $D_\phi(x, y) \geq 0$ with equality if and only if $x = y$.
  - Follows directly from the first-order condition of strict convexity for the function ϕ.
- *Strict Convexity in x*: $D_\phi(x, y)$ is strictly convex in its first argument.
  - Follows directly from the first-order condition of strict convexity for the function ϕ.
- *(Positive) Linearity*: $D_{a_1\phi_1 + a_2\phi_2}(x, y) = a_1 D_{\phi_1}(x, y) + a_2 D_{\phi_2}(x, y)$ given $a_1, a_2 > 0$.
- *Gradient in x*: $\nabla_x D_\phi(x, y) = \nabla \phi(x) - \nabla \phi(y)$.
- *Generalized Law of Cosines*: $D_\phi(x, y) = D_\phi(x, z) + D_\phi(z, y) - \langle \nabla \phi(y) - \nabla \phi(z), x - z \rangle$.
  - Follows directly from the definition. Compare to the law of cosines with in Euclidean spaces:

$$\|x - y\|_2^2 = \|x - z\|_2^2 + \|z - y\|_2^2 - 2\|x - z\|_2 \|z - y\|_2 \cos \angle xzy$$

Here are some examples of some Bregman divergences induced by strictly convex functions:

- *Mahalanobis Distance*: Given $A \succ 0$, let $\Omega = \mathbb{R}^d$ and $\phi(x) = x^T A x$. Then $D_\phi(x, y) = (x - y)^T A (x - y)$.
  - *Euclidean Metric*: Letting $\phi(x) = \|x\|_2^2$ results in the Euclidean metric $D_\phi(x, y) = \|x - y\|_2^2$.
- *Generalized Information Divergence*: Let $\Omega = \{x \in \mathbb{R}^d \mid x_i > 0 \text{ for all } i\}$. Then $\phi(x) = \sum_{i=1}^d x_i \log x_i$ implies that $D_\phi(x, y) = \sum_{i=1}^d \Big(x_i \log(\frac{x_i}{y_i}) + (x_i - y_i) \Big)$.
  - *Relative Entropy/Kullback-Leibler (KL) Divergence*: Additionally require that $\langle x, 1 \rangle = 1$ for all $x \in \Omega$. Then $\phi(x) = \sum_{i=1}^d x_i \log x_i$ results in $D_\phi(x, y) = \sum_{i=1}^d x_i \log \frac{y_i}{x_i}$, the KL divergence between probability mass functions x and y.

Finally, we introduce the concept of a Bregman projection onto a convex set.

Definition 2 (Bregman Projection)**.** *Given a Bregman Divergence $D_\phi : \Omega \times \mathrm{relint}(\Omega) \to \mathbb{R}$, a closed convex set $K \subset \Omega$, and a point $x \in \Omega$, the Bregman projection of x onto K is the unique* (why?) *point*

$$x^\star = \mathrm{argmin}_{\tilde{x} \in K}\, D_\phi(\tilde{x}, x). \tag{3}$$

When we consider the function $\phi(x) = \|x\|_2^2$, note that the Bregman projection corresponds to the orthogonal projection onto a convex set, i.e.,

$$x^\star = \text{argmin}_{\tilde{x} \in K} \|\tilde{x} - x\|_2^2, \tag{4}$$

so the Bregman projection generalizes the notion of an orthogonal projection. One can show that a generalization of the Pythagorean theorem for such a projection $x^\star$ holds. Given any $y \in K$, we have

$$D_\phi(x, y) \geq D_\phi(x, x^\star) + D_\phi(x^\star, y).$$

In the Euclidean case, note that by the law of cosines this implies the angle $\angle x x^\star y$ is obtuse.

2.2 Matrix Bregman Divergences

Let $S^n \subset \mathbb{R}^{n\times n}$ denote the space of real symmetric matrices. Given a strictly convex, differentiable function $\phi : S^n \to \mathbb{R}$, the Bregman matrix divergence [DT07] is defined as

$$D_\phi(A, B) = \phi(A) - \phi(B) - \langle \nabla\phi(B), A - B\rangle.$$

Note here that $\langle A, B\rangle = \text{tr}(AB)$ denotes the inner product on the space of symmetric matrices which induces the Frobenius norm, i.e,

$$\langle A, A\rangle = \|A\|_F^2,$$

the sum of the squared entries of A. Usually the function ϕ will be determined by the composition of an eigenvalue map with another convex function, e.g., $\phi = \varphi \circ \lambda$, where $\lambda : S^n \to \mathbb{R}^n$ yields the eigenvalues of a symmetric matrix in decreasing order.

2.2.1 The Log Det (Burg) Divergence and Properties

One important example yields the objective function employed in [DKJ+07]. By taking the *Burg entropy* of the eigenvalues $\{\lambda_i\}_{i=1}^n$ of A, we have

$$\phi(A) = -\sum_{i=1}^n \log \lambda_i = -\log \prod_{i=1}^n \lambda_i = -\log \det A,$$

which is a strictly convex function with domain of the positive definite cone [BV04]. Using this function yields the so-called “Burg” or “log det” divergence,

$$D_{\ell d}(A, B) = \text{tr}(AB^{-1}) - \log\det(AB^{-1}) - n. \tag{5}$$

To see this, note that $\phi(A) - \phi(B) = -\log\det(AB^{-1})$, the trace is invariant to cyclic permutations, and $\nabla\phi(B) = -B^{-1}$.

To deduce that $\nabla\phi(X) = -X^{-1}$, one approach is given in [BV04] is to argue via a first-order approximation as follows. Let $Z = X + \Delta X$. Then

$$\begin{aligned} \log\det Z &= \log\det(X^{1/2}(I + X^{-1/2}\Delta X X^{-1/2})X^{1/2}) \\ &= \log\det X + \log\det(I + X^{-1/2}\Delta X X^{-1/2}) \\ &= \log\det X + \sum_{i=1}^n \log(1 + \lambda_i), \end{aligned}$$

where λ_i denotes the ith largest eigenvalue of $X^{-1/2}\Delta X X^{-1/2}$. For small x the first order approximation yields $\log(1+x) \approx x$. Since ΔX is small in terms of its eigenvalues, it follows that the λ_i's must be small, and

$$\begin{aligned}\log\det Z &\approx \log\det X + \sum_{i=1}^{n} \lambda_i \\ &= \log\det X + \operatorname{tr}(X^{-1/2}\Delta X X^{-1/2}) \\ &= \log\det X + \operatorname{tr}(X^{-1}\Delta X) \\ &= \log\det X + \operatorname{tr}(X^{-1}(Z-X)),\end{aligned}$$

a first order approximation of $\log\det$ at X. This could also be derived directly,

$$\frac{\partial}{\partial X_{ij}} \log\det X = \frac{1}{\det X}\frac{\partial \det X}{\partial X_{ij}} = \frac{1}{\det X}(\operatorname{adj}(X))_{ji} = (X^{-1})_{ji},$$

where $\operatorname{adj}(X)$ is the *classical adjoint* of a square invertible matrix X.

Important properties of the Burg matrix divergence are as follows:

- Given invertible B, minimizing $D_{\ell d}(A, B)$ over a symmetric matrix A guarantees that A will be invertible given the domain of the log determinant. Thus, one need not explicitly enforce $A \succ 0$ in (2).
- Given any invertible square matrix M, it is easy to verify that

  $$D_{\ell d}(A, B) = D_{\ell d}(M^T AM,\ M^T BM),$$

  whence the divergence of (5) remains *invariant under any rescaling* of the feature space.
- The matrix divergence in equation (5) is (up to a constant) *equivalent to the KL divergence between two multivariate Gaussian distributions with the same mean.* Given Gaussian probability measures P_1 and P_2 with associated densities p_1 and p_2, one may show the KL divergence is

  $$\begin{aligned}D_{KL}(P_1\|P_2) &= \int p_1(x)\log\frac{p_1(x)}{p_2(x)}\,dx \\ &= \frac{1}{2}\Big(\operatorname{tr}\left(\Sigma_2^{-1}\Sigma_1\right) - \log\det\left(\Sigma_2^{-1}\Sigma_1\right) - n + \left(\mu_2-\mu_1\right)^T\Sigma_2^{-1}\left(\mu_2-\mu_1\right)\Big).\end{aligned}$$

  Thus, if we seek to minimize the Burg divergence of a matrix $A \succ 0$ with respect to a reference matrix A_0, we have

  $$D_{\ell d}(A, A_0) = 2D_{KL}(P_0\|P),$$

  where the Gaussian distributions P and P_0 have the same mean and covariance matrices A^{-1}, A_0^{-1}, respectively. Thus, given the usual interpretation of KL divergence, our objective function $D_{\ell d}(A, A_0)$ measures the cost in approximating a Gaussian distribution with precision matrix A in place of the precision matrix A_0.

3 Computing Bregman Projections

3.1 Dykstra's Cyclic Projection Algorithm

Consider the problem of finding a nearest point in the intersection of convex sets. We seek to solve (4) for the case when a point $x^\star \in K = \cap_{i=1}^m C_i$ where each C_i is convex. One intuitive algorithm is to cyclically project the current estimate onto each C_i until we find a point in K. That is, we let $x_0 = x$ in (4), and repeat the following for $t \geq 1$ until a point $x_t \in K$ is found:

$$x_t = \mathcal{P}_{C_{[t]_m}}(x_{t-1}). \tag{6}$$

Here $[t]_m$ denotes t modulo m and $\mathcal{P}_C$ denotes the orthogonal projection onto a convex set C. This simple routine is known as *Dykstra's cyclic projection algorithm*. This algorithm is known to converge generally [DH06a, DH06b, DH08]. In the special case of all C_i being half spaces that defines a polyhedral K, the algorithm converges linearly [DH94], i.e.,

$$\|x_t - \mathcal{P}_K(x)\|_2 \leq c\rho^t \|x - \mathcal{P}_K(x)\|_2$$

for all t for some constants $c > 0$, $\rho \in (0, 1)$.

3.2 Generalized Dykstra's Cyclic Projection Algorithm

The authors of [CR98] extended this idea to the case of the Bregman projection of equation (3), showing it converges in the polyhedral case. The authors of [BL00] analyzed the problem generally, showing that it converges for any finite intersection of convex sets. As far as I am aware, the rates of convergence are not well understood in general, or for the special case of the algorithm employed in [DKJ$^+$07], and remain an open question. Additionally, the costs of projecting onto each C_i is non-trivial in general, but for the constraints employed in [DKJ$^+$07], they may be computed efficiently.

3.3 Bregman Projection of a Matrix onto Equality and Inequality Constraints

Presume we are solving a generalized Dykstra's cyclic projection algorithm to minimize $D_\phi(A, A_0)$ over an intersection of m convex sets, $\cap_{i=1}^m C_i$. Let the current iterate be A_t, and assume $k = [t]_m$. Presume k is such that we must solve the following equality-constrained projection for this iterate:

$$\begin{aligned} \min_{A \succ 0} \quad & D_\phi(A, A_t) \\ \text{s.t.} \quad & \text{tr}(AB_k) = b_k. \end{aligned} \tag{7}$$

To solve (7), introducing the dual variable α_k, we form the Lagrangian

$$L(X, \alpha_k) = D_\phi(A, A_t) + \alpha_k(b_k - \text{tr}(AB_k)).$$

By setting the gradient with respect to A and α_k to zero (recall the *gradient in x* property of Bregman divergence), we obtain the Bregman projection A_{t+1} onto C_k by solving

$$\begin{aligned} \nabla\phi(A) &= \nabla\phi(A_t) + \alpha_k B_k \\ \text{tr}(AB_k) &= b_k \end{aligned} \tag{8}$$

for A and α_k. If we instead had an inequality constraint, i.e.,

$$\begin{aligned} \min_{A \succ 0} \quad & D_\phi(A, A_t) \\ \text{s.t.} \quad & \text{tr}(AB_k) \leq b_k, \end{aligned} \tag{9}$$

we introduce the corresponding dual variable $\lambda_k \geq 0$, which we set to 0 for all $k \in \{1, \ldots, m\}$ when we start the algorithm. Recall the KKT conditions require this dual variable to be non-negative. Thus, after solving (8) for α_k, we letting $\alpha'_k = \min(\lambda_k, \alpha_k)$, we update the Lagrange multiplier λ_k associated with constraint k as follows:

$$\lambda_k \leftarrow \lambda_k - \alpha'_k.$$

Note that this ensures $\lambda_k \geq 0$. Finally, we form the update A_{t+1} by solving

$$\nabla\phi(A) = \nabla\phi(A_t) + \alpha'_k B_k$$

for A subject to $\mathrm{tr}(AB_k) \leq b_k$.

In the case where $\phi(A) = -\log\det A$ and the matrix $B_k = z_k z_k^T$, we may avoid matrix inversion. In this case, solving (8) reduces to solving

$$\begin{aligned} A &= (A_t - \alpha_k z_k z_k^T)^{-1}, \\ b_k &= z_k^T A z_k. \end{aligned} \tag{10}$$

Recall the Sherman-Morrison inverse formula for an invertible matrix M,

$$(M + uv^T)^{-1} = M^{-1} - \frac{M^{-1}uv^T M^{-1}}{1 + v^T M^{-1} u}. \tag{11}$$

Applying (11) to (10), letting $p = z_k^T A_t z_k$, and solving for A, it follows that our next iterate is

$$A_{t+1} = A_t + \beta A_t z_k z_k^T A_t, \tag{12}$$

where

$$\begin{aligned} \alpha_k &= \frac{1}{p} - \frac{1}{b}, \\ \beta &= \frac{\alpha_k}{1 - \alpha_k p}. \end{aligned}$$

4 The "Information-Theoretic" Metric Learning Algorithm

Given the previous section, the algorithm employed in [DKJ$^+$07] should be straightforward to state by noticing that each (i,j) in the constraint set of (2) corresponds to a constraint of the form of (9) with $B_k = (x_i - x_j)(x_i - x_j)^T$. However, it may be the case that the constraint set of (2) is empty. Thus, the authors introduce a vector of slack variables $\xi \in \mathbb{R}^m$ corresponding to each of the m constraints in (2), initialized to ξ_0 (whose components equal u for similarity constraints and ℓ for dissimilarity constraints).

$$\begin{aligned} \min_{A \succeq 0,\, \xi} \quad & D_{\ell d}(A, A_0) + \gamma\, D_{\ell d}(\mathrm{diag}(\xi), \mathrm{diag}(\xi_0)) \\ \text{s.t.} \quad & \mathrm{tr}(A(x_i - x_j)(x_i - x_j)^T) \leq \xi_{c(i,j)} \text{ for } (i,j) \in S, \\ & \mathrm{tr}(A(x_i - x_j)(x_i - x_j)^T) \geq \xi_{c(i,j)} \text{ for } (i,j) \in D. \end{aligned} \tag{13}$$

The parameter $\gamma > 0$ is a regularization parameter chosen via cross-validation. Given the development in the previous section keeping in mind the linearity property of Bregman divergence, it is easy to verify their algorithm. Given a matrix $X \in \mathbb{R}^{d \times n}$ comprised of n training samples, a similarity set S, a dissimilarity set D, an input Mahalanobis matrix A_0, a slack parameter γ, and a constraint index function

$$c : \{1, \ldots, n\} \times \{1, \ldots, n\} \to \{1, \ldots, m\},$$

the algorithm is as follows:

1. *Initialization*:
   (a) $A \leftarrow A_0$
   (b) $\lambda_{ij} \leftarrow 0$ for all i, j.
   (c) $\xi_{c(i,j)} \leftarrow u$ for $(i,j) \in S$.
   (d) $\xi_{c(i,j)} \leftarrow \ell$ for $(i,j) \in D$.

2. *Repeat Until Convergence*:

   (a) Pick a constraint $(i, j) \in S$ or $(i, j) \in D$.

   (b) $p \leftarrow (x_i - x_j)^T A(x_i - x_j)$.

   (c) $\delta \leftarrow 1$ if $(i, j) \in S$, else $\delta \leftarrow -1$ (if $(i, j) \in D$).

   (d) $\alpha \leftarrow \min\left(\lambda_{ij},\ \frac{\delta}{2}\left(\frac{1}{p} - \frac{\gamma}{\xi_{c(i,j)}}\right)\right)$

   (e) $\beta \leftarrow \frac{\delta\alpha}{1-\delta\alpha p}$.

   (f) $\xi_{c(i,j)} \leftarrow \frac{\gamma\xi_{c(i,j)}}{\gamma+\delta\alpha\xi_{c(i,j)}}$.

   (g) $\lambda_{ij} \leftarrow \lambda_{ij} - \alpha$.

   (h) $A \leftarrow A + \beta A(x_i - x_j)(x_i - x_j)^T A$.

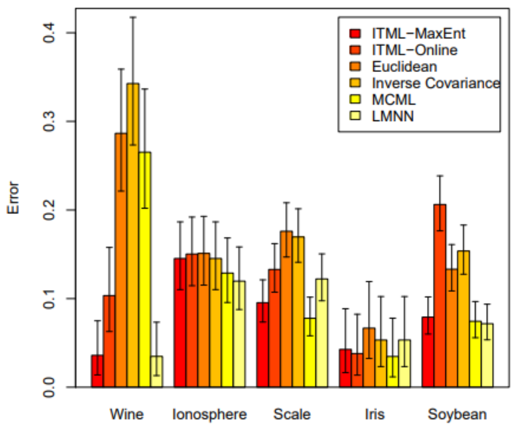
3. *Return*: A.

Note that each constraint projection costs $O(d^2)$, so a single iteration of looping through each of the m constraints costs $O(md^2)$. Typically this cost would be $O(md^3)$ in practice if we depended on a matrix inversion or an eigenvalue decomposition for each of the constraints.

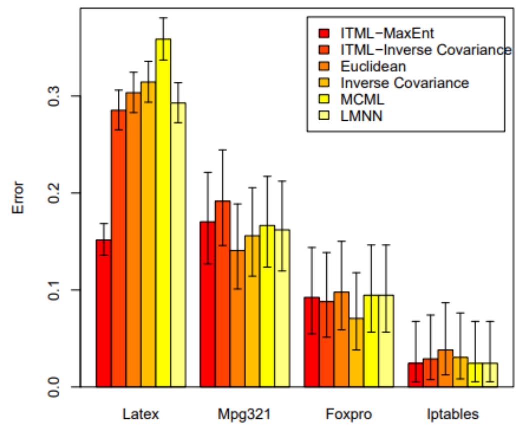
5 Empirical Results

Refer to [DKJ+07] for precise details of the datasets used and the algorithms employed, but we briefly review the experiments run. The main experiments evaluated metric learning for k-nn classification with $k = 4$, averaged over 5 runs. The parameters ℓ and u were chosen, respectively, to be the 5-th and 95-th percentiles of the Euclidean distances amongst points in the training set. The set S was constrained to be of points with the same class label, and the set D was constrained to be points with different class labels. A total of $20c^2$ training points were chosen at random to comprise S and D, where c is the number of classes in the data. The matrix A_0 was chosen to be either the identity (so the objective function corresponded to maximizing the entropy of a Gaussian) or the inverse of the sample covariance. The parameter γ was chosen from $\{.01, .1, 1, 10\}$ via two-fold cross-validation. The results on various datasets with 95% confidence intervals are shown below.

Note: The authors also developed an online version of their algorithm which we did not review here. See [DKJ+07] for details.



(a) UCI Datasets



(b) Clarify Datasets

References

[BL00] Heinz H Bauschke and Adrian S Lewis. Dykstras algorithm with bregman projections: A convergence proof. *Optimization*, 48(4):409–427, 2000.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[CR98] Yair Censor and Simeon Reich. The dykstra algorithm with bregman projections. *Communications in Applied Analysis*, 2(3):407–420, 1998.

[DH94] Frank Deutsch and Hein Hundal. The rate of convergence of Dykstra's cyclic projections algorithm: The polyhedral case. *Numerical Functional Analysis and Optimization*, 15(5-6):537–565, 1994.

[DH06a] Frank Deutsch and Hein Hundal. The rate of convergence for the cyclic projections algorithm i: angles between convex sets. *Journal of Approximation Theory*, 142(1):36–55, 2006.

[DH06b] Frank Deutsch and Hein Hundal. The rate of convergence for the cyclic projections algorithm ii: norms of nonlinear operators. *Journal of Approximation Theory*, 142(1):56–82, 2006.

[DH08] Frank Deutsch and Hein Hundal. The rate of convergence for the cyclic projections algorithm iii: Regularity of convex sets. *Journal of Approximation Theory*, 155(2):155–184, 2008.

[DKJ+07] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[DT07] Inderjit S Dhillon and Joel A Tropp. Matrix nearness problems with bregman divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2007.