DISCOMAX: Distance Correlation Maximization using Graph Laplacians

Praneeth Vepakomma [1] Chetan Tonde [2] Ahmed Elgammal [2]

[1]Public Engines Inc.

[2]Department of Computer Science Rutgers University

RUTGERS

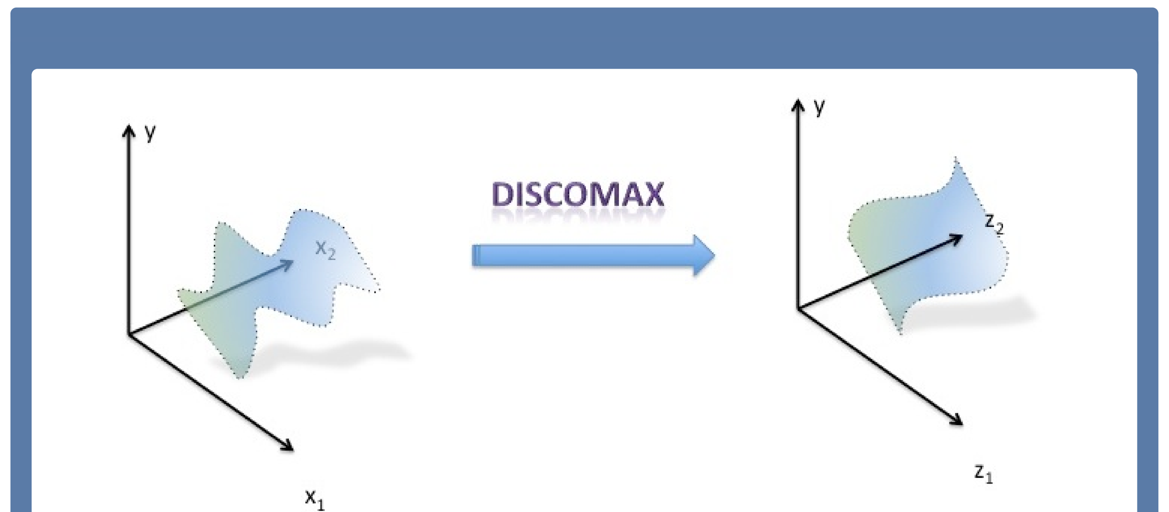
PublicEngines™

Objective

We propose an algorithm (DISCOMAX) to learn feature representations (Z) of input data (X) for the *regression* setting that maximize *statistical distance correlation* [1] between learnt features (Z) and responses (Y).

Introduction

- **Input**: $X \times Y = (\mathbf{x}_i, y_i)^n \subset \mathbb{R}^D \times \mathbb{R}$ I.I.D. samples from joint distribution $P(\mathbf{x}, y)$.
- **Output**: Learn features $Z = \{\mathbf{z}_i\}_{i=1}^n \subset \mathbb{R}^d$ with improved correlation with the response $Y = \{\mathbf{y}_i\}_{i=1}^n$ for better regression.
- We propose to maximize *Distance Correlation*[1], which is a *non-linear* measure of statistical dependence between two r.v.'s, the learnt features Z and responses Y, as opposed to *Pearson's correlation* which is *linear*.



Distance Correlation

- *Distance Correlation* introduced by [1] is a measure of any *nonlinear dependencies* between r.v.'s of *arbitrary dimensions*.

$$\rho^2(Z,Y) = \frac{\nu^2(Z,Y;\phi)}{\sqrt{\nu^2(Z,Z;\phi)\nu^2(Y,Y;\phi)}} \qquad (1)$$

- *Distance Covariance* between two r.v.'s is, $\nu^2(Z,Y;\phi)$: $\int_{\mathbb{R}^{h+m}} |f_{Z,Y}(t,s) - f_Z(t) f_Y(s)|^2 \phi(t,s) dt ds$ where $f_Z, f_Y, f_{Z,Y}$ are characteristic functions and $\phi(t,s)$ is a specific *weight function*.

- *Sample Distance Correlation* is given by,

$$\hat{\rho}^2(Z,Y) = \frac{\hat{\nu}^2(Z,Y;\phi)}{\sqrt{\hat{\nu}^2(Z,Z;\phi)\hat{\nu}^2(Y,Y;\phi)}} \qquad (2)$$

where $\hat{\nu}^2(Z,Y) = \frac{1}{n^2}\sum_{k,l=1}^n [\mathbf{E}^X]_{kl}[\mathbf{E}^Y]_{kl}$ and $\mathbf{E}^X, \mathbf{E}^Y$ are double-centered squared euclidean distance matrices.

Graph Laplacian Formulation

We propose the *Graph Laplacian* form of *sample distance correlation* $\hat{\rho}^2(\mathbf{Z},\mathbf{Y})$ with Laplacians $\mathbf{L_Z}$ and $\mathbf{L_Y}$ formed over adjacency matrices $\mathbf{E^Z}$, $\mathbf{E^Y}$ as below,

$$\hat{\rho}^2(Z,Y) = \frac{n}{2}\frac{\mathbf{tr}(\mathbf{Z^T L_Y Z})}{\mathbf{tr}(\mathbf{Y^T L_Y Y})\mathbf{tr}(\mathbf{Z^T L_Z Z})} \qquad (3)$$

Problem Formulation

- We propose the the following objective function with an additional regularization parameter $C > \kappa^2 = \alpha_{max}\mathbf{tr}(\mathbf{L}_Y)$,

$$\min_{\mathbf{Z}} \frac{\mathbf{tr}(\mathbf{Z}^T(\mathbf{L_Z} + C\mathbf{I})\mathbf{Z})}{\mathbf{tr}(\mathbf{Z}^T\mathbf{L_Y Z})} \quad \text{subject to } \mathbf{Z} \in \mathbb{R}^d \backslash \{0\} \qquad (4)$$

- We oppose an auxiliary objective function by replacing $\mathbf{L_Z}$ and $\mathbf{L_X}$, which we minimize,

$$\min_{\mathbf{Z}} \frac{\mathbf{tr}(\mathbf{Z}^T(\mathbf{L_X} + C\mathbf{I})\mathbf{Z})}{\mathbf{tr}(\mathbf{Z}^T\mathbf{L_Y Z})} \quad \text{subject to } \mathbf{Z} \in \mathbb{R}^d \backslash \{0\} \qquad (5)$$

- This is a Quadratic Fractional Programming Problem and is equivalent to minimizing the parametric problem for some α ([2]),

$$\min_{\mathbf{Z}} F(\alpha) = \mathbf{tr}(\mathbf{Z}^T\mathbf{L_X Z}) - \alpha\mathbf{tr}(\mathbf{Z}^T\mathbf{L_Y Z})) + C\mathbf{tr}(\mathbf{Z}^T\mathbf{Z}) \quad \text{subject to } \mathbf{Z} \in \mathbb{R}^d \backslash \{0\} \qquad (6)$$

- **Theorem 1**: (Majorization-Minimization, [3]) For any fixed $\gamma^2 > 1$ and for the iteration $\mathbf{Z}_t = \mathbf{H}Z_{t-1}$ with $\mathbf{H} = (\gamma^2\mathbf{D}_x + \mathbf{CI} - \alpha\mathbf{L}_y)^{-1}$ $(\gamma^2\mathbf{D}_x - \mathbf{L}_x)$ monotonically minimizes (6).
- **Theorem 2**: For the above iteration, we have $\rho(\mathbf{H}_t) < 1$, (See [4]) .
- **Theorem 3**: Monotonically minimizing (5) also monotonically minimizes (4).

Algorithm

Algorithm: DISCOMAX

- **Step 0**: Pick regularizer $C > \kappa^2 = \alpha_{max}\mathbf{tr}(\mathbf{L}_Y)$, $\alpha_t^{min} = 0$ and $\alpha_t^{max} = \frac{\mathbf{tr}(\mathbf{X}(\mathbf{L_X}+C\mathbf{I})\mathbf{X})}{\mathbf{tr}(\mathbf{X L_Y X})}$, $\eta = \frac{1+\sqrt{5}}{2}$.
- **Step 1**: Set $d = \eta(\alpha_t^{max} - \alpha_t^{min})$, $x_1 = \alpha_t^{min} + d$ and $x_2 = \alpha_t^{max} - d$.
- **Step 2**: Solve (6) for $F(x_1)$ using *Theorem 1*.
- **Step 3**: Solve (6) for $F(x_2)$ using *Theorem 1*.
- **Step 4**: **if** $|\alpha_t^{max} - \alpha_t^{min}| \leq \epsilon$ then, return $\mathbf{Z}^*$ and terminate. Otherwise,
- **Step 5**: **if** $(F(x_1) > F(x_1))$ **then**, $\alpha_t^{min} = x_2$, $x_2 = x_1$ and $x_1 = \alpha_t^{min} + \eta(\alpha_t^{max} - \alpha_t^{min})$.
- **Step 6**: **if** $(F(x_1) < F(x_2))$ **then**, $\alpha_t^{max} = x_1$, $x_1 = x_2$ and $x_2 := \alpha_t^{max} - \eta(\alpha_t^{max} - \alpha_t^{min})$.
- **Step 7**: Let $t := t + 1$ and return to **Step 1**.

Experiments

Regression/Features	Original	DISCOMAX
Linear Regression (LR)	0.1885 (0.0332)	**0.1514 (0.0321)**
Random Forest (RF)	0.1509 (0.0376)	**0.0874 (0.0352)**
Node Harvest (NH)	0.1735 (0.0387)	**0.1189 (0.0344)**
Support Vect. Reg. (SVR)	0.1686 (0.0364)	**0.0826 (0.0349)**

Table 1: Boston Housing: Cross Validation RMSE (SD)

Regression/Features	Original	DISCOMAX
Linear Regression (LR)	2.5369 (0.3352)	2.0721 (0.3837)
Random Forest (RF)	1.8658 (0.3984)	**0.8687 (0.3856)**
Node Harvest (NH)	2.3570 (0.4171)	2.2285 (0.4296)
Support Vect. Reg. (SVR)	1.9013 (0.3761)	**0.8572 (0.3883)**

Table 2: Energy Efficiency, Univ. of Oxford: Cross Validation RMSE (SD)

Regression/Features	Original	DISCOMAX
Linear Regression (LR)	2.1064 (0.1258)	**1.4695 (0.1372)**
Random Forest (RF)	2.0914 (0.1326)	**1.6537 (0.1322)**
Node Harvest (NH)	2.2514 (0.1608)	**1.5752 (0.1415)**
Support Vect. Reg. (SVR)	2.1752 (0.1423)	**1.4960 (0.1404)**

Table 3: Wind Speed: Cross Validation RMSE (SD)

Regression/Features	Original	DISCOMAX
Linear Regression (LR)	10.6523 (0.4901)	5.0951 (0.4063)
Random Forest (RF)	6.1548(0.5449)	6.1306 (0.386)
Node Harvest (NH)	9.1833 (0.5203)	**7.7326 (0.3217)**
Support Vect. Reg. (SVR)	6.2134 (0.5123)	**4.5178 (0.4053)**

Table 4: Compressive Strength: Cross Validation RMSE (SD)

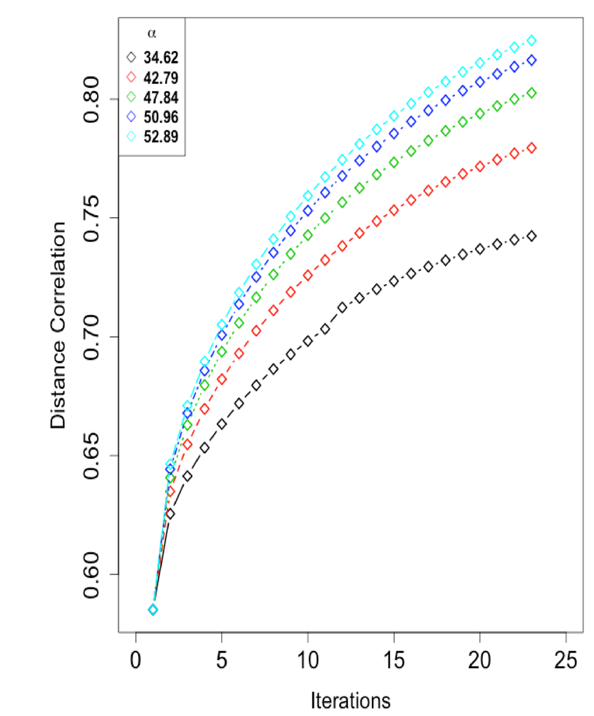


Figure 1: α_t versus DCorr.

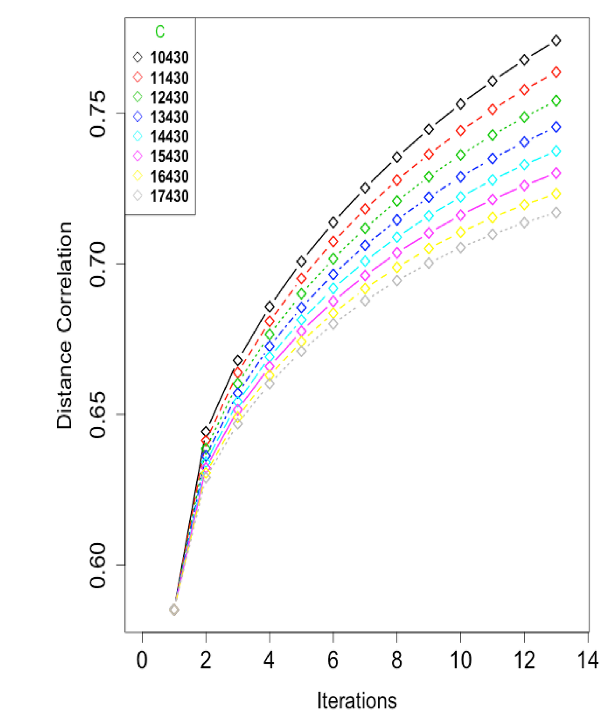


Figure 1: C versus DCorr.

Conclusion

- We observe that the features learnt from DISCOMAX improve the cross-validation error in comparison to using the original features.
- We also observe the concave nature of distance correlation with respect parameter α_t (Figure 1).
- We also observe as expected, increasing C regularizes the maximum *distance correlation* achieved for a fixed number of iterations (Figure 2).

References

[1] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and Testing Dependence by Correlation of Distances," *The annals of statistics*, vol. 35, pp. 2769–2794, Dec. 2007.

[2] A. Zhang, "Quadratic Fractional Programming Problems with Quadratic Constraints," Feb. 2008.

[3] K. Lange, D. R. Hunter, and I. Yang, "Optimization Transfer Using Surrogate Objective Functions," *Journal of Computational and Graphical Statistics*, vol. 9, p. 1, Mar. 2000.

[4] Y. Zhang, R. Tapia, and L. Velazquez, "On Convergence of Minimization Methods: Attraction, Repulsion, and Selection," *Journal of Optimization Theory and Applications*, vol. 107, no. 3, pp. 529–546, 2000.

Contact Information

Email: praneeth.vepakomma@publicengines.com, {cjtonde,elgammal}@cs.rutgers.edu