# Bregman Divergences for Data Mining Meta-Algorithms

Joydeep Ghosh

University of Texas at Austin

*ghosh@ece.utexas.edu*

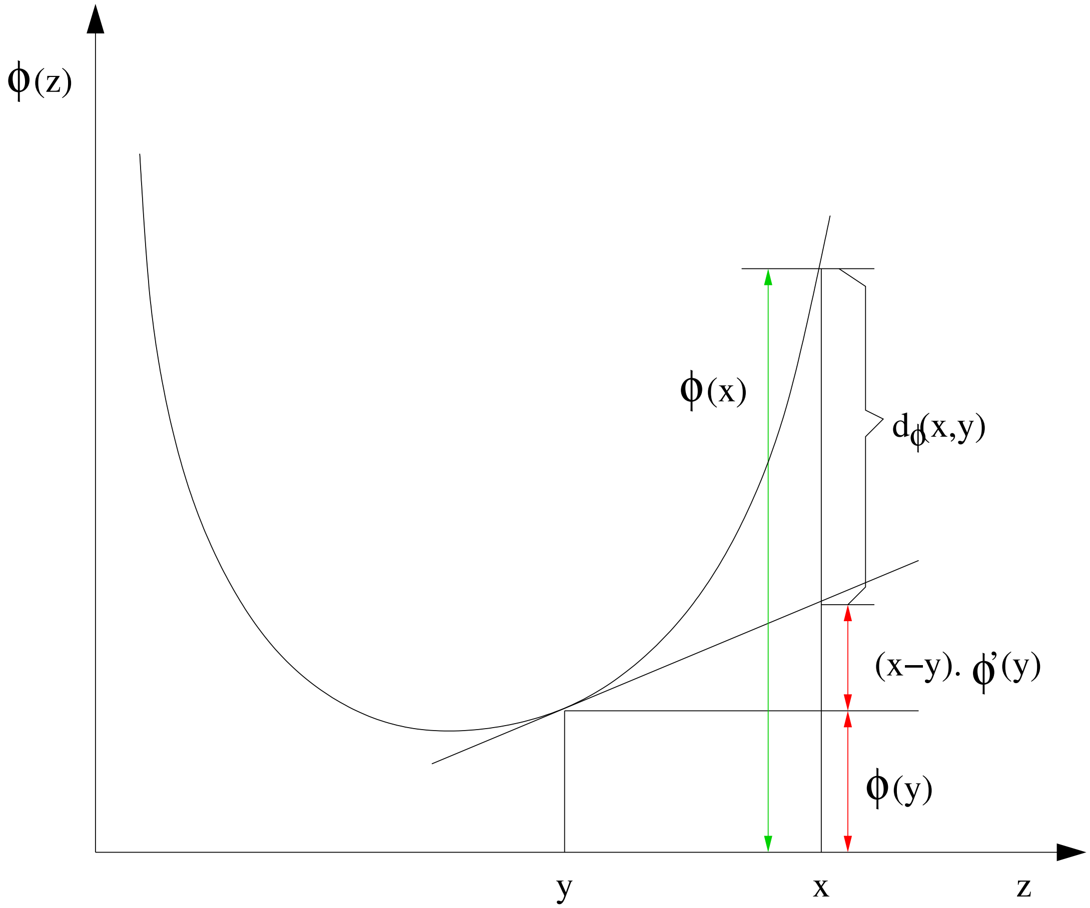
Reflects joint work with

Arindam Banerjee, Srujana Merugu, Inderjit Dhillon, Dharmendra Modha

# Measuring Distortion or Loss

- Squared Euclidean distance.
  - kmeans clustering, least square regression, Weiner filtering,..
- Squared loss is not appropriate in many situations
  - Sparse, high-dimensional data
  - Probability distributions
    - KL-divergence (relative entropy)
- What distortion/loss functions make sense, and where?
- Common properties? (meta-algorithms)

# Bregman Divergences




$\phi$ is strictly convex, differentiable

$$d_\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla\phi(\mathbf{y}) \rangle$$

# Examples

- $\phi(\mathbf{x}) = \|\mathbf{x}\|^2$ is strictly convex and differentiable on $\mathbb{R}^m$
  - $d_\phi(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$ [ squared Euclidean distance ]

- $\phi(\mathbf{p}) = \sum_{j=1}^m p_j \log p_j$ (negative entropy) is strictly convex and differentiable on the $m$-simplex
  - $d_\phi(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^m p_j \log\left(\frac{p_j}{q_j}\right)$ [ KL-divergence ]

- $\phi(\mathbf{x}) = -\sum_{j=1}^m \log x_j$ is strictly convex and differentiable on $\mathbb{R}^m_{++}$
  - $d_\phi(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^m \left(\frac{x_j}{y_j} - \log\left(\frac{x_j}{y_j}\right) - 1\right)$ [ Itakura-Saito distance ]

# Properties of Bregman Divergences

- $d_\phi(\mathbf{x}, \mathbf{y}) \geq 0$, and equals $0$ iff $\mathbf{x} = \mathbf{y}$, but not a metric (symmetry, triangle inequality do not hold)
- Convex in the first argument, but not necessarily in the second one
- KL divergence between two distributions of the same exponential family is a Bregman divergence
- Generalized Law of Cosines and Pythagoras Theorem:

$$d_\phi(\mathbf{x}, \mathbf{y}) = d_\phi(\mathbf{z}, \mathbf{y}) + d_\phi(\mathbf{x}, \mathbf{z}) - \langle (\mathbf{x} - \mathbf{z}), (\nabla\phi(\mathbf{y}) - \nabla\phi(\mathbf{z})) \rangle$$

When $\mathbf{x} \in$ convex (affine) set $\Omega$ & $\mathbf{z}$ is the Bregman projection onto $\Omega$

$$\mathbf{z} \equiv P_\Omega(\mathbf{y}) = \underset{\omega \in \Omega}{\operatorname{argmin}}\, d_\phi(\omega, \mathbf{y}),$$

the inner product term becomes negative (equals zero)

# Bregman Information

- For squared loss
  - Mean is the best constant predictor of a random variable

$$\boldsymbol{\mu} = \underset{\mathbf{c}}{\text{argmin}}\ E[\|X - \mathbf{c}\|^2]$$

  - The minimum loss is the variance $E[\|X - \boldsymbol{\mu}\|^2]$
- <u>Theorem:</u> For all Bregman divergences

$$\boldsymbol{\mu} = \underset{\mathbf{c}}{\text{argmin}}\ E[d_\phi(X, \mathbf{c})]$$

- <u>Definition:</u> The minimum loss is the Bregman information of $X$

$$I_\phi(X) = E[d_\phi(X, \boldsymbol{\mu})]$$

- (minimum distortion at Rate = 0)

# Examples of Bregman Information

- $\phi(\mathbf{x}) = \|\mathbf{x}\|^2$, $X \sim \nu$ over $\mathbb{R}^m$
  - $I_\phi(X) = E_\nu[\|X - E_\nu[X]\|^2]$ [ Variance ]

- $\phi(\mathbf{x}) = \sum_{j=1}^m x_j \log x_j$, $X \sim p(z)$ over $\{p(Y|z)\} \subset m$-simplex
  - $I_\phi(X) = I(Z;Y)$ [ Mutual Information ]

- $\phi(\mathbf{x}) = -\sum_{j=1}^m \log x_j$, $X \sim$ uniform over $\{\mathbf{x}^{(i)}\}_{i=1}^n \subset \mathbb{R}^m$
  - $I_\phi(X) = \sum_{j=1}^m \log \left( \frac{\boldsymbol{\mu}_j}{\mathbf{g}_j} \right)$ [ log AM/GM ]

# Bregman Hard Clustering Algorithm

- (Std. Objective is same as minimizing loss in Bregman Information when using K representatives.)
- Initialize $\{\boldsymbol{\mu}_h\}_{h=1}^k$
- Repeat until *convergence*
  - { Assignment Step }
    Assign $\mathbf{x}$ to nearest cluster $\mathcal{X}_h$ where

$$h = \underset{h'}{\operatorname{argmin}}\, d_\phi(\mathbf{x}, \boldsymbol{\mu}_{h'})$$

  - { Re-estimation step }
    For all $h$, recompute mean $\boldsymbol{\mu}_h$ as

$$\boldsymbol{\mu}_h = \frac{\sum_{\mathbf{x} \in \mathcal{X}_h} \mathbf{x}}{n_h}$$

# Properties

- Guarantee: Monotonically decreases objective function till convergence
- Scalability: Every iteration is linear in the size of the input
- Exhaustiveness: If such an algorithm exists for a loss function $L(\mathbf{x}, \boldsymbol{\mu})$, then $L$ has to be a Bregman divergence
- Linear Separators: Clusters are separated by hyperplanes
- Mixed Data types: Allows appropriate Bregman divergence for subsets of features

# Example of Algorithms

| Convex function | Bregman divergence | Algorithm |
|---|---|---|
| Squared norm | Squared Loss | KMeans [M'67] |
| Negative entropy | KL-divergence | Information Theoretic [DMK'03] |
| Burg entropy | Itakura-Saito distance | Linde-Buzo-Gray [LBG'80] |

# Bijection between BD and Exponential Family

Regular exponential families $\longleftrightarrow$ Regular Bregman divergences

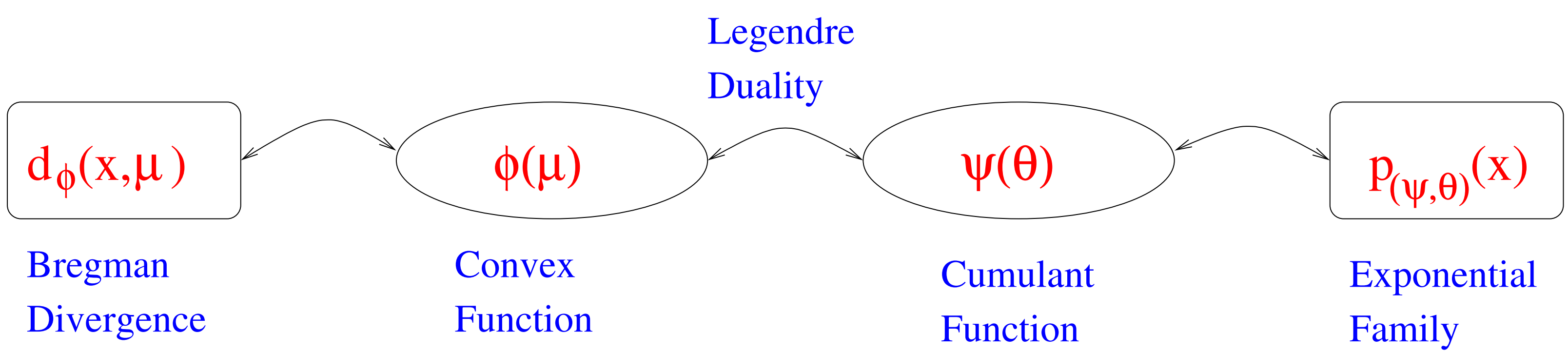
| | | |
|---|---|---|
| Gaussian | $\longleftrightarrow$ | Squared Loss |
| Multinomial | $\longleftrightarrow$ | KL-divergence |
| Geometric | $\longleftrightarrow$ | Itakura-Saito distance |
| Poisson | $\longleftrightarrow$ | I-divergence |

# Bregman Divergences and Exponential Family

- Theorem: For any regular exponential family $p_{(\psi,\boldsymbol{\theta})}$, for all $\mathbf{x} \in \mathrm{dom}(\phi)$,

$$p_{(\psi,\boldsymbol{\theta})}(\mathbf{x}) = \exp(-d_\phi(\mathbf{x},\boldsymbol{\mu}))b_\phi(\mathbf{x}),$$

for a uniquely determined $b_\phi$, where $\boldsymbol{\theta}$ is the natural parameter and $\mu$ is the expectation parameter



# Bregman Soft Clustering

- Soft Clustering
  - Data modeling with mixture of exponential family distributions
  - Solved using Expectation Maximization (EM) algorithm
- Maximum log-likelihood $\equiv$ Minimum Bregman divergence

$$\log p_{(\psi,\boldsymbol{\theta})}(\mathbf{x}) \equiv -d_\phi(\mathbf{x},\mu)$$

- Bijection implies a Bregman divergence viewpoint
  - Efficient algorithm for soft clustering

# Bregman Soft Clustering Algorithm

- Initialize $\{\pi_h, \boldsymbol{\mu}_h\}_{h=1}^k$
- Repeat until *convergence*
  - { Expectation Step }
    For all $\mathbf{x}$, $h$, the posterior probability

    $$p(h|\mathbf{x}) = \pi_h \exp(-d_\phi(\mathbf{x}, \boldsymbol{\mu}_h))/Z(\mathbf{x}),$$

    where $Z(\mathbf{x})$ is the normalization function
  - { Maximization step }
    For all $h$,

    $$\pi_h = \frac{1}{n}\sum_{\mathbf{x}} p(h|\mathbf{x})$$

    $$\boldsymbol{\mu}_h = \frac{\sum_{\mathbf{x}} p(h|\mathbf{x})\ \mathbf{x}}{\sum_{\mathbf{x}} p(h|\mathbf{x})}$$

# Rate Distortion with Bregman Divergences

- Theorem: If distortion is a Bregman divergence,
  - Either, $R(D)$ is equal to the Shannon-Bregman lower bound
  - Or, $|\hat{X}|$ is finite
- When $|\hat{X}|$ is finite

| | | |
|---|---|---|
| Bregman divergences | $\leftrightarrow$ | Exponential family distributions |
| Rate distortion with Bregman divergences | $\leftrightarrow$ | Modeling with mixture of exponential family distributions |

- $R(D)$ can be obtained either analytically or computationally
- Compression vs. loss in Bregman information formulation
  - Information bottleneck as a special case

# Online Learning (Warmuth)

- Setting: For trials $t = 1, \cdots, T$ do
  - Predict target $\hat{y}_t = g(\mathbf{w}_t \cdot \mathbf{x}_t)$ for instance $\mathbf{x}_t$ using link function $g$
  - Incur loss $L_t^{curr}(\mathbf{w}_t)$
- Update Rule: $\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \left( \underbrace{L^{hist}(\mathbf{w})}_{Deviation\ from\ history} + \eta_t \underbrace{L_t^{curr}(\mathbf{w})}_{Current\ loss} \right)$
  - When $L^{hist}(\mathbf{w}) = d_F(\mathbf{w}, \mathbf{w}_t)$, i.e., a Bregman loss function and $L_t^{curr}(\mathbf{w})$ is convex, the update rule reduces to

$$\mathbf{w}_{t+1} = f^{-1}\left(f(\mathbf{w}_t) + \eta_t \nabla L_t^{curr}(\mathbf{w}_t)\right) \text{ where } f = \nabla F$$

  - Also get Regret Bounds.
  - and density estimation Bounds.

# Examples

| History loss:Update family | Current loss | Algorithm |
|---|---|---|
| Squared Loss: Gradient Descent | Squared Loss | Widrow Hoff(LMS) |
| Squared Loss: Gradient Descent | Hinge Loss | Perceptron |
| KL-divergence: Exponentiated Gradient Descent | Hinge Loss | Normalized Winnow |

# Generalizing PCA to the Exponential Family

(Collins/Dasgupta/Schapire, NIPS 2001)

- PCA: Given data matrix $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$, find an orthogonal basis $V \in \mathbb{R}^{m \times k}$ and a projection matrix $A \in \mathbb{R}^{k \times n}$ that solve:

$$\min_{A,V} \sum_{i=1}^{n} ||\mathbf{x}_i - V\mathbf{a}_i||^2$$

  - Equivalent to maximizing likelihood when $\mathbf{x}_i \sim Gaussian(V\mathbf{a}_i, \sigma^2)$

- Generalized PCA: For any specified exponential family $p_{(\psi,\boldsymbol{\theta})}$, find $V \in \mathbb{R}^{m \times k}$ and $A \in \mathbb{R}^{k \times n}$ that maximize the data likelihood, i.e.,

$$\max_{A,V} \sum_{i=1}^{n} \log \left(p_{(\psi,\boldsymbol{\theta}_i)}(\mathbf{x}_i)\right) \quad \text{where} \quad \boldsymbol{\theta}_i = V\mathbf{a}_i, [i]_1^n$$

  - Bregman divergence formulation: $\min_{A,V} \sum_{i=1}^{n} d_\phi(\mathbf{x}_i, \nabla\psi(V\mathbf{a}_i))$

# Uniting Adaboost, Logistic Regression

(Collins/Schapire/Singer, *Machine Learning*, 2002)

Boosting: minimize exponential loss; sequential updates

Logistic Regression: min. log-loss; parallel updates

- Both are special cases of a classical Bregman projection problem: Find $\mathbf{p} \in S = \mathrm{dom}(\phi)$ that is "closest" in Bregman divergence to a given vector $\mathbf{q}_0 \in S$ subject to certain linear constraints:

$$\min_{\mathbf{p} \in S:\ A\mathbf{p} = A\mathbf{p}_0} d_\phi(\mathbf{p}, \mathbf{q}_0)$$

Boosting: I-divergence;

LR: binary relative entropy

# Implications

- convergence proof for Boosting
- parallel versions of Boosting algorithms
- Boosting with [0,1] bounded weights
- Extension to multi-class problems
- ....

# Misc. Work on Bregman Divergences

- *Duality results and auxiliary functions for the Bregman projection problem*. Della Pietra, Della Pietra and Lafferty
- *Learning latent variable models using Bregman divergences.* Wang and Schuurmans
- *U-Boost: Boosting with Bregman divergences.* Murata et al.

# Historical Reference

- L. M. Bregman. “The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming.” *USSR Computational Mathematics and Physics*, 7:200-217, 1967.
  - Problem:

    $$\min \varphi(x) \qquad \text{subject to} \qquad \mathbf{a_i^T x = b_i}, \;\; \mathbf{i = 0, \ldots, m-1}$$

  - Iterative procedure:
    1. Start with $\mathbf{x^{(0)}}$ that satisfies $\nabla\varphi(\mathbf{x^0}) = -\mathbf{A^T}\pi$. Set $t = 0$.
    2. Compute $\mathbf{x^{(t+1)}}$ to be the “Bregman” projection of $\mathbf{x^{(t)}}$ onto the hyperplane $\mathbf{a_i^T x = b_i}$, where $i = t \mod m$. Set $t = t + 1$ and repeat.
  - Converges to globally optimal solution. This cyclic projection method can be extended to halfspace and convex constraints, where each projection is followed by a correction).
- Censor and Lent (1981) coined the term “Bregman distance”

# Bertinoro Challenge

- What other learning formulations can be generalized with BDs?
- Given a problem/application, how to find the “best” Bregman divergence to use?
- Examples of unusual but practical Bregman Divergences?
- Other useful families of loss functions

# References


**BGW'05** On the optimality of conditional expectation as a Bregman predictor IEEE Trans. Info Theory, July 2005

**BMDG'05** Clustering with Bregman Divergences Journal of Machine Learning Research (JMLR, Oct05; SDM04)

**BDGM'04** An Information Theoretic Analysis of Maximum Likelihood Mixture Estimation for Exponential Families ICML, 2004

**BDGMM'04** A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation (KDD), 2004

# Backups

# The Exponential Family

- Definition: A multivariate parametric family with density

$$p_{(\psi,\boldsymbol{\theta})}(\mathbf{x}) = \exp\{\langle \mathbf{x}, \boldsymbol{\theta} \rangle - \psi(\boldsymbol{\theta})\} p_0(\mathbf{x})$$

- $\psi$ is the cumulant or log-partition function
- $\psi$ uniquely determines a family
  - Examples: Gaussian, Bernoulli, Multinomial, Poisson
- $\boldsymbol{\theta}$ fixes a particular distribution in the family
- $\psi$ is a strictly convex function

# Online Learning (Warmuth & Co)

- Setting: For trials $t = 1, \cdots, T$ do
  - Predict target $\hat{y}_t = g(\mathbf{w}_t \cdot \mathbf{x}_t)$ for instance $\mathbf{x}_t$ using link function $g$
  - Incur loss $L_t^{curr}(\mathbf{w}_t)$ (depends on true $y_t$ and predicted $\hat{y}_t$)
  - Update $\mathbf{w}_t$ to $\mathbf{w}_{t+1}$ using past history and the current trial
- Update Rule: $\mathbf{w}_{t+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \left( \underbrace{L^{hist}(\mathbf{w})}_{Deviation\ from\ history} + \eta_t \underbrace{L_t^{curr}(\mathbf{w})}_{Current\ loss} \right)$
  - When $L^{hist}(\mathbf{w}) = d_F(\mathbf{w}, \mathbf{w}_t)$, i.e., a Bregman loss function and $L_t^{curr}(\mathbf{w})$ is convex, the update rule reduces to

$$\mathbf{w}_{t+1} = f^{-1}\left(f(\mathbf{w}_t) + \eta_t \nabla L_t^{curr}(\mathbf{w}_t)\right) \text{ where } f = \nabla F$$

  - Further, if $L_t^{curr}(\mathbf{w})$ is the link Bregman loss $d_G(\mathbf{w} \cdot \mathbf{x}_t, g^{-1}(y_t))$

$$\mathbf{w}_{t+1} = f^{-1}\left(f(\mathbf{w}_t) + \eta_t (g(\mathbf{w} \cdot \mathbf{x}_t) - y_t)\mathbf{x}_t\right) \text{ where } g = \nabla G$$

# Examples and Bounds

| History loss:Update family | Current loss | Algorithm |
|---|---|---|
| Squared Loss: Gradient Descent | Squared Loss | Widrow Hoff(LMS) |
| Squared Loss: Gradient Descent | Hinge Loss | Perceptron |
| KL-divergence: Exponentiated Gradient Descent | Hinge Loss | Normalized Winnow |

- Regret Bounds: For a convex loss $L^{curr}$ and a Bregman loss $L^{hist}$

$$L_{alg} \leq \min_{\mathbf{w}} \left( \sum_{t=1}^{T} L_t^{curr}(\mathbf{w}) \right) + \text{ constant},$$

where $L_{alg} = \sum_{t=1}^{T} L_t^{curr}(\mathbf{w}_t)$ is the total algorithm loss

# Uniting Adaboost, Logistic Regression

(Collins/Schapire/Singer, *Machine Learning*, 2002)

Task: Learn target function $y(x)$ from labeled data $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ s.t. $y_i \in \{+1, -1\}$

| Boosting | Logistic Regression |
|---|---|
| Weak hypotheses $\{h_1(x), \cdots, h_m(x)\}$ | Features $\{h_1(x), \cdots, h_m(x)\}$ |
| Predicts $\hat{y}(x) = \text{sign}(\sum_{j=1}^{m} w_j h_j(x))$ | (same) |
| Minimizes Exponential Loss | Minimizes Log Loss |
| $\sum_{i=1}^{n} \exp(-y_i(\mathbf{w} \cdot \mathbf{h}(x)))$ | $\sum_{i=1}^{n} \log(1 + \exp(-y_i(\mathbf{w} \cdot \mathbf{h}(x))))$ |
| Sequential updates | Parallel updates |

- Both are special cases of a classical Bregman projection problem

# Bregman Projection Problem

Find $\mathbf{p} \in S = \mathrm{dom}(\phi)$ that is "closest" in Bregman divergence to a given vector $\mathbf{q}_0 \in S$ subject to certain linear constraints:

$$\min_{\mathbf{p} \in S:\ A\mathbf{p} = A\mathbf{p}_0} d_\phi(\mathbf{p}, \mathbf{q}_0)$$

- Ada-Boost: $S = \mathbb{R}^n_+,\ \mathbf{p}_0 = \mathbf{0},\ \mathbf{q}_0 = \mathbf{1},\ A = [y_1\mathbf{h}(x_1), \cdots, y_n\mathbf{h}(x_n)]$ and

$$d_\phi(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} \left( p_i \log\left(\frac{p_i}{q_i}\right) - p_i + q_i \right)$$

- Logistic Regression: $S = [0,1]^n,\ \mathbf{p}_0 = \mathbf{0},\ \mathbf{q}_0 = \frac{1}{2}\mathbf{1},\ A = [y_1\mathbf{h}(x_1), \cdots, y_n\mathbf{h}(x_n)]$ and

$$d_\phi(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} \left( p_i \log\left(\frac{p_i}{q_i}\right) + (1 - p_i) \log\left(\frac{1 - p_i}{1 - q_i}\right) \right)$$

- Optimal combining weights $\mathbf{w}^*$ can be obtained from the minimizer $\mathbf{p}^*$