A Projection Algorithm Based on the Pythagorian Theorem and Its Applications

Shotaro Akaho[1], Hideitsu Hino[2], Neneka Nara[3], Ken Takano[3], Noboru Murata[3]
[1] AIST, [2] University of Tsukuba, [3] Waseda University

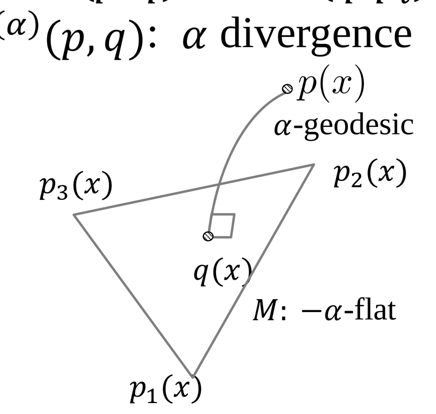
Overview

- A simple and robust method to find α-projection is proposed, which only uses values of α divergence
- Application 1 (m-projection to e-flat subspace) : Transfer learning, in which we have only small number of data for a target task while many data are available for similar tasks
- Application 2 (e-projection to m-flat subspace) : Nonnegative matrix factorization, dimension reduction for frequency data

Flat subspace and Pythagorian theorem

Pythagorian theorem[1]
$-\alpha$-flat submanifold $M = \{\sum \theta_i p_i | \sum \theta_i = 1\}$,
p_i: $-\alpha$ affine coordinate of $p_i(x)$

If α-geodesic connecting p and $q \in M$ is orthogonal to M,
$r_i = D^{(\alpha)}(p,q) + D^{(\alpha)}(q,p_i) - D^{(\alpha)}(p,p_i) = 0$,
$D^{(\alpha)}(p,q)$: α divergence

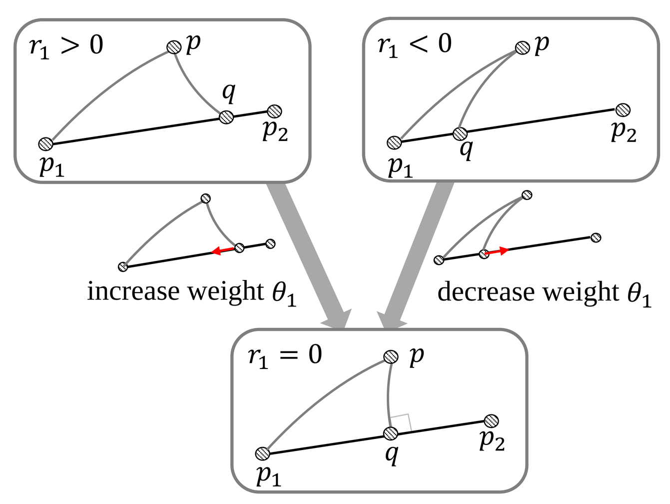


Exponential and mixture
- Important two cases of α are $\alpha = \pm 1$,
  e(xponential) for $\alpha = 1$,
  m(ixture) for $\alpha = -1$
- ± 1(e and m) divergence is Kullback-Leibler divergence
  $D^{(m)}(p,q) = D^{(e)}(q,p) = \int p(x) \log \frac{p(x)}{q(x)} dx$

Divergence-based projection algorithm

An algorithm to find the α-projection q

Basic idea
If r_i is larger than zero, q should be closer to p_i, and
if r_i is smaller than zero, q should be more distant from p_i



Here we assume $\theta_i \geq 0$ for simplicity
Algorithm
1. Initialize $\theta_i^{(0)}$
2. Repeat the following for $t = 0,1,2,\ldots$ until convergence
   1. $q := \sum \theta_i^{(t+1)} p_i$
   2. Calculate $r_i = D^{(\alpha)}(p,q) + D^{(\alpha)}(q,p_i) - D^{(\alpha)}(p,p_i)$
   3. Update θ_i by $\theta_i^{(t+1)} := \theta_i^{(t)} f(r_t)$
   4. Normalize $\theta_i^{(t+1)}$

f is a monotonically increasing function
s.t. $f(x) > 0, f(0) = 1$ e.g. $f(x) = 2/(1+\exp(-\beta x))$

Properties of the algorithm
- Simple
- Only dependent on values of divergence
- Robust (if divergence is calculated robustly)

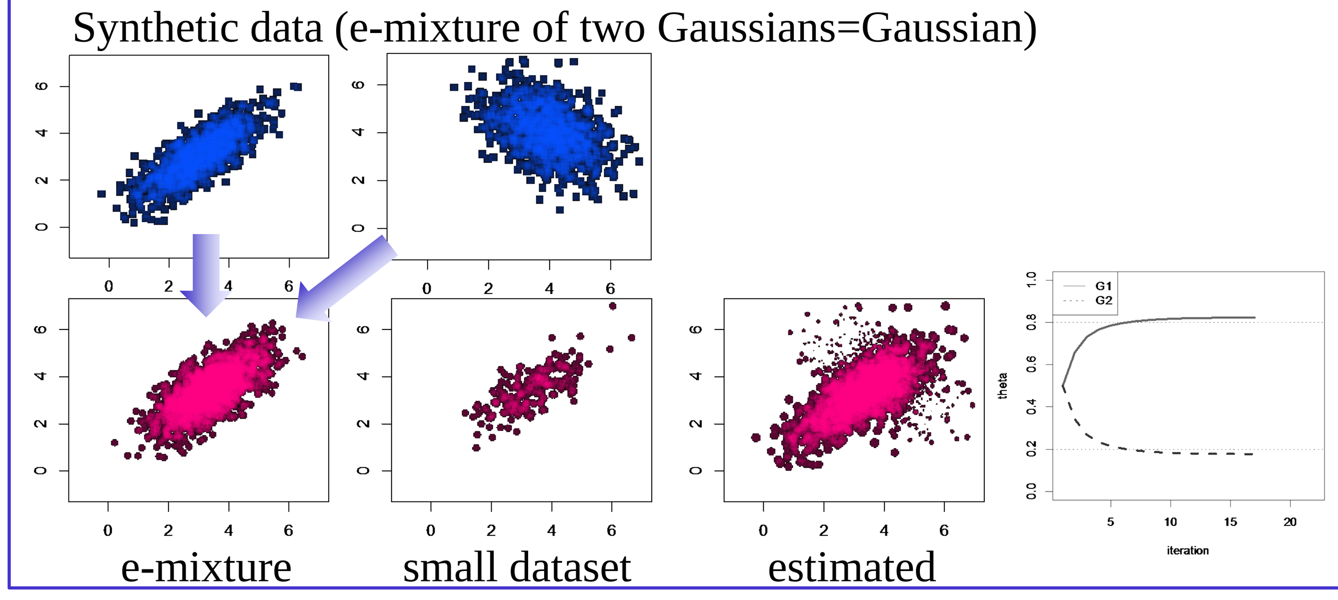
Application 1: Transfer learning by nonparametric e-mixture estimation

Transfer learning
a framework of machine learning, where performance of a certain learning task is improved by using other (similar) tasks. Here, the target empirical distribution is projected onto a subspace spanned by other distributions.
e-mixture (cf. m-mixture)

$$p_e(x;\theta) = \exp\left(\sum \theta_i \log p_i(x) - b(\theta)\right), \quad \sum \theta_i = 1$$

The e-mixture satisfies the maximum entropy principle
The problem is to find the m-projection from a target distribution $p(x)$ to an e-flat submanifold spanned by $\log p_i(x)$, which can be optimized by the divergence based projection algorithm.
Characterization of e-mixture based on divergence[2]
e-mixture is characterized by divergence $p_e(x;\theta) = \arg\min_q \sum \theta_i D^{(m)}(q,p_i)$
Nonparametric extension
Target $p(x)$ and auxiliary distributions $p_i(x)$ are all given by empirical distribution
m-representation
Since e-mixture for empirical distributions are not well-defined, so we use the characterization of e-mixture as its definition. The distribution is represented by m-representation $q(x) = \sum w_j \delta(x - x_j)$, $\sum w_j = 1$
Nonparametric estimation of divergence[3]
We use a nearest neighbor based method to estimate between two (weighted) empirical distributions
Nonparametric e-mixture Algorithm
1. Initialize θ_i
2. Repeat the following until convergence
   1. Obtain m-representation w_j to satisfy the characterization of e-mixture by fixing θ_i
   2. Estimate divergence $D^{(m)}(q,p_i)$,
   3. update θ_i by the divergence-based algorithm

Experiments
Synthetic data (e-mixture of two Gaussians=Gaussian)

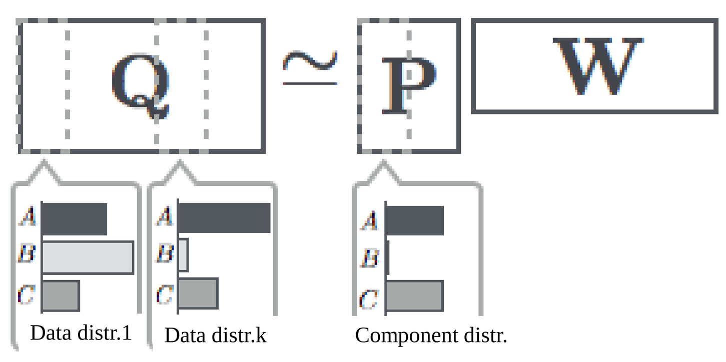


EEG data (5 subjects, each one is examined with small data)

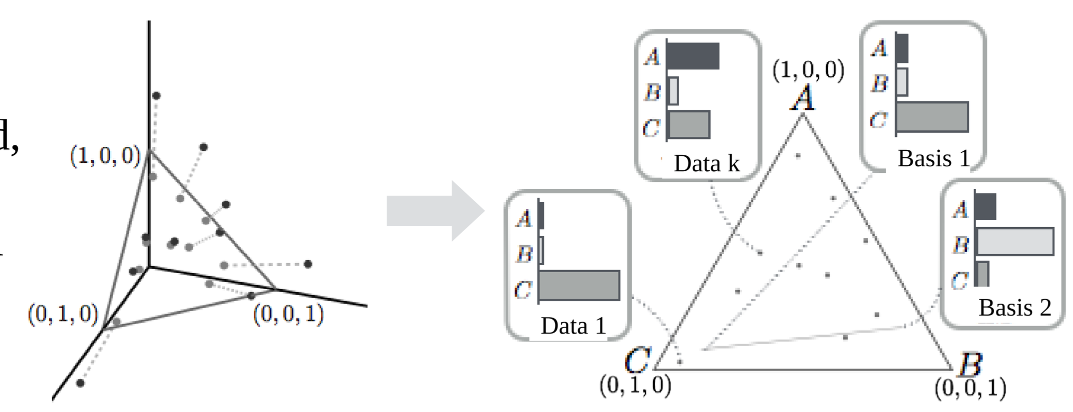
subjects/ method	i	ii	iii	iv	v
small	37.22 ±8.67	33.73 ±12.03	**29.05** ±12.27	41.27 ±9.81	40.87 ±4.38
uniform	36.03 ±11.70	31.35 ±12.37	39.68 ±9.86	40.63 ±6.22	37.22 ±4.29
reg.	31.51 ±9.79	**21.59** ±9.47	31.83 ±11.22	30.08 ±11.90	29.44 ±12.40
e-mixture	**29.12** ±9.07	23.57 ±10.97	35.71 ±10.77	**30.00** ±12.28	**27.22** ±10.04

Application 2: Nonnegative matrix factorization

Nonnegative matrix factorization
- Dimension reduction in the space of positive value matrix $Q \cong PW$
- Using column-wise normalization operator Π, $\Pi[Q] \cong \Pi[P]\Pi[W]$
- This model is called "topic models" in machine learning community, in particular natural language processing (pLSA, LDA, etc.)
  [documents-words] = [documents-topics] x [topics-words]
- P: basis vectors, W: coefficients vectors



Optimization criterion
- The problem is to find m-flat subspace M spanned by P
- Ordinary pLSA optimizes parameters by max likelihood, which is equivalent to m-projection
- However, e-projection is more natural from geometrical viewpoint[4]
- Resulting optimization problem is
  $\min_{P,W} \sum D^{(e)}(q_j, \hat{q}_j)$, $\hat{q}_j \in M$ is a projection of q_j
- Alternating optimization algorithm
  Repeat the following two steps (e-projection to m-flat subspace) until convergence
  1. Optimize P with fixing W
  2. Optimize W with fixing P



Experiments
Comparison with existing method[5] by synthetic data (50x4→50x3)

Number of improvements	199/200
Reduced error ratio[%]	2.34

References

1. Amari, S. (1985) *Differential-Geometrical Methods in Statistics*, Springer
2. Murata, N., Fujimoto, Y. (2009) Bregman divergence and density integration, *Journal of Math for Industry*, 1, 97-104
3. Hino, H., Murata, N. (2013) Information estimators for weighted observations, *Neural Networks*, 1, 260-275
4. Akaho, S. (2004) The e-PCA and m-PCA: dimension reduction of parameters by information geometry, In *Proc. of IJCNN*, 129-134
5. Sra, S., Dhillon, I. S. (2005) Generalized nonnegative matrix approximations with Bregman divergences. In *NIPS*, 283-290