

Optimization Framework for Semi-supervised and Transfer Learning using Multiple Classifiers and Clusterers



Joydeep Ghosh

UT Austin

May 21, 2012

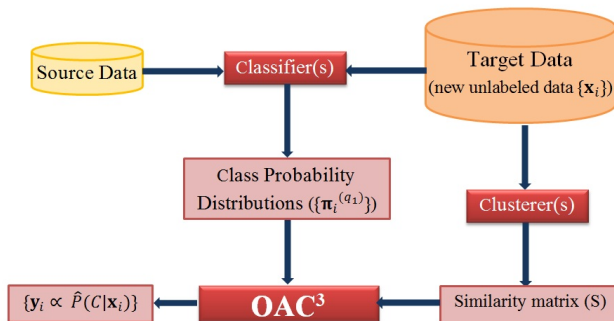
- From independent, diversified classifiers, the ensemble created is usually more accurate than its individual components.
- Cluster ensembles generate more accurate, robust, and stable results. In addition:
 - ① Perform distributed computing under privacy and sharing constraints;
 - ② Reuse existing knowledge.
- Combine classifier and cluster ensembles?

- Unsupervised models provide a variety of supplementary constraints for classifying new data.
- Similar new instances in the target set are more likely to share the same class label.

Applications?

- 1 Improve performance given weak classifiers/few labeled data.
- 2 Better handle concept drift.
- 3 Semi-supervised and transfer learning.

Optimization Algorithm for Combining Classifiers and Clusterers



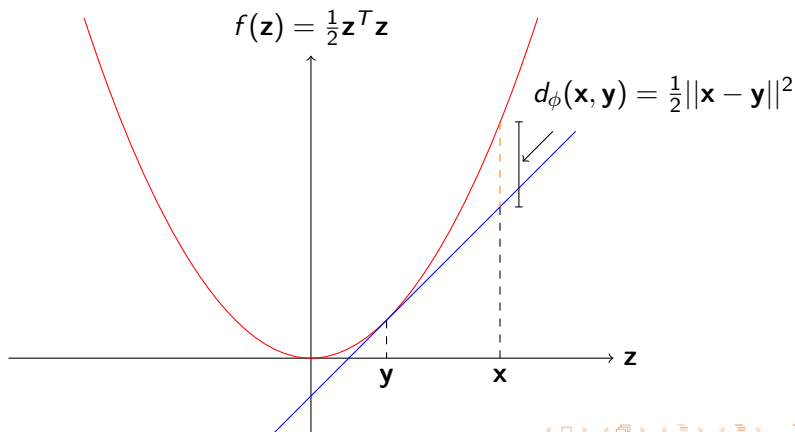
Formulation of the problem

Objective function: $J = \sum_{i \in \mathcal{X}} \mathcal{L}(\boldsymbol{\pi}_i, \mathbf{y}_i) + \alpha \sum_{(i,j) \in \mathcal{X}} s_{ij} \mathcal{L}(\mathbf{y}_i, \mathbf{y}_j)$, where,

- $\boldsymbol{\pi}_i$ is the predicted class label of i^{th} instance from the classifiers.
- \mathbf{y}_i is the estimated (refined) class label of i^{th} instance.
- s_{ij} is the similarity between i^{th} and j^{th} instances.
- \mathcal{L} can be any loss function but we concentrate on some specific **Bregman Divergences**.
- Minimize J over $\{\mathbf{y}_i\}_{i=1}^n$.

Illustration of Bregman divergence

- Let $\phi : \mathcal{S} \rightarrow \mathbb{R}$ be a differentiable, strictly convex function of Legendre type ($\mathcal{S} \subseteq \mathbb{R}^d$).
- The Bregman Divergence $d_\phi : \mathcal{S} \times \text{ri}(\mathcal{S}) \rightarrow \mathbb{R}$ is defined as $d_{\phi}(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle (\mathbf{x} - \mathbf{y}), \nabla \phi(\mathbf{y}) \rangle$.



Examples of jointly convex Bregman divergence

Domain	$\phi(\mathbf{p})$	$d_\phi(\mathbf{p}, \mathbf{q})$	Divergence
\mathbb{R}	p^2	$(p - q)^2$	Squared Loss
$[0, 1]$	$p \log(p) + (1 - p) \log(1 - p)$	$p \log(\frac{p}{q}) + (1 - p) \log(\frac{1-p}{1-q})$	Logistic Loss
\mathbb{R}_+	$p \log(p) - (1 + p) \log(1 + p)$	$p \log(\frac{p}{q}) - (1 + p) \log(\frac{1+p}{1+q})$	Bose-Einstein Entropy
\mathbb{R}_{++}	$-\log(p)$	$\frac{p}{q} - \log(\frac{p}{q}) - 1$	Itakura-Saito Distance
\mathbb{R}^k	$\ \mathbf{p}\ ^2$	$\ \mathbf{p} - \mathbf{q}\ ^2$	Squared Euclidean Distance
k -simplex	$\sum_{i=1}^k p_i \log_2(p_i)$	$\sum_{i=1}^k p_i \log_2(\frac{p_i}{q_i})$	KL-Divergence
\mathbb{R}_+^k	$\sum_{i=1}^k p_i \log(p_i)$	$\sum_{i=1}^k p_i \log(\frac{p_i}{q_i}) - \sum_{i=1}^k (p_i - q_i)$	Generalized I-Divergence

Theorem ([3])

Let Y be a random variable that takes values in $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^n \subset \mathcal{S} \subseteq \mathbb{R}^k$ following a probability measure ν such that $\mathbb{E}_\nu[Y] \in \text{ri}(\mathcal{S})$. Given a Bregman divergence $d_\phi: \mathcal{S} \times \text{ri}(\mathcal{S}) \rightarrow [0, \infty)$, the optimization problem $\min_{\mathbf{s} \in \text{ri}(\mathcal{S})} \mathbb{E}_\nu[d_\phi(Y, \mathbf{s})]$ has a unique minimizer given by $\mathbf{s}^* = \boldsymbol{\mu} = \mathbb{E}_\nu[Y]$.

- Single best representative of a set of vectors is simply the expectation of this set provided the divergence is computed with this representative as the 2nd argument.
- However, no simple form of optimal solution exists if the variable to be optimized occurs as the 1st argument.

Solution in the Legendre dual space

- Work in the Legendre dual space - the optimal solution has a simple form.
- Legendre Dual: $\psi(\mathbf{y}) = \langle \mathbf{y}, \nabla \phi^{-1}(\mathbf{y}) \rangle - \phi(\nabla \phi^{-1}(\mathbf{y}))$
- $d_{\phi}(\mathbf{Y}, \mathbf{s}) = d_{\psi}(\nabla \phi(\mathbf{s}), \nabla \phi(\mathbf{Y}))$.
- Apply previous theorem in the Legendre dual space, compute the optimal value and project the solution back to primal space.
Projection is one-to-one because of strict convexity of ϕ and ψ .

- Original: $J = \sum_{i \in \mathcal{X}} \mathcal{L}(\boldsymbol{\pi}_i, \mathbf{y}_i) + \alpha \sum_{(i,j) \in \mathcal{X}} s_{ij} \mathcal{L}(\mathbf{y}_i, \mathbf{y}_j).$
- Modified: $J = \sum_{i \in \mathcal{X}} d_{\phi}(\boldsymbol{\pi}_i, \mathbf{y}_i) + \alpha \sum_{(i,j) \in \mathcal{X}} s_{ij} d_{\phi}(\mathbf{y}_i, \mathbf{y}_j).$
- However, the variables appear in both first and second argument of Bregman divergences – no closed form solution.

- Create two copies for each $\mathbf{y}_i : \mathbf{y}_i^{(l)}$ and $\mathbf{y}_i^{(r)}$;
- Let us review the objective function with the left and right copies inserted:

$$J = \sum_{i \in \mathcal{X}} d_{\phi}(\boldsymbol{\pi}_i, \mathbf{y}_i^{(r)}) + \alpha \sum_{(i^{(l)}, j^{(r)}) \in \mathcal{X}} s_{i^{(l)}j^{(r)}} d_{\phi}(\mathbf{y}_i^{(l)}, \mathbf{y}_j^{(r)})$$

$$J_{[\mathbf{y}_i]} = d_{\phi}(\boldsymbol{\pi}_i, \mathbf{y}_i^{(r)}) + \alpha \left[\sum_{j^{(r)} \in \mathcal{X}} s_{i^{(l)}j^{(r)}} d_{\phi}(\mathbf{y}_i^{(l)}, \mathbf{y}_j^{(r)}) + \sum_{j^{(l)} \in \mathcal{X}} s_{j^{(l)}i^{(r)}} d_{\phi}(\mathbf{y}_j^{(l)}, \mathbf{y}_i^{(r)}) \right]$$

- Additional constraint is required to ensure that the two copies remain close – penalty term.

- Copies are updated iteratively (Alternating Minimization).
- **Inputs:** $\{\pi_i\}$, **S.** **Output:** $\{\mathbf{y}_i\}$.
- Initialize $\{\mathbf{y}_i^{(r)}\}, \{\mathbf{y}_i^{(l)}\}$ so that $\mathbf{y}_{i\ell}^{(r)} = \mathbf{y}_{i\ell}^{(l)} = \frac{1}{k} \forall i \in \{1, 2, \dots, n\}, \forall \ell \in \{1, 2, \dots, k\}$.
- Primal Space:

$$\min_{\mathbf{y}_j^{(r)}} \left[d_\phi(\pi_j^{(r)}, \mathbf{y}_j^{(r)}) + \alpha \sum_{i^{(l)} \in \mathcal{X}} s_{i^{(l)}j^{(r)}} d_\phi(\mathbf{y}_i^{(l)}, \mathbf{y}_j^{(r)}) + \lambda_j^{(r)} d_\phi(\mathbf{y}_j^{(l)}, \mathbf{y}_j^{(r)}) \right],$$

- Dual Space:

$$\min_{\nabla \phi(\mathbf{y}_i^{(l)})} \left[\alpha \sum_{j^{(r)} \in \mathcal{X}} s_{i^{(l)}j^{(r)}} d_\psi(\nabla \phi(\mathbf{y}_j^{(r)}), \nabla \phi(\mathbf{y}_i^{(l)})) + \lambda_i^{(l)} d_\psi(\nabla \phi(\mathbf{y}_i^{(r)}), \nabla \phi(\mathbf{y}_i^{(l)})) \right].$$

- Finally, set $\mathbf{y}_i = (\mathbf{y}_i^{(l)} + \mathbf{y}_i^{(r)})/2$.

- Computation of $\{\pi_i\}_{i=1}^n$ requires $O(nr_1k)$.
- Computing similarity matrix is $O(r_2n^2)$.
- **OAC**³: computational cost per iteration is $O(kn^2)$.
- Where n = number of instances in the target set k = number of class labels r_1 = number of components of the classifier ensemble r_2 = number of components of the cluster ensemble.
- Compare this with any line search or trust region method or the cubic time complexity of the nearest method **BGCM** [4]! A detailed comparison is available in [2].
- Update of left and right copies can be parallelized over instances.

- Convergence is guaranteed for Bregman divergences with some special properties (see Table 7 for examples).
- Rate of convergence of **OAC**³ is linear at least for squared Euclidean distance, KL divergence and generalized I divergence [2].

Half-moon Data

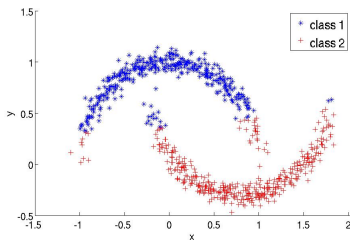


Figure: Classifier Output

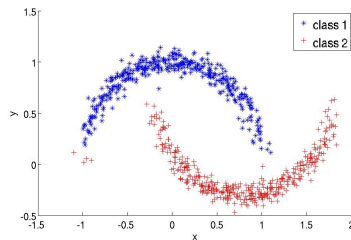


Figure: OAC³ Output

Pedagogical example continued...

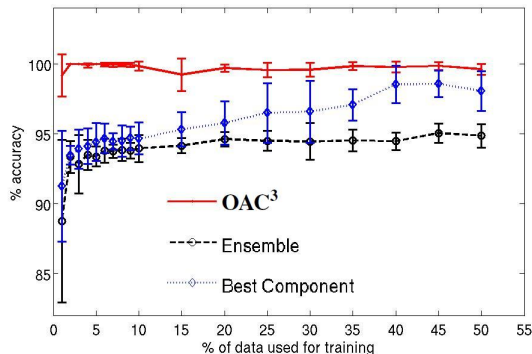


Figure: Average Accuracies and Standard Deviations for Halfmoon Data.

Semi-supervised learning on text data

- MCLA: Meta CLustering Algorithm,
- HBGF: Hybrid Bipartite Graph Formulation,
- BGCM: Bipartite Graph-based Consensus Maximization.

Method	News1	News2	News3	News4	News5	News6	Cora1	Cora2	Cora3	Cora4	DBLP
M ₁	79.67	88.55	85.57	88.26	87.65	88.80	77.45	88.58	86.71	88.41	93.37
M ₂	77.21	86.11	81.34	86.76	83.58	85.63	77.97	85.94	85.08	88.79	87.66
M ₃	80.56	87.96	86.58	89.83	87.16	90.20	77.79	88.33	86.46	88.13	93.82
M ₄	77.70	85.71	81.49	84.67	85.43	85.78	74.76	85.94	78.10	90.16	79.49
MCLA	75.92	81.73	82.53	86.86	82.95	85.46	87.03	83.88	88.92	87.16	89.53
HBGF	81.99	92.44	88.11	91.52	89.91	91.25	78.34	91.11	84.81	89.43	93.57
BGCM	81.28	91.01	86.08	91.25	88.64	90.88	86.87	91.55	89.65	90.90	94.17
OAC ³	85.01	93.64	89.64	93.80	91.22	92.59	88.54	90.79	90.60	91.49	94.38

Table: Comparison of **OAC³** with Other Algorithms – Classification Accuracies (Best Results in Boldface) [1].

S^3VM : Semi-Supervised Support Vector Machines

Dataset	$ \mathcal{X} $	Ensemble	Best Component	S^3VM	BGCM	OAC ³
Half-moon(2%)	784	92.53(± 1.83)	93.02(± 0.82)	99.61(± 0.09)	92.16(± 1.47)	99.64 (± 0.08)
Circles(2%)	1568	60.03(± 8.44)	95.74(± 5.15)	54.35(± 4.47)	78.67(± 0.54)	99.61 (± 0.83)
Pima(2%)	745	68.16(± 5.05)	69.93(± 3.68)	61.67(± 3.01)	69.21(± 4.83)	70.31 (± 4.44)
Heart(7%)	251	77.77(± 2.55)	79.22(± 2.20)	77.07(± 4.77)	82.78(± 4.82)	82.85 (± 5.25)
G. Numer(10%)	900	70.96(± 1.00)	70.19(± 1.52)	73.00(± 1.50)	73.70(± 1.06)	74.44 (± 3.44)
Wine(10%)	900	79.87(± 5.68)	80.37(± 5.47)	80.73(± 4.49)	75.37(± 13.66)	83.62 (± 6.27)

Table: Comparison of **OAC³** with **BGCM** and **S^3VM** — Average Accuracies \pm (Standard Deviations)

Transfer learning on text data

- WIN: Winnow,
- LR: Logistic Regression,
- TSVM: Transductive Support Vector Machine.

Dataset	Mode	WIN	LR	SVM	Ensemble	TSVM	LWE	OAC ³
20 Newsgroup	C vs S	66.61	67.17	67.02	69.58	76.97	77.07	91.25
	R vs T	60.43	68.79	63.87	65.98	89.95	87.46	90.11
	R vs S	80.11	76.51	71.40	77.39	89.96	87.81	92.90
	S vs T	73.93	72.16	71.51	75.11	85.59	81.99	91.83
	C vs R	89.00	77.36	81.50	85.18	89.64	91.09	93.75
	C vs T	93.41	91.76	93.89	93.48	88.26	98.90	98.70
Reuters-21758	O vs Pe	70.57	66.19	69.25	73.30	76.94	76.77	80.97
	O vs Pl	65.10	67.87	69.88	69.21	70.08	67.59	68.91
	Pe vs Pl	56.75	56.48	56.20	57.59	59.72	59.90	67.46
Spam	spam 1	79.15	56.92	66.28	68.64	76.92	65.60	80.29
	spam 2	81.15	59.76	73.15	75.07	84.92	73.36	87.05
	spam 3	88.28	64.43	78.71	81.87	90.79	93.79	91.27

Table: Classification of *20 Newsgroup*, *Reuters-21758* and *Spam* Data (Best Results in Boldface).

Real world challenge – hyper-spectral data



Figure: Botswana May 2001



Figure: Botswana June 2001



Figure: Botswana July 2001

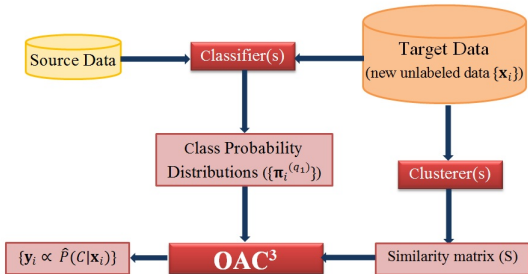
Results on hyper-spectral data

- NBW: Naïve Bayes Wrapper,
- ML: Maximum Likelihood Classifier.

Dataset	Original to Target	NBW	NBW+OAC ³	ML	ML+OAC ³	α	λ	PCs
Area 1	may to june	70.68	72.61 (± 0.42)	74.47	81.93 (± 0.52)	0.0010	0.1	9
	may to july	61.85	63.11 (± 0.29)	58.58	64.32 (± 0.53)	0.0001	0.2	12
	june to july	70.55	72.47 (± 0.17)	79.71	80.06 (± 0.26)	0.0012	0.1	127
	may+june to july	75.53	80.53 (± 0.31)	85.78	85.91 (± 0.23)	0.0008	0.1	123
Area 2	may to june	66.10	71.02 (± 0.28)	70.22	81.48 (± 0.43)	0.0070	0.1	9
	may to july	61.55	63.74 (± 0.14)	52.78	64.15 (± 0.22)	0.0001	0.2	12
	june to july	54.89	57.65 (± 0.53)	75.62	77.04 (± 0.37)	0.0060	0.1	80
	may+june to july	63.79	64.58 (± 0.16)	77.33	79.59 (± 0.23)	0.0040	0.1	122

Table: Transfer learning results on Botswana data

Objective function: $J = \sum_{i \in \mathcal{X}} \mathcal{L}(\boldsymbol{\pi}_i, \mathbf{y}_i) + \alpha \sum_{(i,j) \in \mathcal{X}} s_{ij} \mathcal{L}(\mathbf{y}_i, \mathbf{y}_j).$





A. Acharya, E. R. Hruschka, J. Ghosh, and S. Acharyya.
C³E: A Framework for Combining Ensembles of Classifiers and Clusterers.

In 10th Int. Workshop on MCS, 2011.



A. Acharya, E.R. Hruschka, J. Ghosh, and S. Acharyya.
Optimization framework for semi-supervised and transfer learning using multiple classifiers and clusterers.

Technical report, 2012.



A. Banerjee, S. Merugu, Inderjit S. Dhillon, and J. Ghosh.
Clustering with bregman divergences.

J. Mach. Learn. Res., 6:1705–1749, December 2005.



J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han.
Graph-based consensus maximization among multiple supervised and unsupervised models.

In Proc. of NIPS, pages 1–9, 2009.