Learning with Multiple Models and from Multiple Domains

Ayan Acharya

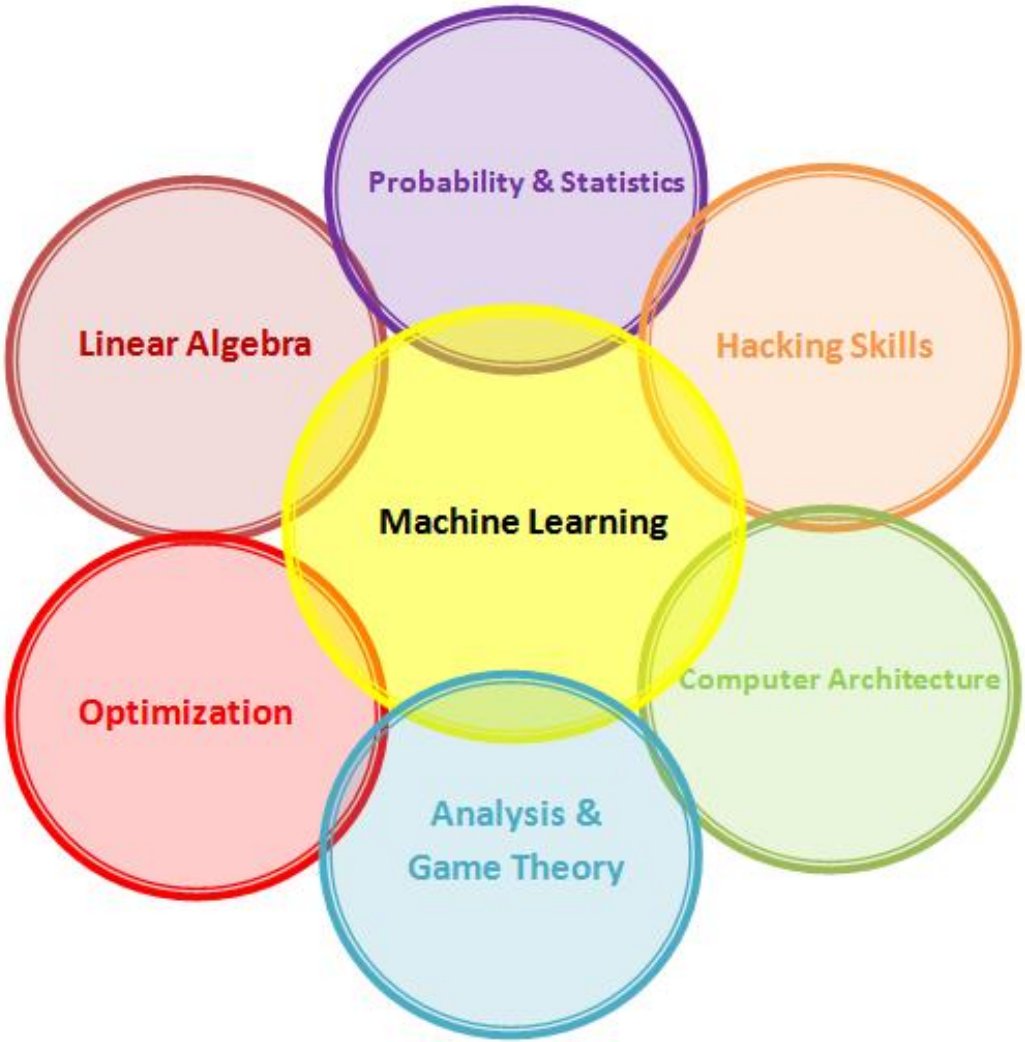
Dept. of ECE, UT Austin

April 5, 2013

Machine Learning Applications

- Web search ranking (Google, Yahoo!, Bing search engines).
- Fraud detection, spam filtering.
- Speech and object recognition.
- Stock market analysis.
- Recommender Systems (Amazon, NetFlix, eBay).
- Bioinformatics – DNA sequence classification.
- BIG Data Analytics – the new buzzword!

Tools used in Machine Learning



Types of Machine Learning Algorithms

- Supervised Learning – classification, regression.
- Unsupervised Learning.
- Semi-supervised Learning.
- Transfer, Multitask and Multi-view Learning.
- Others.

Ensemble Learning

- Complicated models can be built by combination of simple (or complex) models.
- Classifier Ensemble – combination of simple classifiers [Example – magic of AdaBoost].
- Clustering Ensemble – combination of simple clustering algorithms.
- How to combine the two sets of ensemble effectively and what could be possible applications?

Motivation

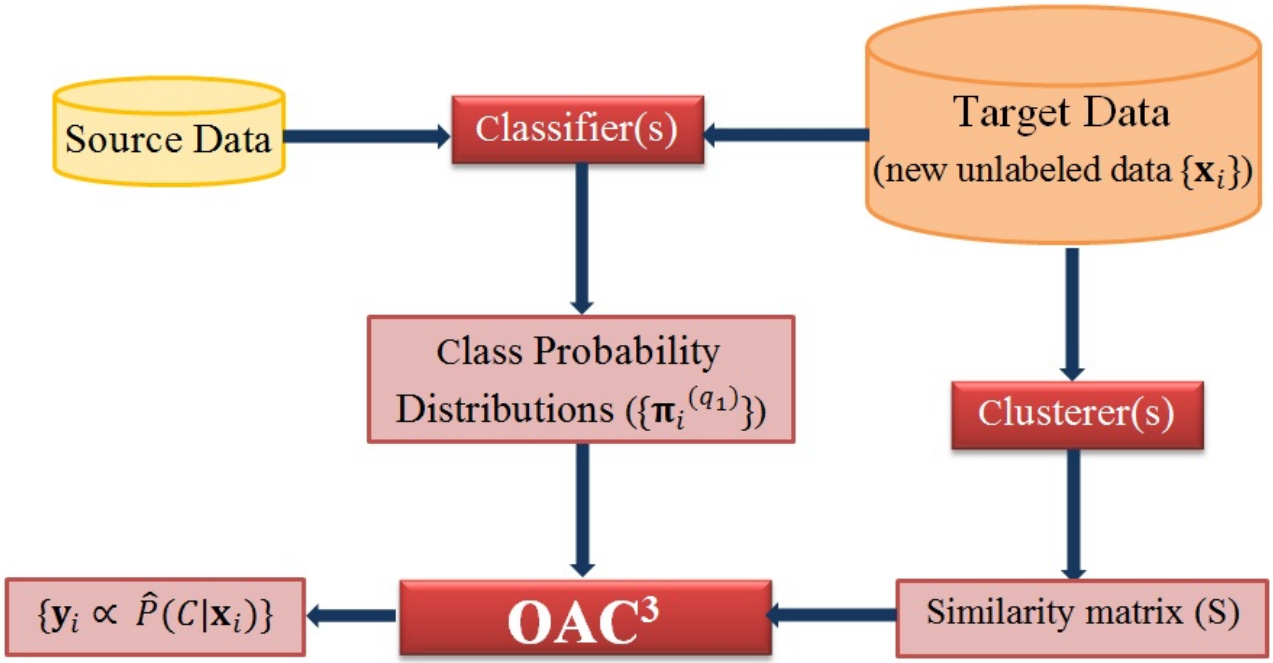
- Unsupervised models provide a variety of supplementary constraints for classifying new data.
- Similar new instances in the target set are more likely to share the same class label.

Applications?

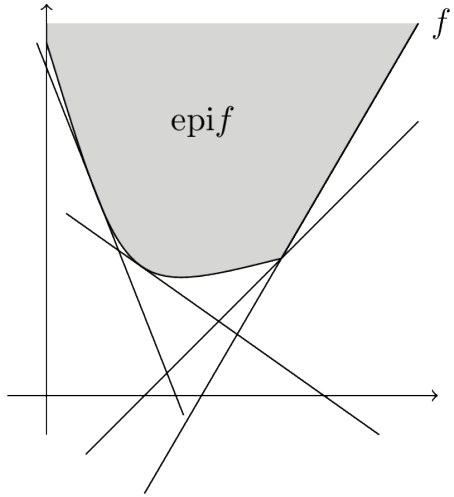
- Improve performance given weak classifiers/few labeled data.
- Better handle concept drift.
- Semi-supervised and transfer learning.

Overview of **OAC**^{3}

Optimization **A**lgorithm for **C**ombining **C**lassifiers and **C**lusterers



Convex Function and Dual Representation



- $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)\ \forall x_1, x_2 \in \mathcal{X}$ and $\lambda \in [0,1]$.
- $f^*(x^*) = \sup_{x \in \mathcal{X}} \{\langle x^*, x \rangle - f(x)\}$

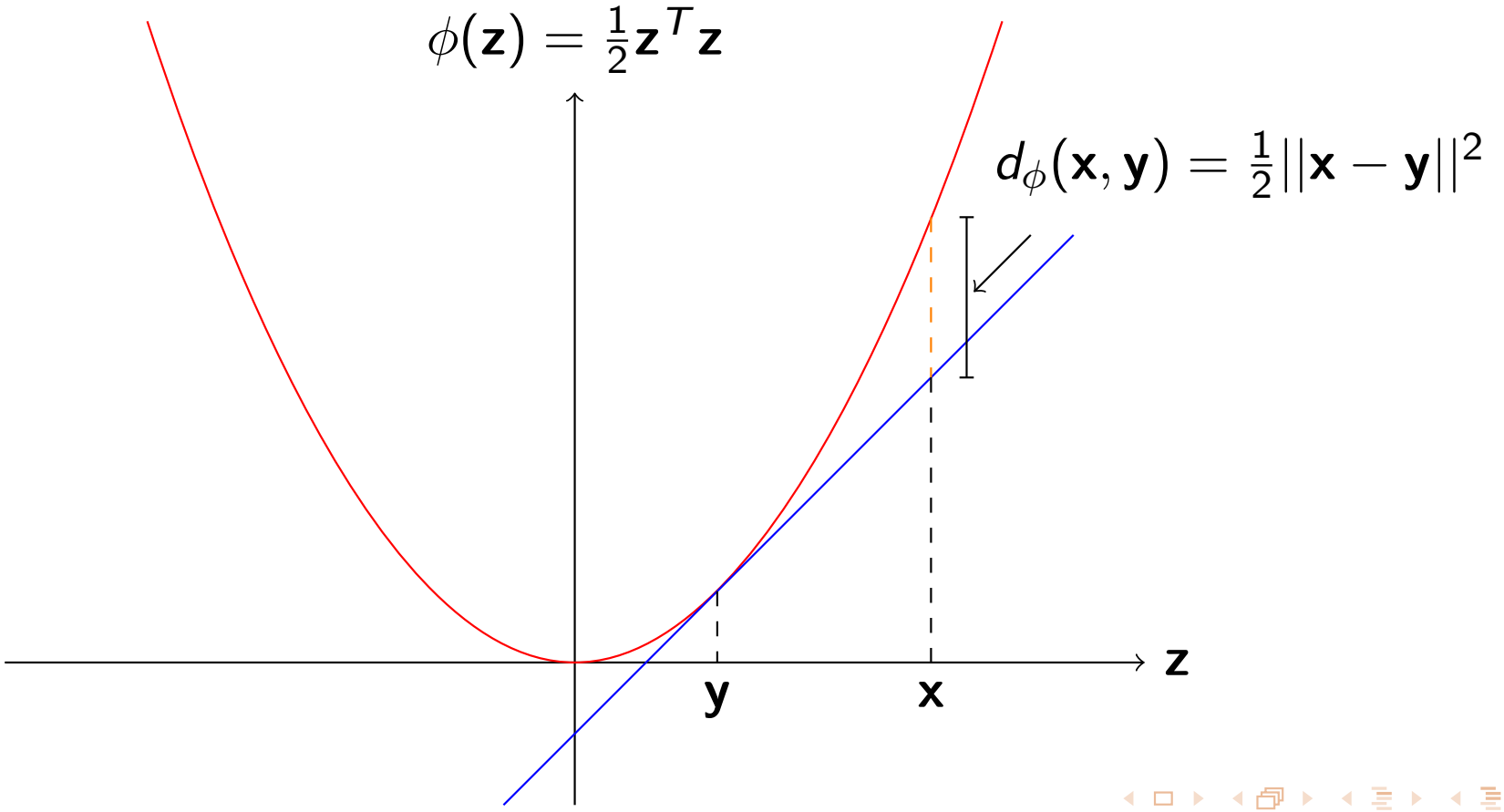
Formulation of the problem

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 $\phi : \mathcal{S} \to \mathbb{R}$ be a differentiable, strictly convex function of Legendre type ($\mathcal{S} \subseteq \mathbb{R}^d$).
- The Bregman Divergence $d_\phi : \mathcal{S} \times \text{ri}(\mathcal{S}) \to \mathbb{R}$ 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
$[0,1]$	$p\log(p)+(1-p)\log(1-p)$	$p\log\left(\frac{p}{q}\right)+(1-p)\log\left(\frac{1-p}{1-q}\right)$	Logistic Loss
$\mathbb{R}_+$	$p\log(p)-(1+p)\log(1+p)$	$p\log\left(\frac{p}{q}\right)-(1+p)\log\left(\frac{1+p}{1+q}\right)$	Bose-Einstein Entropy
$\mathbb{R}_{++}$	$-\log(p)$	$\frac{p}{q}-\log\left(\frac{p}{q}\right)-1$	Itakura-Saito Distance
$\mathbb{R}^k$	$\|\|\mathbf{p}\|\|^2$	$\|\|\mathbf{p}-\mathbf{q}\|\|^2$	Squared Euclidean Distance
k-simplex	$\sum_{i=1}^{k} p_i\log_2(p_i)$	$\sum_{i=1}^{k} p_i\log_2\left(\frac{p_i}{q_i}\right)$	KL-Divergence
$\mathbb{R}_+^k$	$\sum_{i=1}^{k} p_i\log(p_i)$	$\sum_{i=1}^{k} p_i\log\left(\frac{p_i}{q_i}\right)-\sum_{i=1}^{k}(p_i-q_i)$	Generalized I-Divergence

Remarkable property of Bregman divergences

Theorem

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 v such that $\mathbb{E}_v[Y] \in ri(\mathcal{S})$. Given a Bregman divergence $d_\phi : \mathcal{S} \times ri(\mathcal{S}) \to [0, \infty)$, the optimization problem $\min_{\mathbf{s} \in ri(\mathcal{S})} \mathbb{E}_v[d_\phi(Y, \mathbf{s})]$ has a unique minimizer given by $\mathbf{s}^* = \boldsymbol{\mu} = \mathbb{E}_v[Y]$.

- 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 2^{nd} argument.
- However, no simple form of optimal solution exists if the variable to be optimized occurs as the 1^{st} argument.

Solution in the Legendre dual space

- Work in the Legendre dual space - the optimal solution has a simple form.
- Legendre Dual: $\psi(\mathbf{y}) = \langle \mathbf{y}, \boldsymbol{\nabla}\phi^{-1}(\mathbf{y}) \rangle - \phi(\boldsymbol{\nabla}\phi^{-1}(\mathbf{y}))$.
- $d_\phi(\mathbf{Y}, \mathbf{s}) = d_\psi(\boldsymbol{\nabla}_\phi(\mathbf{s}), \boldsymbol{\nabla}_\phi(\mathbf{Y}))$.
- 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.

Objective Function in **OAC^3**

- 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.

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.

The algorithm – OAC^3

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

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

- Dual Space:

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

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

Time complexities

- Computation of $\{\boldsymbol{\pi}_i\}_{i=1}^{n}$ 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**!
- Update of left and right copies can be parallelized over instances.

Convergence guarantee

- Convergence is guaranteed for Bregman divergences with some special properties (see Table 11 for examples).
- Rate of convergence of **OAC^3** is linear at least for squared Euclidean distance, KL divergence and generalized I divergence.

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 **OAC^3** with Other Algorithms – Classification Accuracies (Best Results in Boldface).

Semi-supervised learning on UCI data

S^3VM: Semi-Supervised Support Vector Machines

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

Table: Comparison of **OAC^3** 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

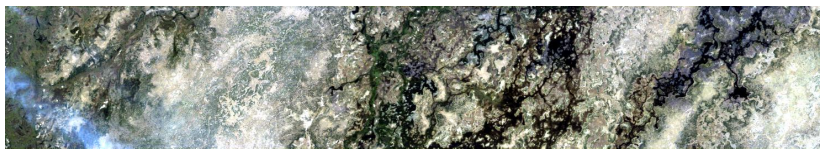


Figure: Botswana June 2001

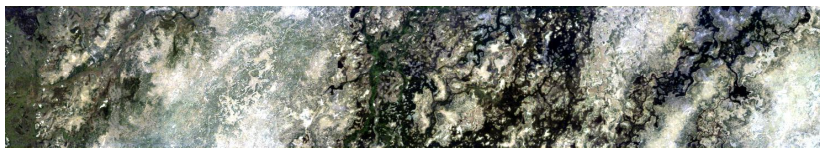


Figure: Botswana July 2001

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** ($\pm$0.42)	74.47	**81.93** ($\pm$0.52)	0.0010	0.1	9
	may to july	61.85	**63.11** ($\pm$0.29)	58.58	**64.32** ($\pm$0.53)	0.0001	0.2	12
	june to july	70.55	**72.47** ($\pm$0.17)	79.71	**80.06** ($\pm$0.26)	0.0012	0.1	127
	may+june to july	75.53	**80.53** ($\pm$0.31)	85.78	**85.91** ($\pm$0.23)	0.0008	0.1	123
Area 2	may to june	66.10	**71.02** ($\pm$0.28)	70.22	**81.48** ($\pm$0.43)	0.0070	0.1	9
	may to july	61.55	**63.74** ($\pm$0.14)	52.78	**64.15** ($\pm$0.22)	0.0001	0.2	12
	june to july	54.89	**57.65** ($\pm$0.53)	75.62	**77.04** ($\pm$0.37)	0.0060	0.1	80
	may+june to july	63.79	**64.58** ($\pm$0.16)	77.33	**79.59** ($\pm$0.23)	0.0040	0.1	122

Table: Transfer learning results on Botswana data

Multitask Learning

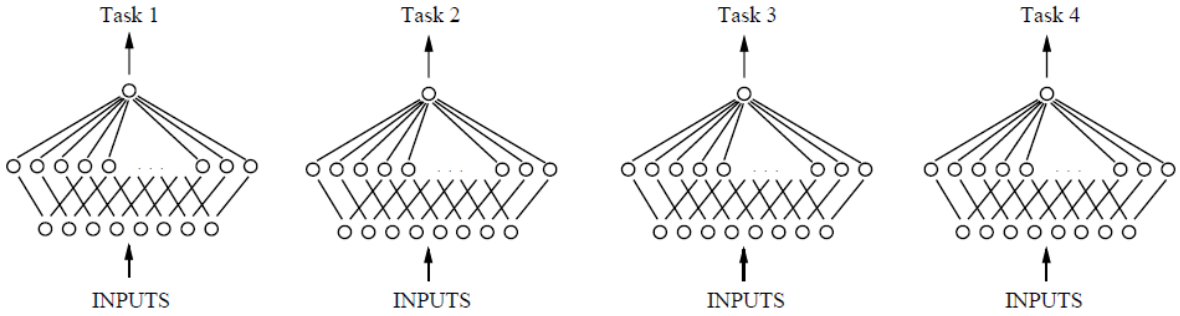


Figure: Learning Tasks Separately

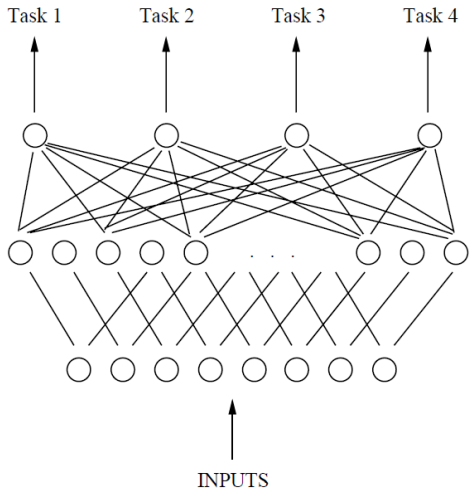
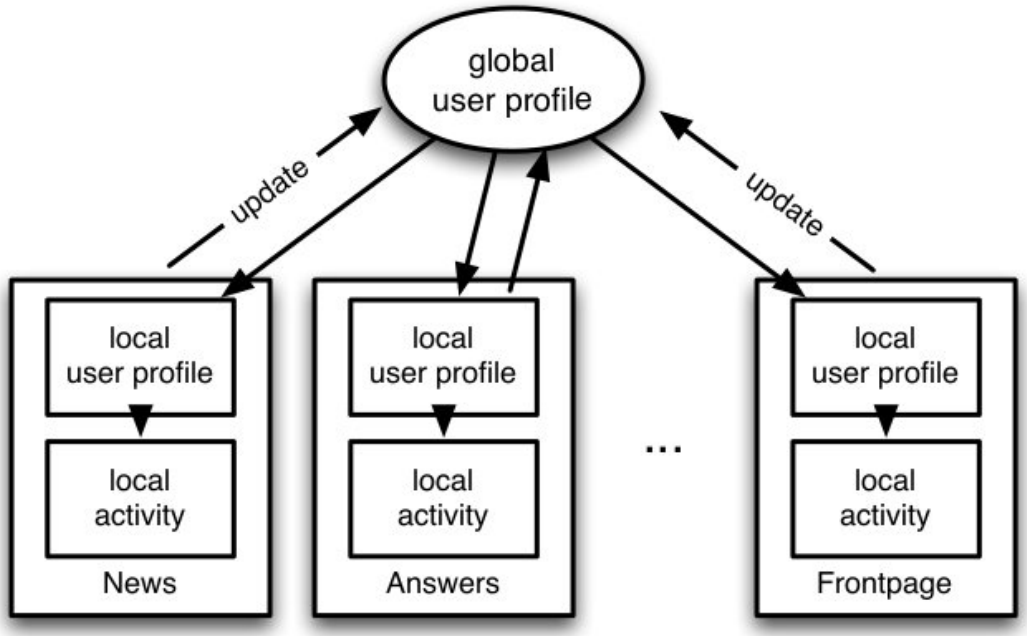


Figure: Learning Tasks Jointly

Applications



Mathematical Formulation

- Using ℓ_1/ℓ_q (Group Sparse) regularization.
- Using graphical models with shared structure.
- Other methods.

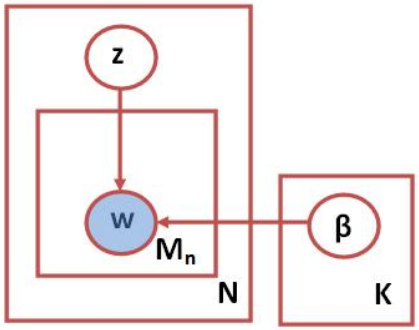
Problem Definition

- Object detection from images.
- Both annotations and class labels are used in prediction.
- Some variant of Latent Dirichlet Allocation is used as a model for learning.

Topic Models for Analysis of Text Data

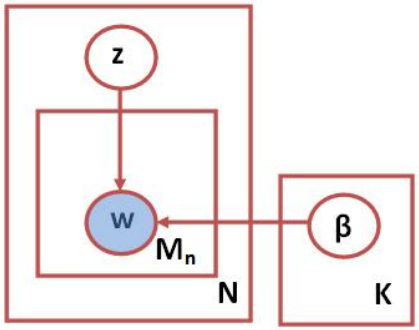
- Singular Value Decomposition – decomposition of a matrix into a weighted sum of rank-one matrices.
- Low rank structure of the document-word matrix is assumed – LSI – Latent Semantic Indexing.
- Mixture of unigrams.
- pLSI – Probabilistic Latent Semantic Indexing – words in a given document are sampled from multiple topics.
- Generalization to unseen documents – Latent Dirichlet Allocation (LDA).

Mixture of Unigrams

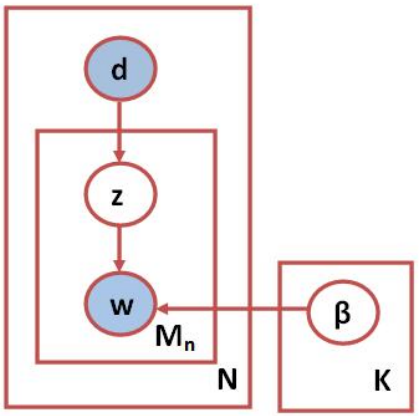


Mixture of Unigrams and pLSI

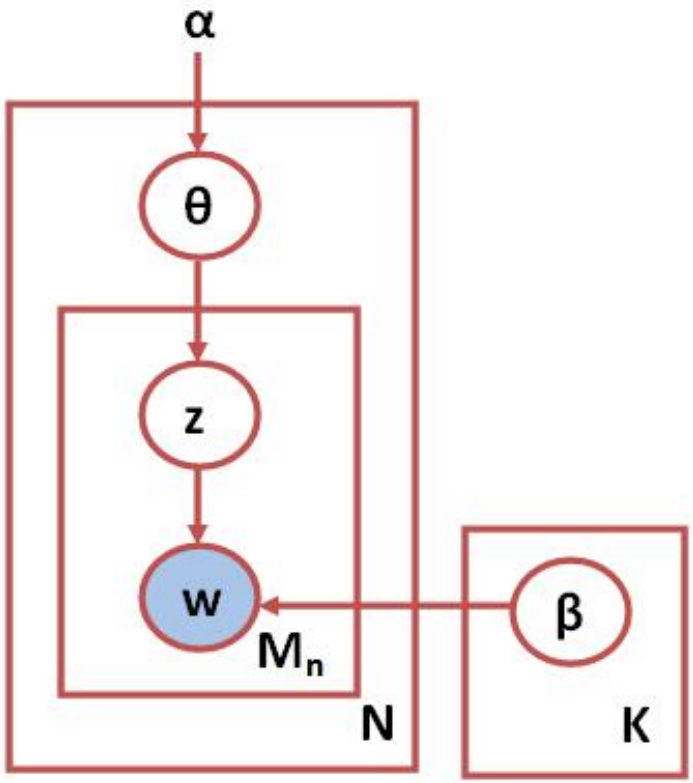
Mixture of Unigrams



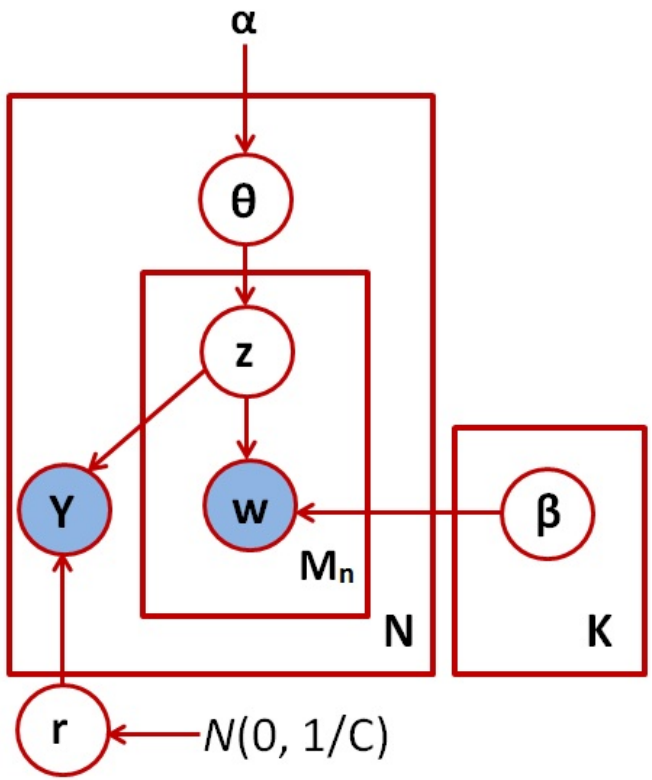
pLSI



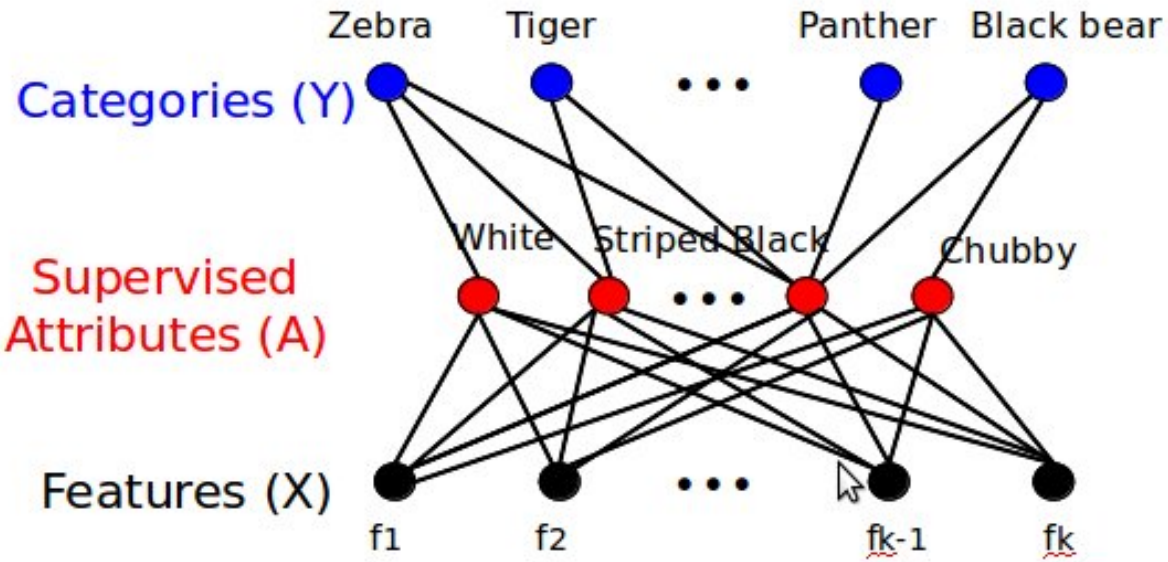
Latent Dirichlet Allocation



Max-Entropy Discriminant Latent Dirichlet Allocation

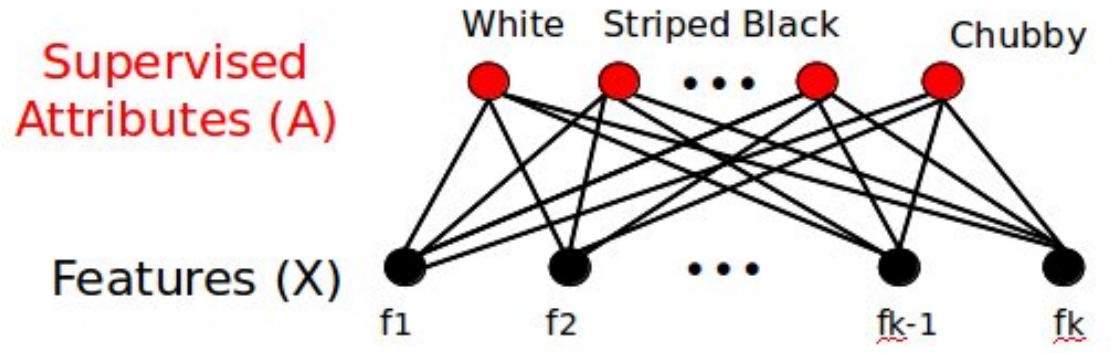


Transfer with Supervised Shared Attributes



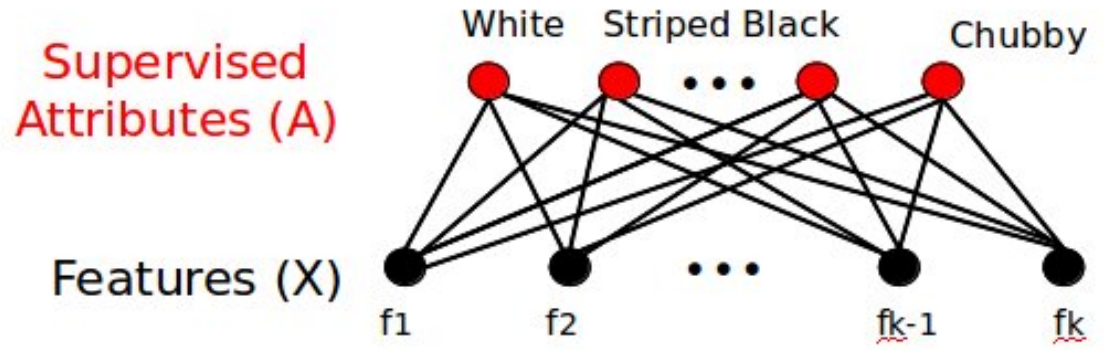
Two Level Inference

Infer attributes from features.

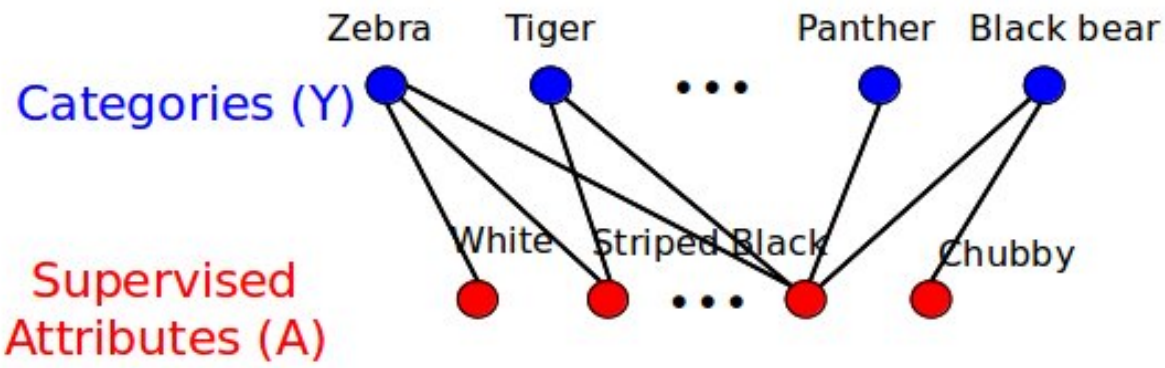


Two Level Inference

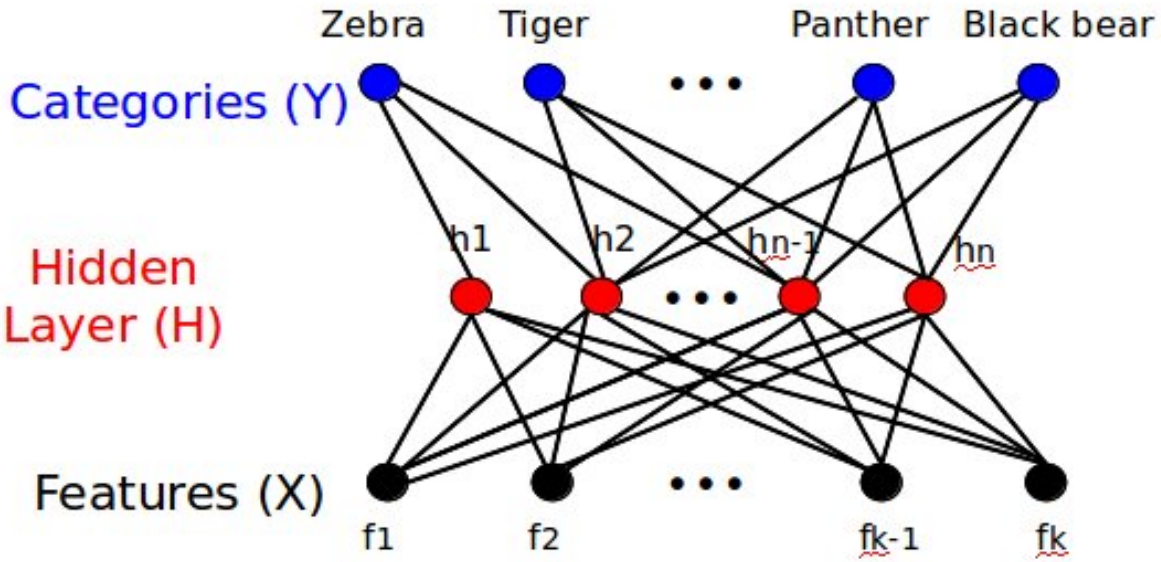
Infer attributes from features.



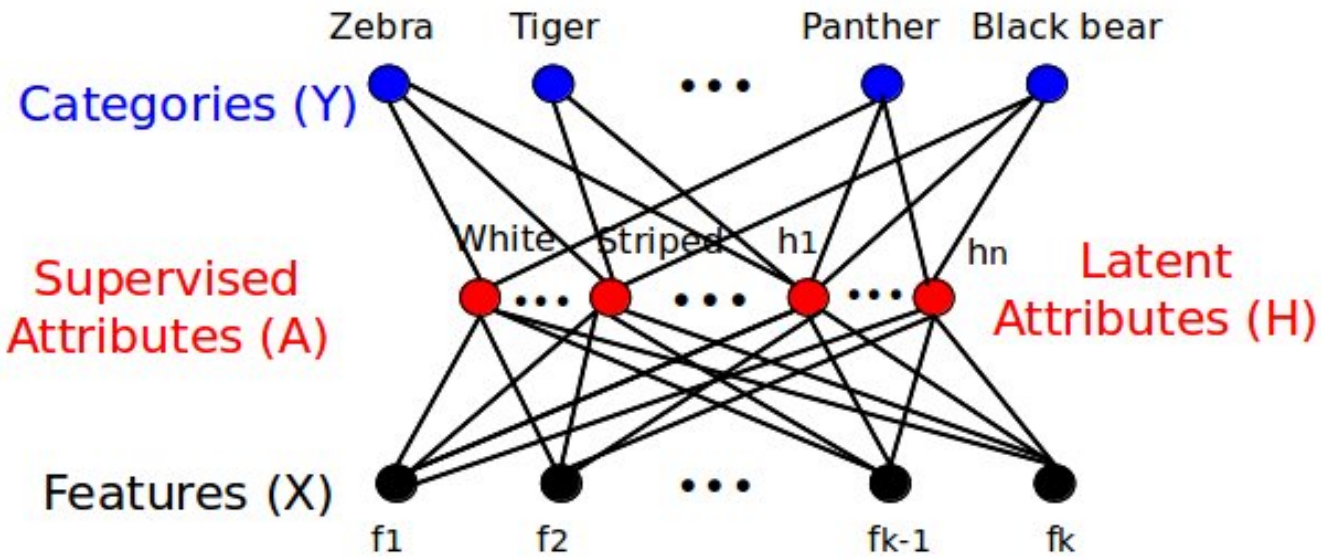
Infer categories from attributes.



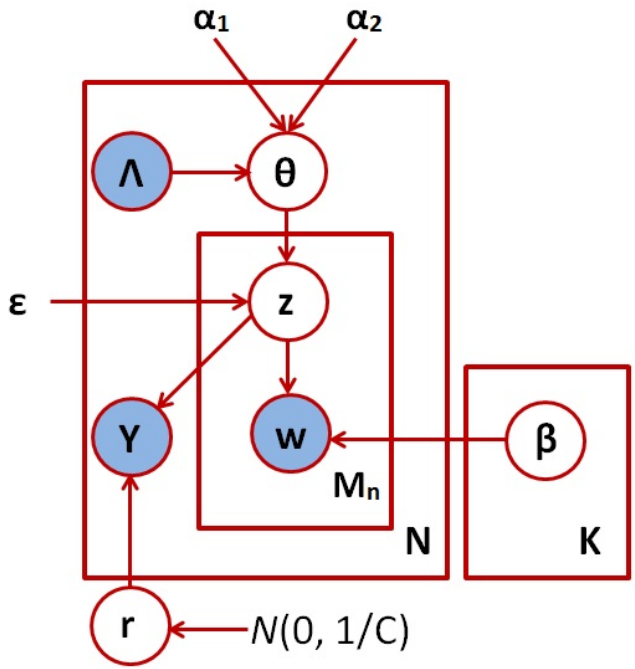
Multitask Learning with Shared Latent Attributes



MTL with Shared Latent and Supervised Attributes



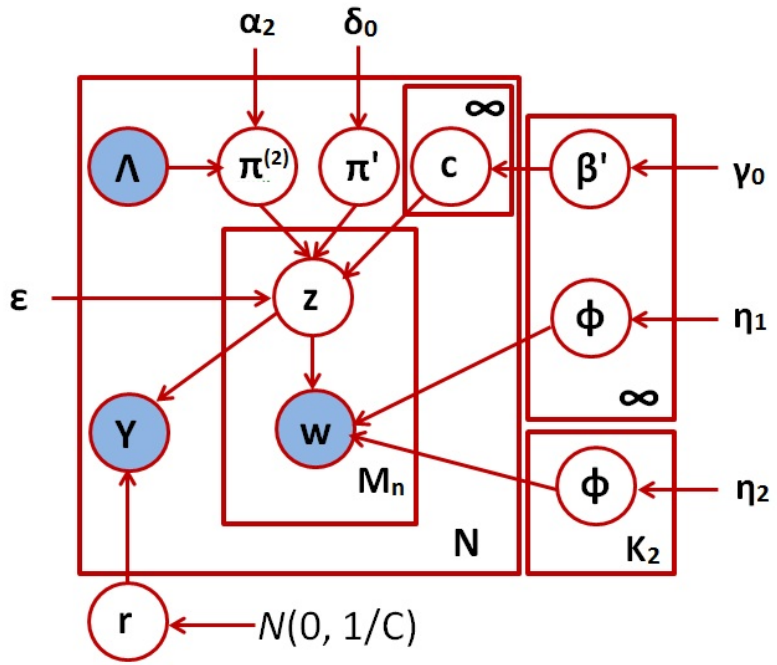
Doubly Supervised LDA



Generative Process

- For the n^{th} document, sample $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\alpha}_n)$, where $\boldsymbol{\alpha}_n = \boldsymbol{\Lambda}_n \boldsymbol{\alpha}$.
- For the m^{th} word in the n^{th} document, sample a topic $z_{nm} \sim \text{multinomial}(\boldsymbol{\theta}'_n)$, where $\boldsymbol{\theta}'_n = (1-\epsilon)\{\boldsymbol{\theta}_{nk}\}_{k=1}^{k_1} \epsilon \{\Lambda_{n,kk}\boldsymbol{\theta}_{nk}\}_{k=1+k_1}^{K}$.
- Sample the word $w_{nm} \sim \text{multinomial}(\boldsymbol{\beta}_{z_{nm}})$, where $\boldsymbol{\beta}_k$ is a multinomial distribution over the vocabulary of words corresponding to the k^{th} topic.
- For the n^{th} document, generate $Y_n = \arg\max_y \mathbf{r}_y^T \mathbb{E}(\bar{\mathbf{z}}_n)$ where Y_n is the class label associated with the n^{th} document, $\bar{\mathbf{z}}_n = \sum_{m=1}^{M_n} \mathbf{z}_{nm}/M_n$.
- $\mathbf{r}_y$ is a K-dimensional real vector corresponding to the y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. M_n is the number of words in the n^{th} document.

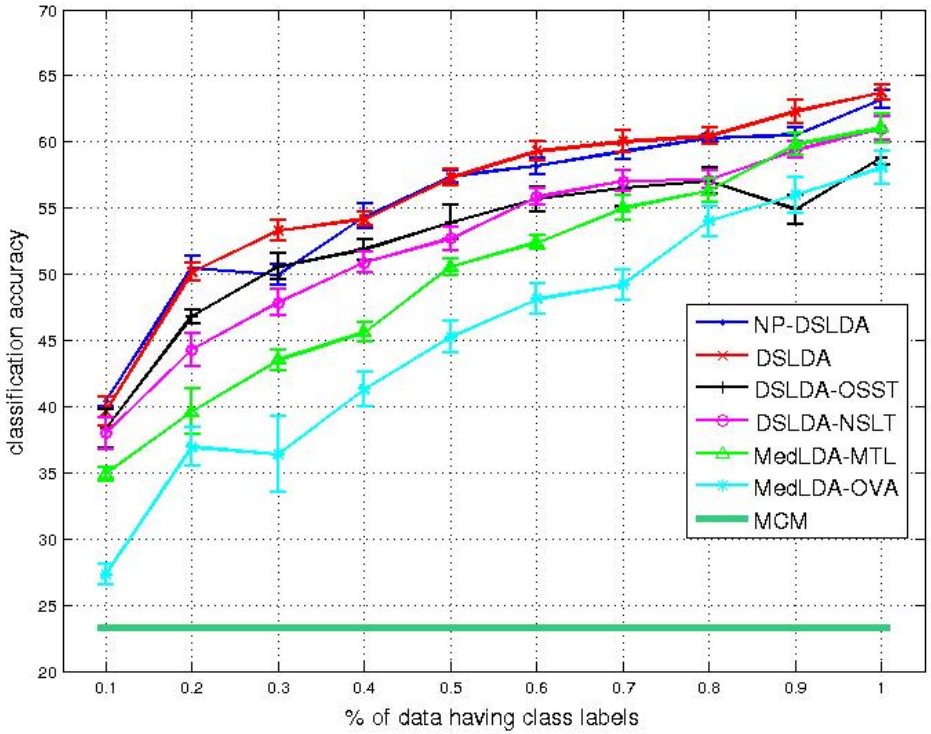
Non-parametric Doubly Supervised LDA



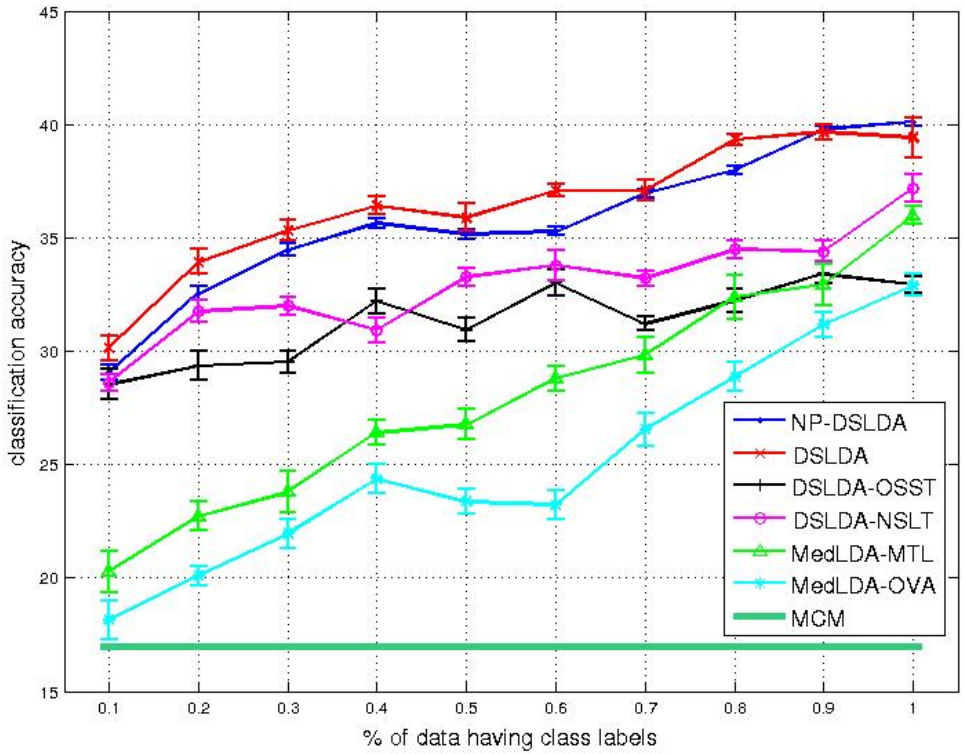
Generative Process

- Sample $\boldsymbol{\phi}_{k_1} \sim \text{Dir}(\boldsymbol{\eta}_1)\ \forall k_1 \in \{1, 2, \cdots, \infty\}$ and $\boldsymbol{\phi}_{k_2} \sim \text{Dir}(\boldsymbol{\eta}_2)\ \forall k_2 \in \{1, 2, \cdots, K_2\}$.
- Sample $\beta'_{k_1} \sim \text{Beta}(1, \delta_0)\ \forall k_1 \in \{1, 2, \cdots, \infty\}$.
- For the n^{th} document, sample $\boldsymbol{\pi}_n^{(2)} \sim \text{Dir}(\boldsymbol{\Lambda}_n \boldsymbol{\alpha}_2)$.
- $\forall n, \forall t \in \{1, 2, \cdots, \infty\}$, sample $\pi'_{nt} \sim \text{Beta}(1, \alpha_0)$. Assume $\boldsymbol{\pi}_n^{(1)} = (\pi_{nt})_t$ where $\pi_{nt} = \pi'_{nt} \prod_{l<t}(1 - \pi'_{nl})$.
- $\forall n, \forall t$, sample $c_{nt} \sim \text{multinomial}(\boldsymbol{\beta})$ where $\beta_{k_1} = \beta'_{k_1} \prod_{l<k_1}(1 - \beta'_l)$.
- For the m^{th} word in the n^{th} document, sample $z_{nm} \sim \text{multinomial}((1 - \epsilon)\boldsymbol{\pi}_n^{(1)}, \epsilon\boldsymbol{\pi}_n^{(2)})$.
- Sample w_{nm} from a multinomial given by Eq. (**??**).

Results on Text Data



Results on Image Data



Questions?

1. An Optimization Framework for Semi-Supervised and Transfer Learning using Multiple Classifiers and Clusterers.
2. Using Both Supervised and Latent Shared Topics for Multitask Learning.

Acknowledgement: Dr. Joydeep Ghosh, Dr. Raymond J. Mooney, Dr. Eduardo R. Hruschka, Sreangsu Acharyya.