Adopting Semi-supervised Learning Algorithms for Mining Remote Sensing Imagery: Summary of Results and Open Research Problems

Ranga Raju Vatsavai[1,3],
Shashi Shekhar[1], and Thomas E. Burk[2].

[1]Spatial Database Research Group,
Department of Computer Science.

[2]Remote Sensing and Geospatial Analysis Lab,
Department of Forest Resources.
University of Minnesota.

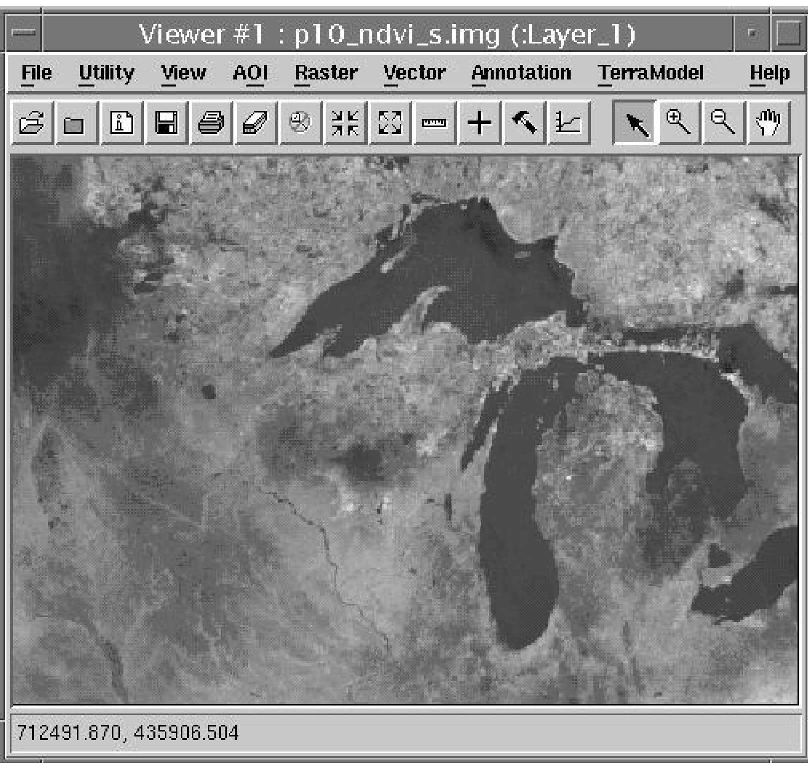
[3]IBM-Research, India Research Laboratory
Indian Institute of Technology-Delhi, India.

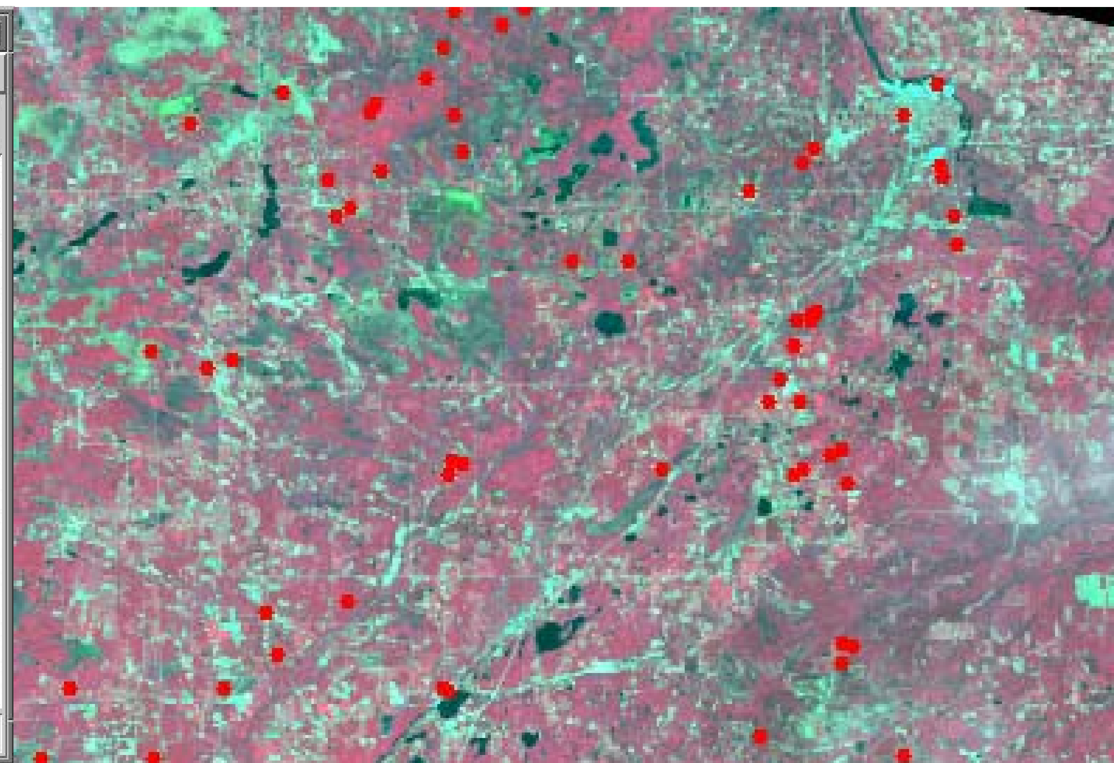
Outline

- Introduction
- Semi-supervised Classification
- Results
- Conclusions and Open Research Problems

Introduction

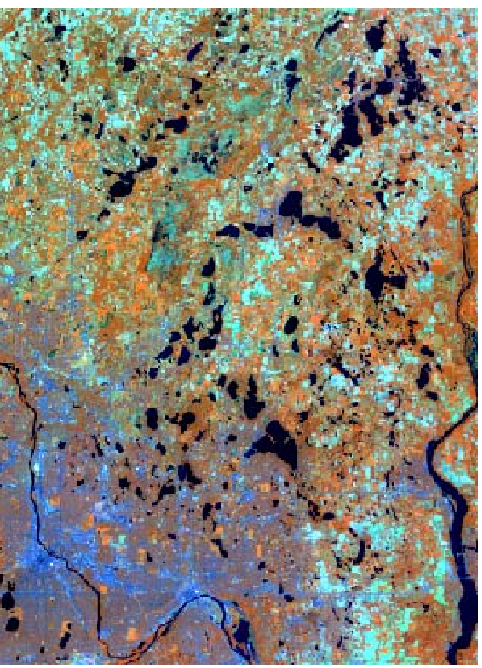
- Remote Sensing
  - Spectral Resolution – multi-spectral to hyper-spectral
  - Spatial Resolution – low (60m) to high (1m)

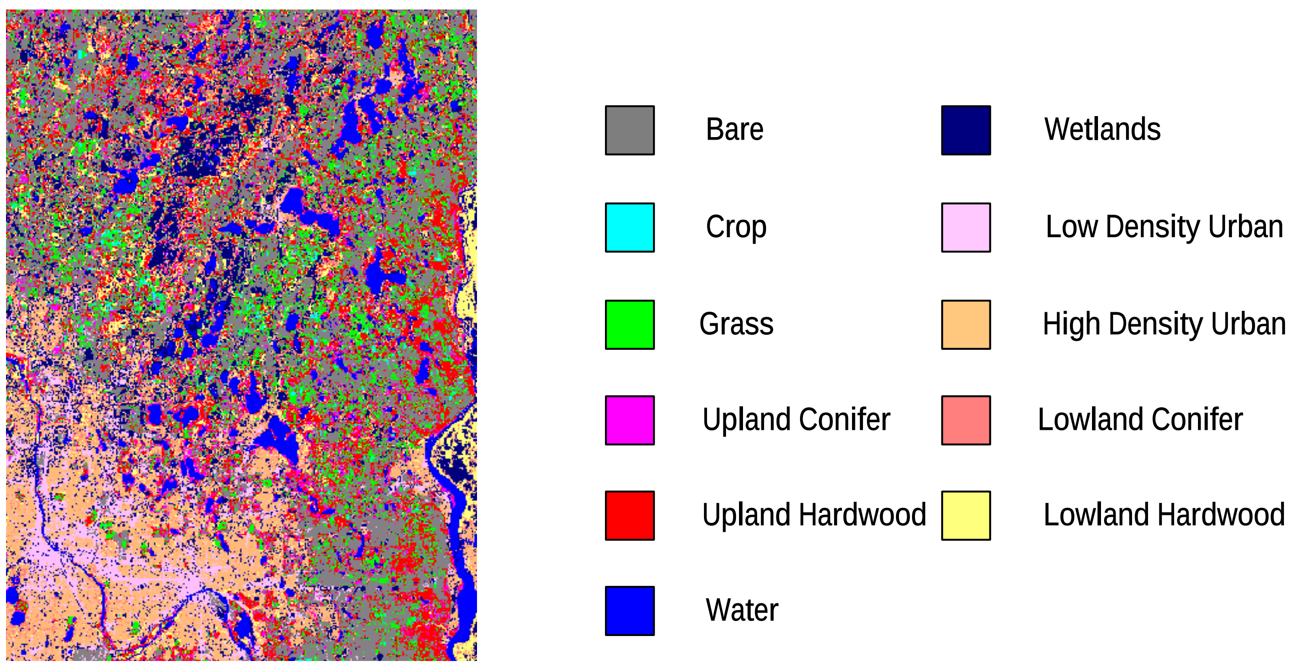




Introduction

- Classification*
  - Supervised – MLC, MAP, DT, NN, ...
  - Unsupervised – Clustering (kNN, kMeans, GMM...)

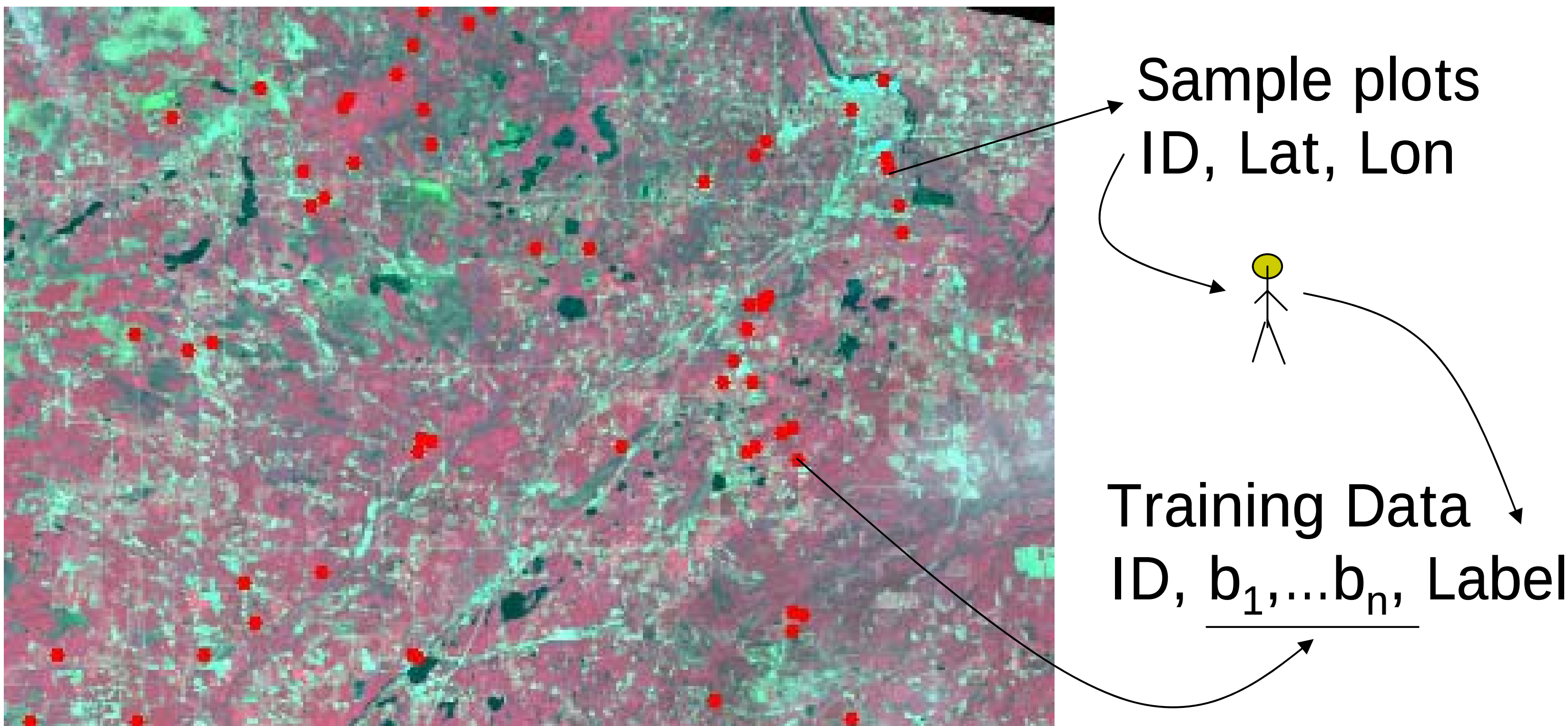




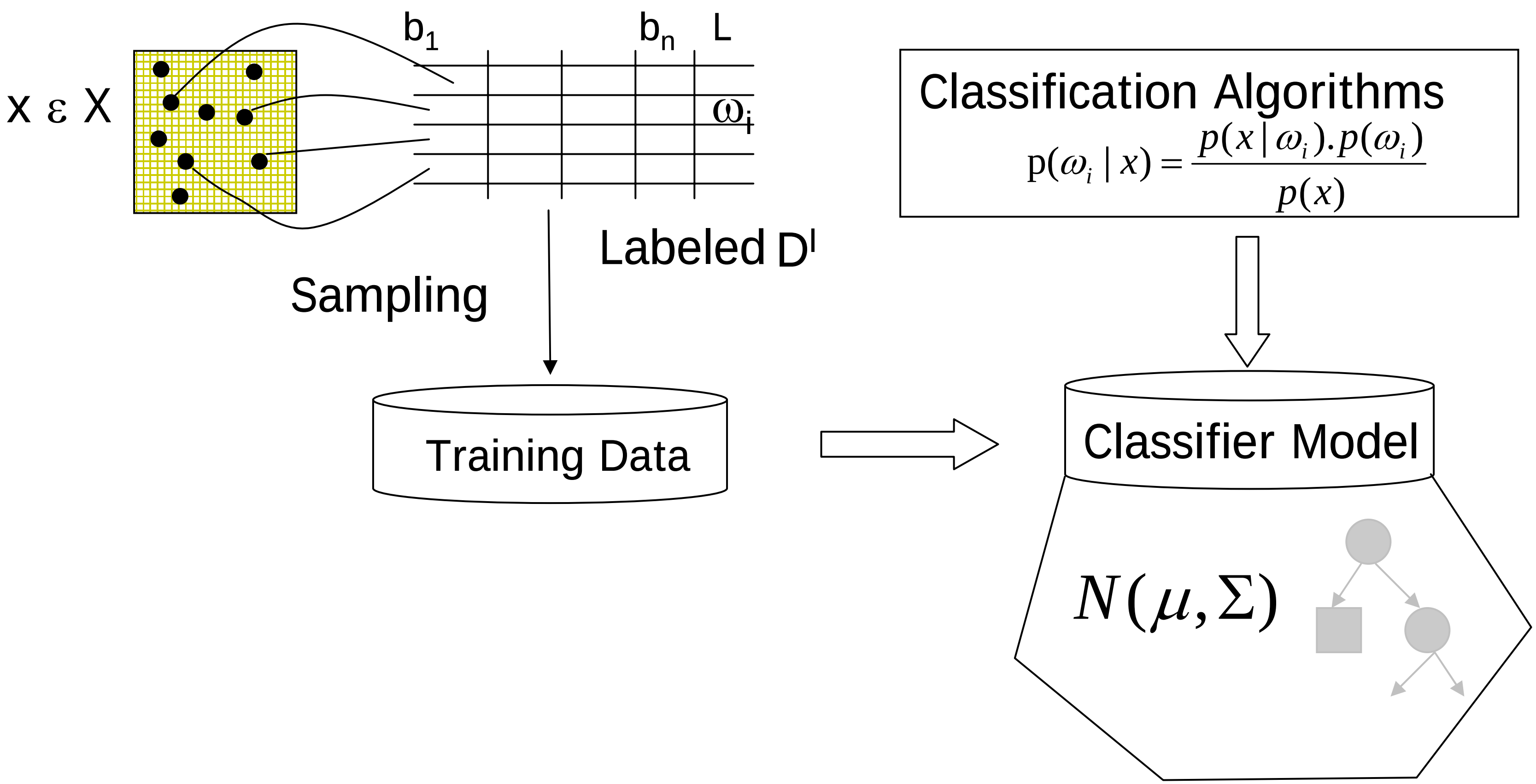
*Anil Jain, et. al. Statistical Pattern Recognition – A Review. IEEE PAMI, 22(1), 2000.

Supervised Classification Process

- Example Application – Thematic Mapping
  - 7Bands, 7000Px7000L, 30m Pixel size, 16day revisit, 170x183km.



Supervised Classification Process



Supervised Classification Process

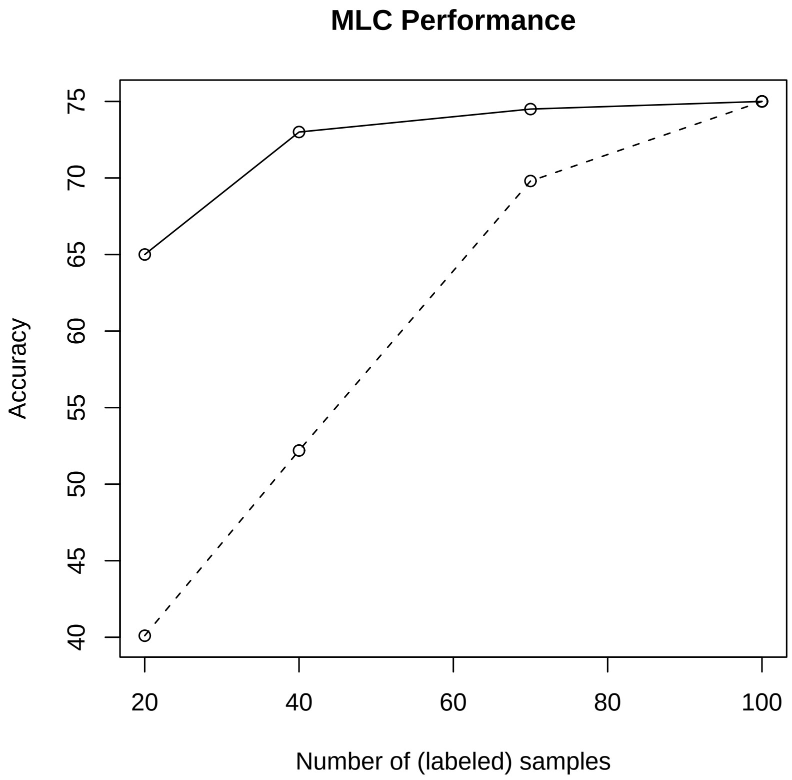
- Bayes' Theorem: $\mathrm{p}(\omega_i \mid x) = \dfrac{p(x \mid \omega_i).p(\omega_i)}{p(x)}$
- Assuming $D_i = \{x_1, ..., x_n\}$, and x's i.i.d, the likelihood function is simply: $p(D_1 \mid \theta_1) = \prod_{k=1}^{n} p(x_k \mid \theta_1)$
- The maximum-likelihood estimation of θ_1 is the value that maximizes $p(D_1 \mid \theta_1)$. $\hat{\theta}_1 = \arg\max_{\theta_1} \; l(\theta_1)$

$$\hat{\mu}_1 = \frac{1}{n}\sum_{k=1}^{n} x_k$$

- MLE gives familiar quantities: $\hat{\Sigma}_1 = \dfrac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu}_1)(x_k - \hat{\mu}_1)'$
  - Asymptotically unbiased, consistent, efficient
  - All these properties are true if n -> ∞
  - |D| ~ (10 to 100) * (number of dimensions)

Supervised Classification Process

- MLC performance as |D| increases



Symbols

x = Feature Vector

n = #of features or #dimensions

μ = Mean Vector

One μ per class

∑ = Covariance Matrix

One ∑ per class

Subscripts are context dependent

$$x = \begin{bmatrix} b_1 \\ \ldots \\ b_n \end{bmatrix}_{nx1} = \begin{bmatrix} 63 \\ 53 \\ 35 \\ 121 \\ 76 \\ 31 \end{bmatrix}, \quad \mu_\omega = \begin{bmatrix} m_1 \\ \ldots \\ m_n \end{bmatrix}_{nx1} = \begin{bmatrix} 67.33 \\ 60.44 \\ 41.44 \\ 89.77 \\ 79.66 \\ 40.44 \end{bmatrix}$$

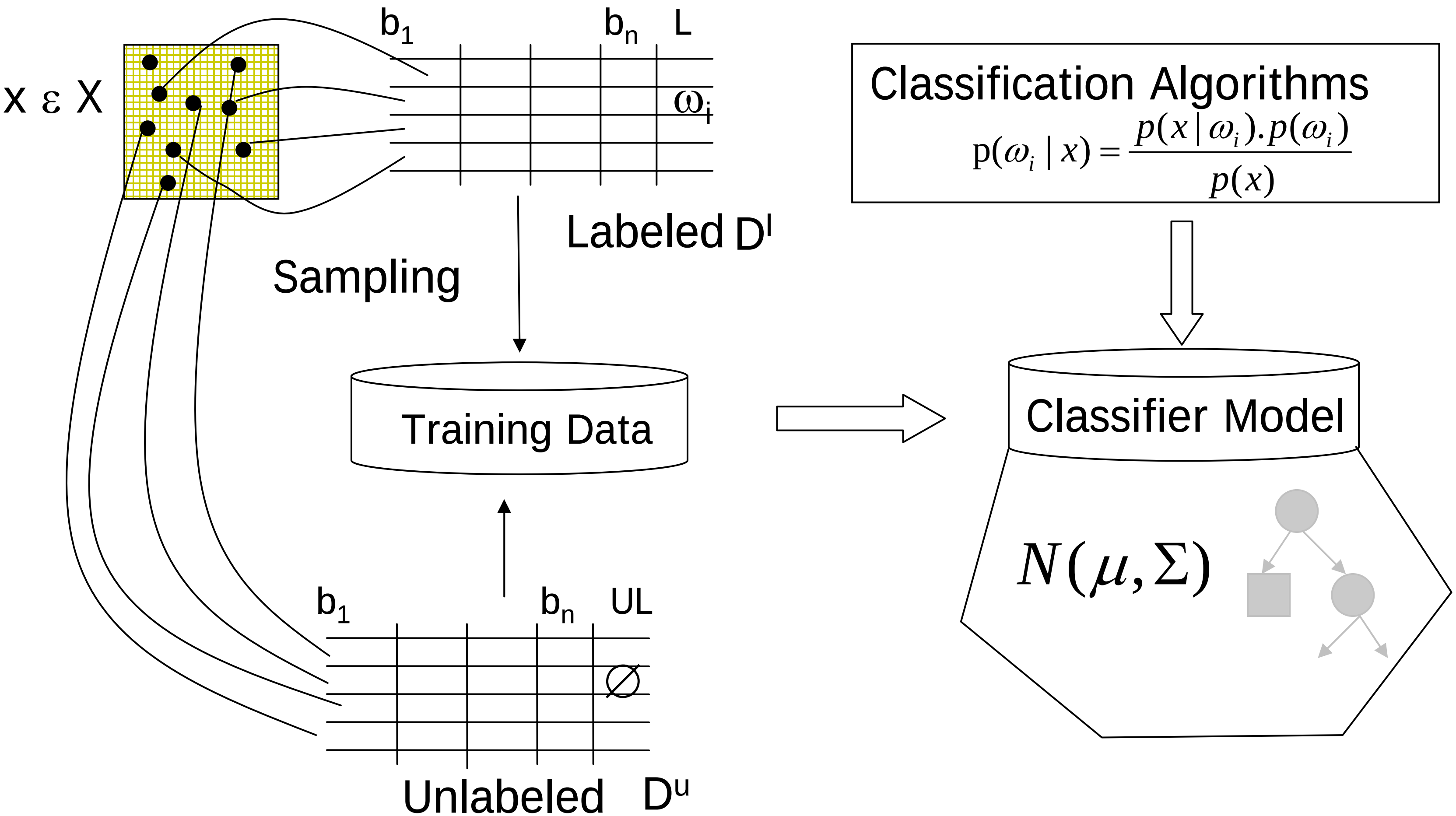
$$\Sigma_\omega = \begin{bmatrix} v_{1,1} & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & v_{n,n} \end{bmatrix}_{nxn} = \begin{bmatrix} 2.0 & .2 & .33 & .58 & 1.13 & -0.7 \\ .41 & 1.7 & .6 & .2 & -.3 & -1.2 \\ & & & & & \\ & & & & & \\ -1.16 & & & & & 2.7 \end{bmatrix}$$

Semi-supervised Classification

- Acquiring Labels
  - Costly, Time consuming, Labor intensive
  - Error prone
  - May be impossible at times –
    - Emergency situations (fires, floods, cyclones, ...)
    - Accessibility, Privacy

- Samples are readily available, but not labels
- There are several classification problems that have to deal with insufficient learning samples
- Q? Can we still learn with partially labeled training data

Semi-supervised Classification Process



Learning with incomplete data $D = D^l \cup D^u$

- Assume current estimate of parameter is θ_i
- See what happens to L when new θ is estimated

$$L(\theta) - L(\theta_i) = \ln p(x \mid \theta) - \ln p(x \mid \theta_i) = \ln \frac{p(x \mid \theta)}{p(x \mid \theta_i)}$$

- We would like to choose θ to maximize r.h.s
- EM*, introduce unobserved vars Z such that if Z is known, the θ is computed easily

$$L(\theta) - L(\theta_i) = \ln \frac{\sum_z p(x \mid z, \theta) p(z \mid \theta)}{p(x \mid \theta_i)}$$

*A. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

Learning with incomplete data $D = D^l \cup D^u$

- The EM algorithm first finds the expected value of the complete-data log-likelihood –
  - E-step $$Q(\theta;\theta^{(i)}) = E[\log p(D^l, D^u;\theta) \mid D^l;\theta^{(i)}]$$

- The second step (M-step) of the EM algorithm is to maximize the E[] of Ist step.
  - M-step $$\theta^{(i+1)} = \arg\max_{\theta} Q(\theta,\theta^{(i)})$$

EM update equations for GMM*

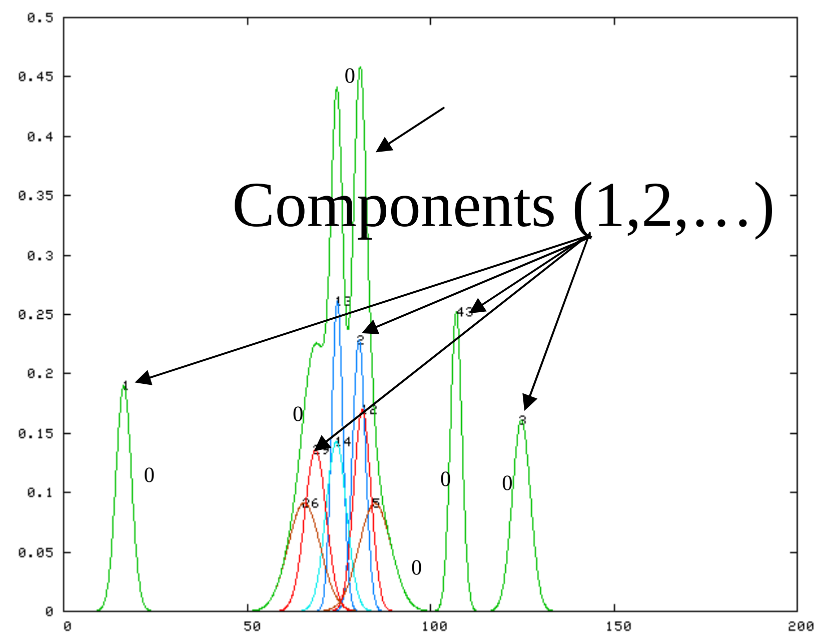
- Assume that D is drawn from a Gaussian Mixture Model, given by

$$p(x \mid \theta) = \sum_{i=1}^{M} \alpha_i p_i(x \mid \theta_i)$$

where $\theta = (\alpha_1,\ldots,\alpha_M;\theta_1,\ldots\theta_M)$

such that $\sum_1^M \alpha_i = 1, \quad 0 < \alpha_i < 1$ and

p_i pdf parameterized by θ_i



*J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report, University of Berkeley, ICSI-TR-97-021, 1997., 1997.

GMM – Update Equations

- E-Step

$$e_{ij} = \frac{\left|\hat{\Sigma}_j^k\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(x_i - \hat{\mu}_j^k\right)^T \hat{\Sigma}_j^{-1,k}\left(x_i - \hat{\mu}_j^k\right)\right\}}{\sum_{l=1}^{M}\left|\hat{\Sigma}_l^k\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(x_i - \hat{\mu}_l^k\right)^T \hat{\Sigma}_l^{-1,k}\left(x_i - \hat{\mu}_l^k\right)\right\}}$$

- M-Step

$$\alpha_j = \frac{\sum_{i=1}^{N} e_{ij}}{N}, \qquad \hat{\mu}_j^{k+1} = \frac{\sum_{i=1}^{N} e_{ij} x_i}{\sum_{i=1}^{N} e_{ij}},$$

$$\text{and} \quad \hat{\Sigma}_j^{k+1} = \frac{\sum_{i=1}^{N} e_{ij}\left(x_i - \hat{\mu}_j^{k+1}\right)\left(x_i - \hat{\mu}_j^{k+1}\right)^T}{\sum_{i=1}^{N} e_{ij}}$$

i^{th} data vector, j^{th} class

GMM – New Update Equations

$$\alpha_j = \frac{\left(\lambda_l m_j + \lambda_u \sum_{i=1}^{n} e_{ij}\right)}{\lambda_l m + \lambda_u n},$$

$$\hat{\mu}_j^{k+1} = \frac{\left(\lambda_l \sum_{i=1}^{m_j} y_{ij} + \lambda_u \sum_{i=1}^{n} e_{ij} x_i\right)}{\lambda_l m_j + \lambda_u \sum_{i=1}^{n} e_{ij}},$$

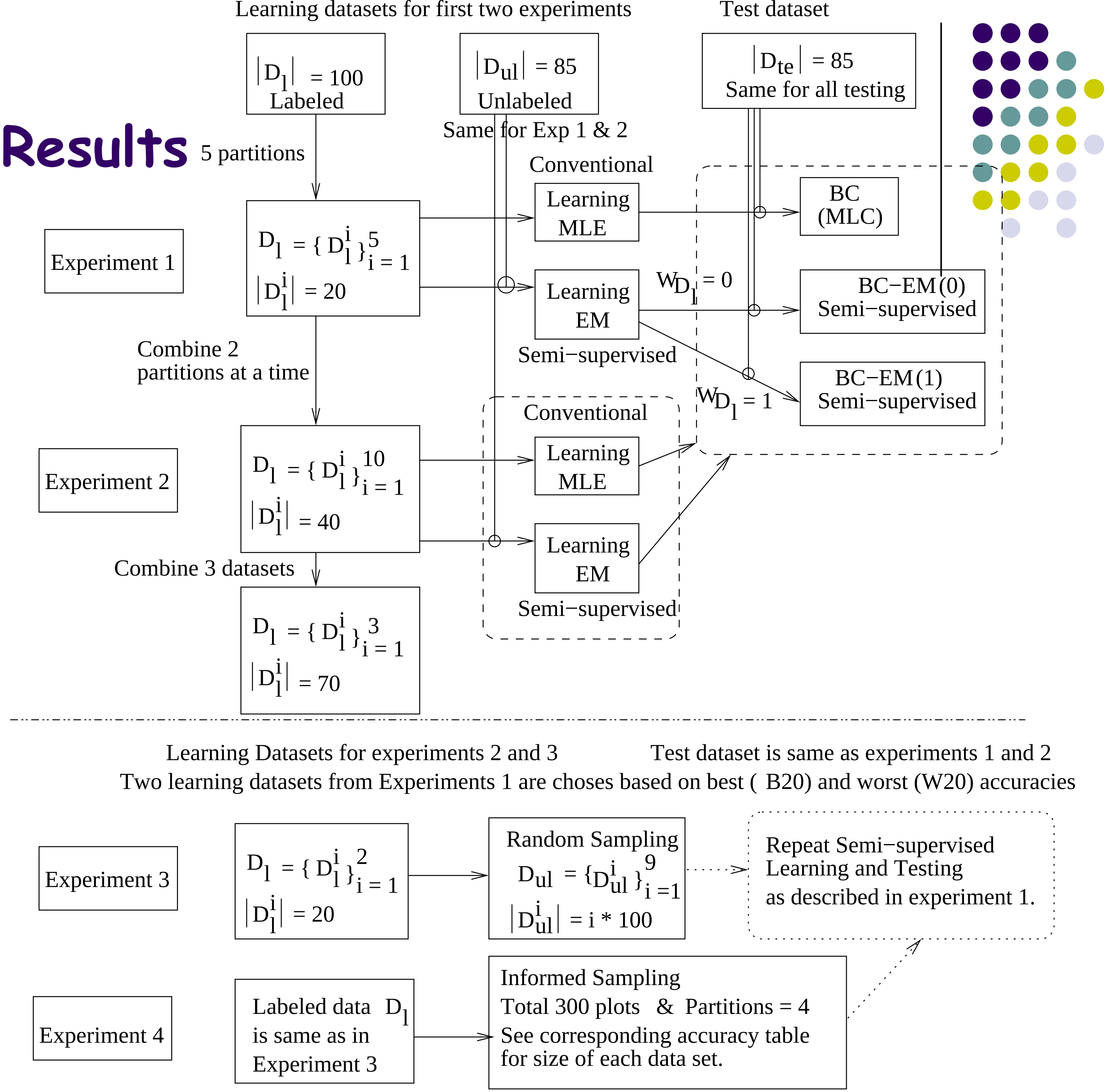
$$\hat{\Sigma}_j^{k+1} = \frac{\lambda_l \sum_{i=1}^{m_j} \left(y_{ij} - \hat{\mu}_j^{k+1}\right)\left(y_{ij} - \hat{\mu}_j^{k+1}\right)^T + \lambda_u \sum_{i=1}^{n} e_{ij} \left(x_i - \hat{\mu}_j^{k+1}\right)\left(x_i - \hat{\mu}_j^{k+1}\right)^T}{\lambda_l m_j + \lambda_u \sum_{i=1}^{n} e_{ij}}$$

i^{th}data vector, j^{th} class

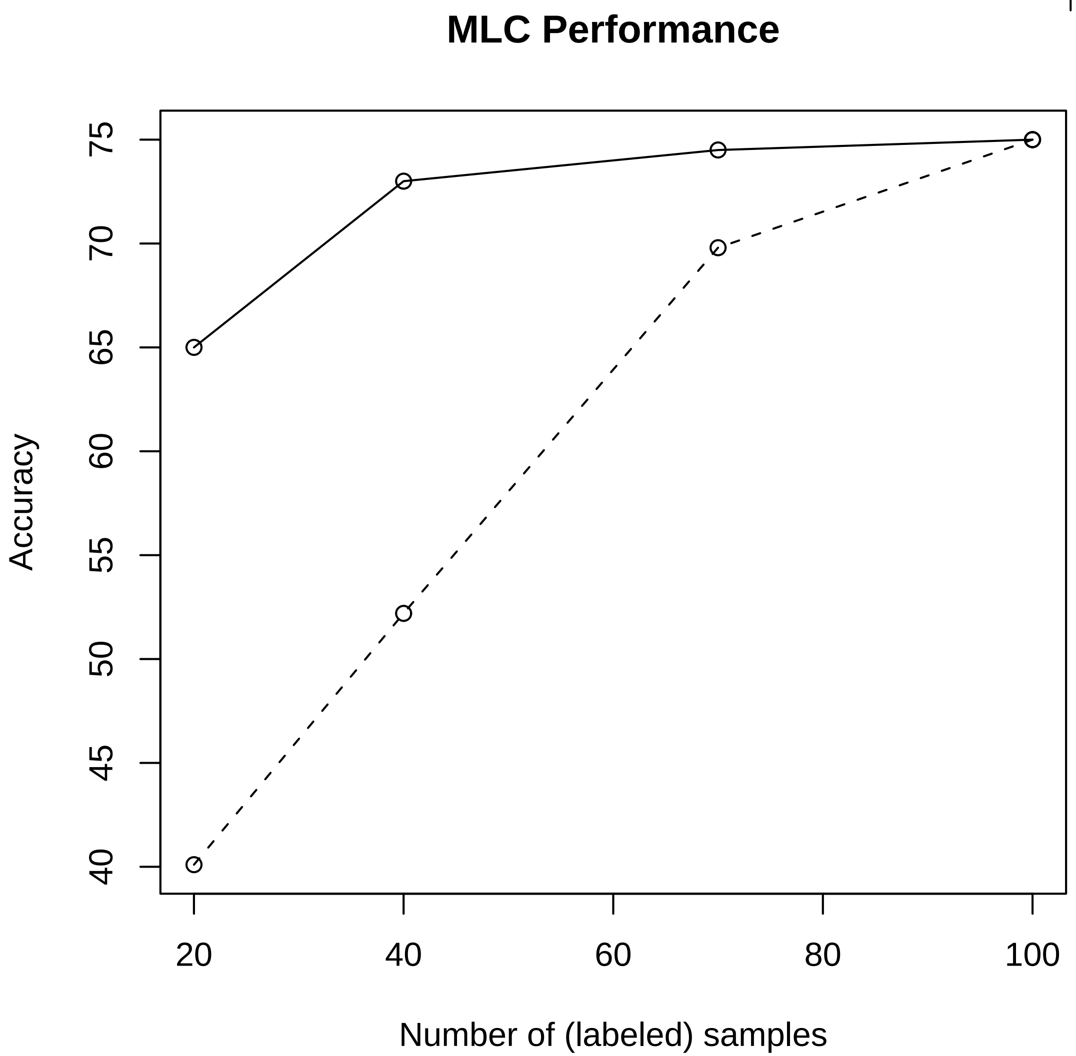
Semi-supervised learning Algorithm - Outline

- Inputs: Collection D consisting of D^l and D^u
- Build initial classifier, estimate $\hat{\theta} = \arg\max_{\theta} l(\theta)$
- Loop while parameter estimate improves
  - (E-Step): Use current classifier estimate, $\hat{\theta}$, to estimate component membership of each data vector, i.e., the prob. that each mixture component (class) generated each sample
  - (M-Step): Re-estimate the classifier parameter, given the estimated component membership of each sample
- Output: A classifier, with improved estimates, that takes any new data sample and predicts a class label.

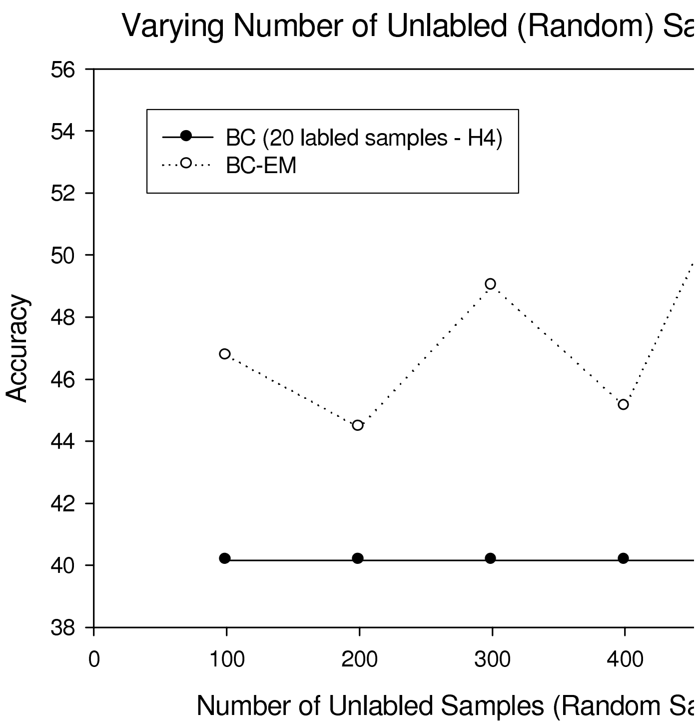
Experimental Results

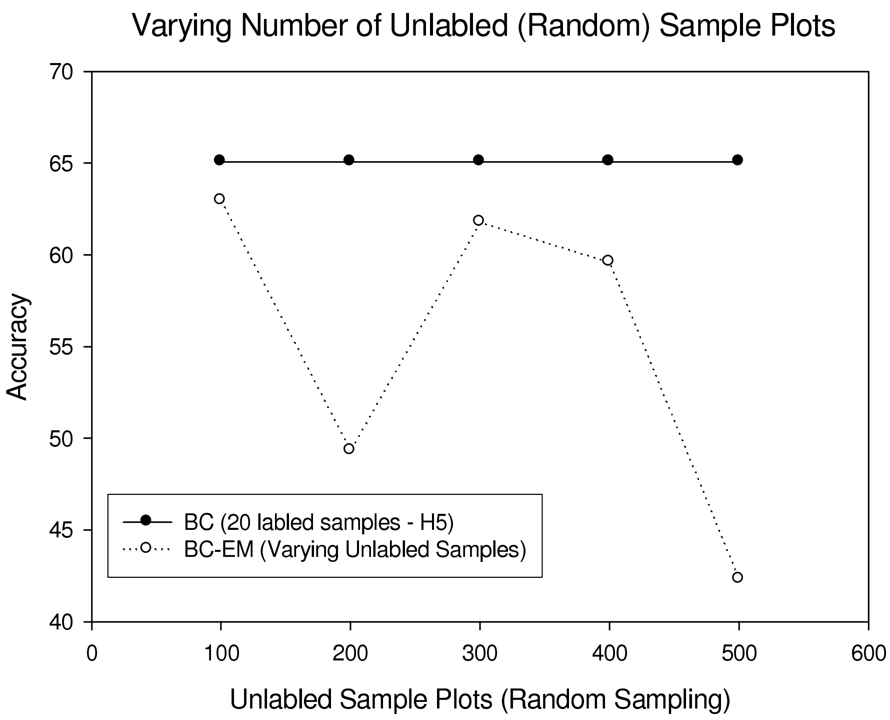


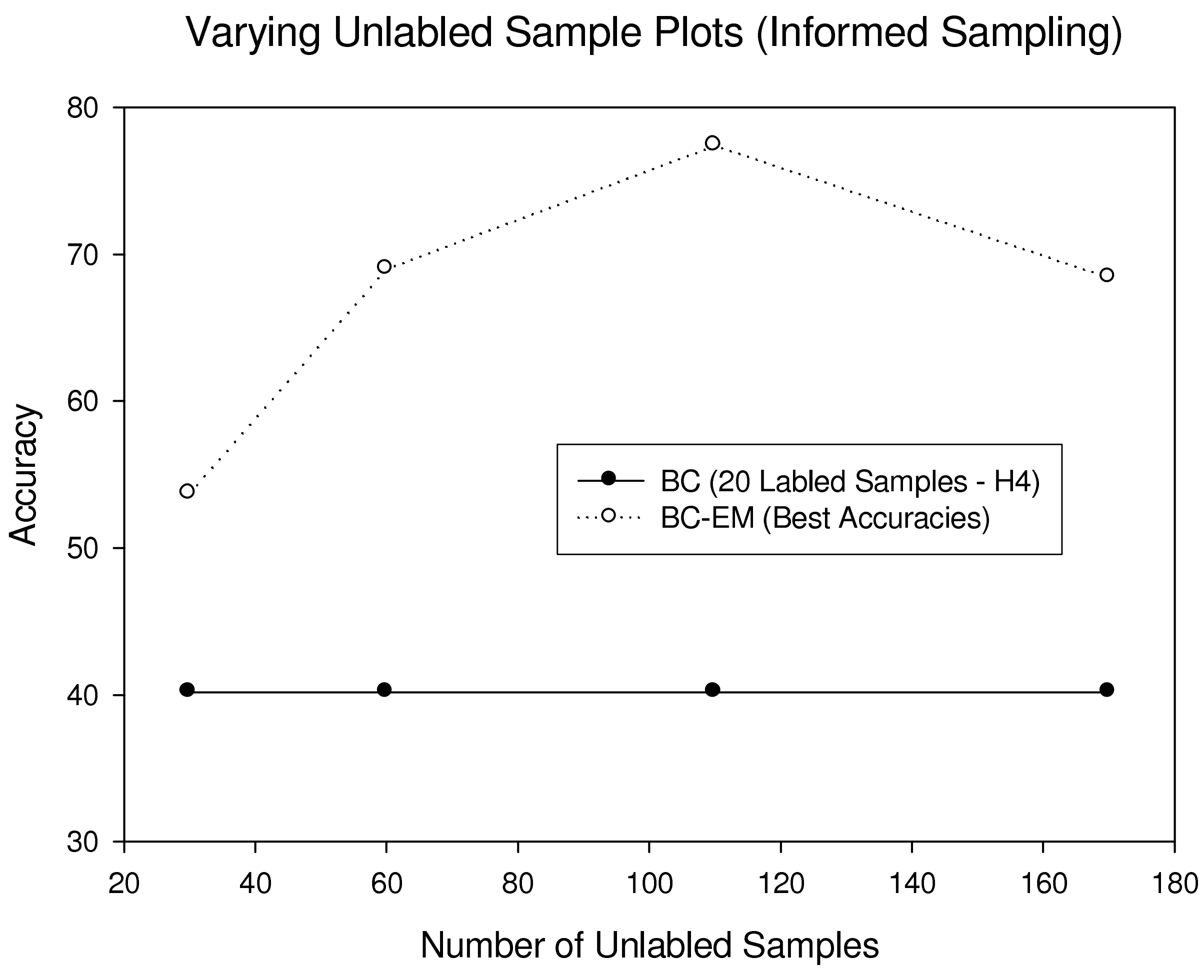
Experimental Results



Experimental Results

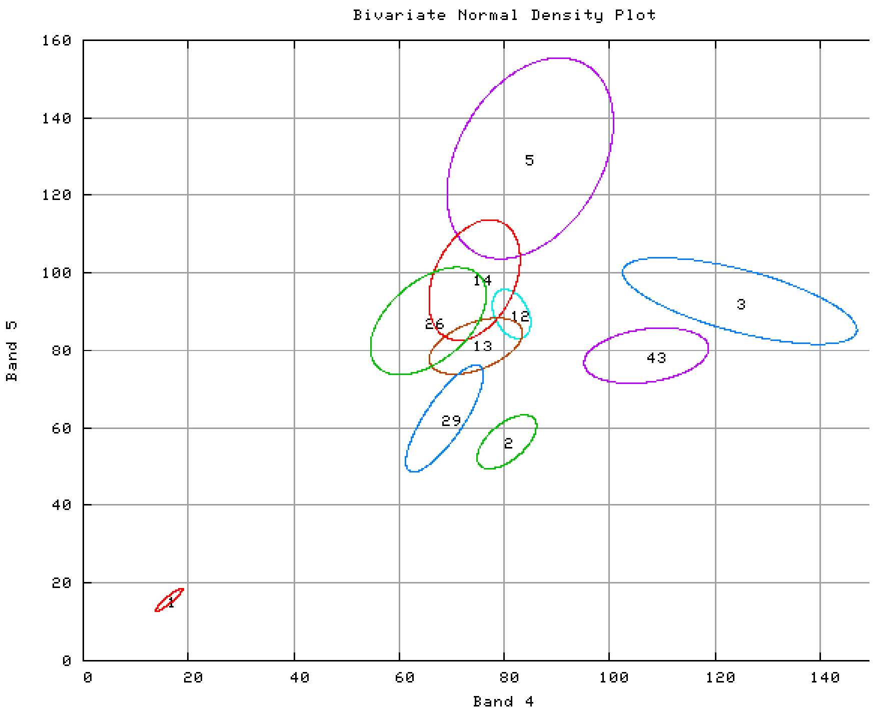




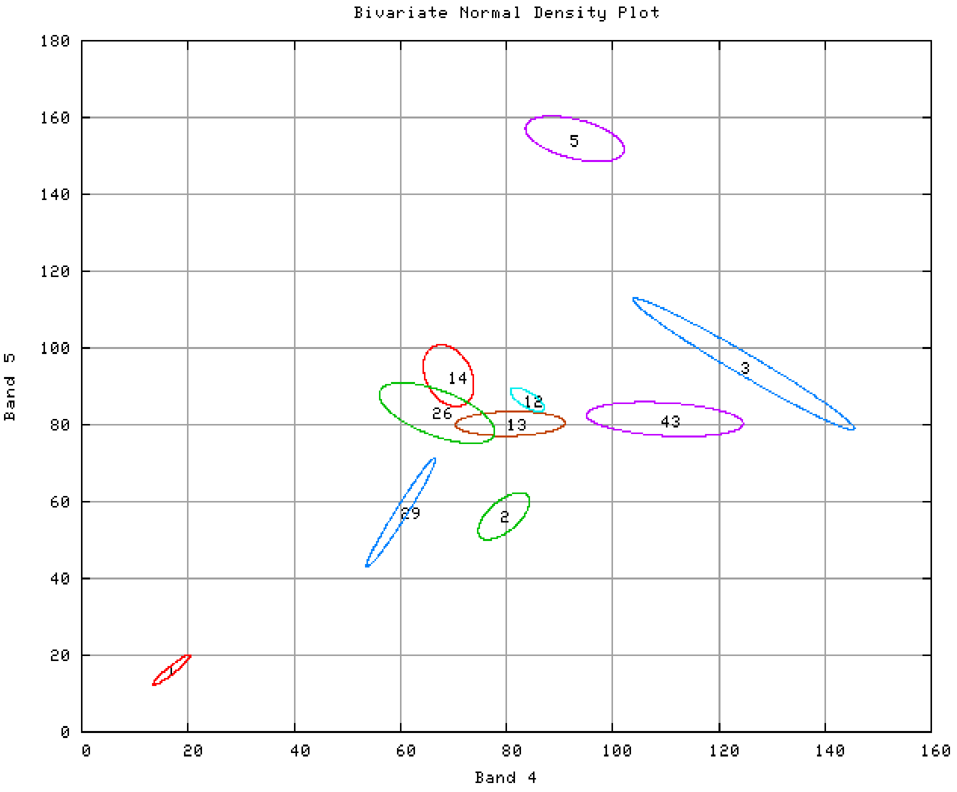


Varying Unlabled Sample Plots (Informed Sampling)
80
70
60
50
40
30
20
40
60
80
100
120
140
160
180
Accuracy
Number of Unlabled Samples
BC (20 Labled Samples - H4)
BC-EM (Best Accuracies)

Experimental Results

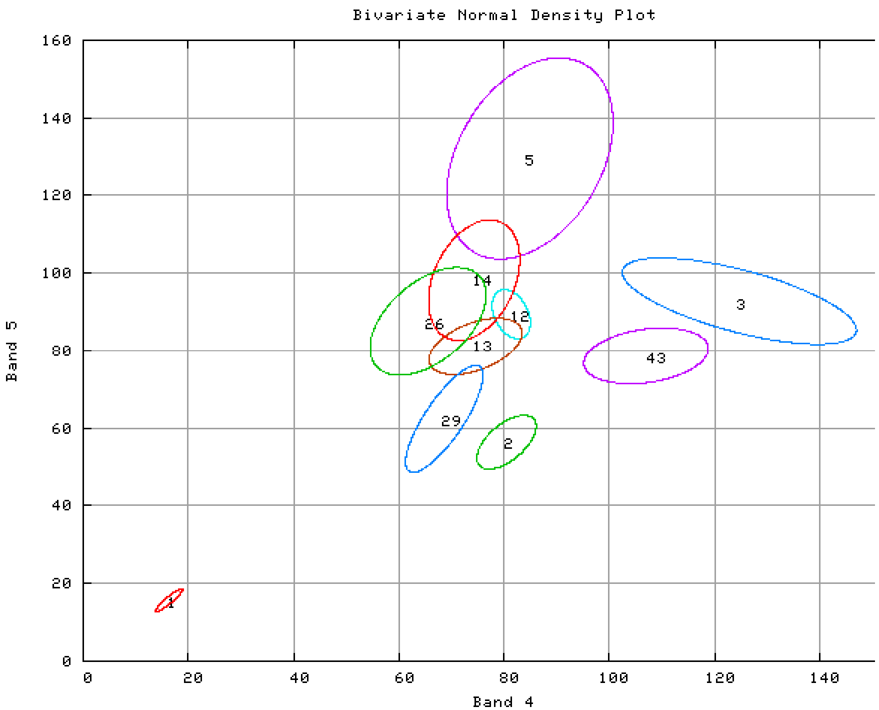


MLE (100 labeled)

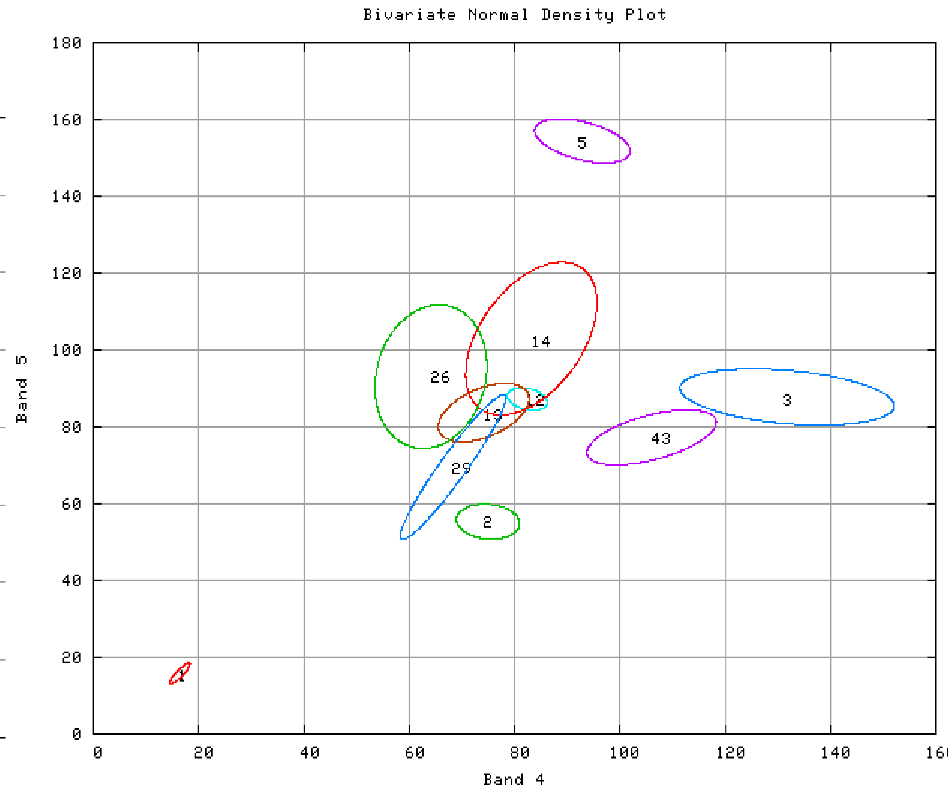


MLE (20 lab)

Experimental Results



MLE (100 labeled)



EM (20 lab + 85 ul)

Spatial Semi-supervised

- iid assumptions are not valid
- MAP/MRF
  - p(c|x) = p(x|c)P(c)/p(x)

$$P(c) = P(c(i,j) \mid c(k,l); \{k,l\} \neq \{i,j\}).$$
$$= P(c(i,j) \mid c(k,l); \{k,l\} \in s).$$
$$= \frac{1}{z} e^{\frac{-U(\omega)}{T}}$$

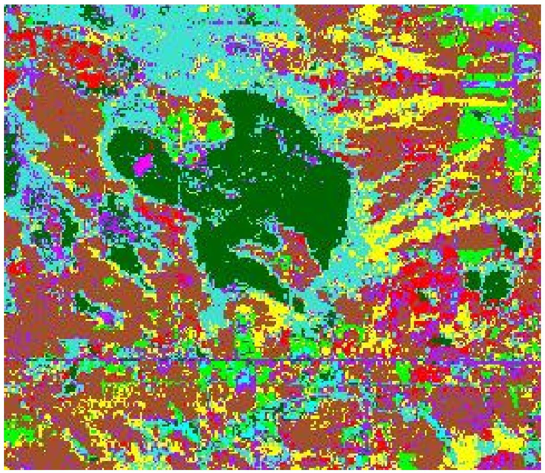
- Z in not computable
  - Assume
    - 10 classes,
    - 256x256 image
  - Total configurations
    - $10^{(256x256)}$

$$U_s(C(i,j)) = \sum_{\{k,l\} \in s_{ij}} \beta I(C(i,j), C(k,l))$$

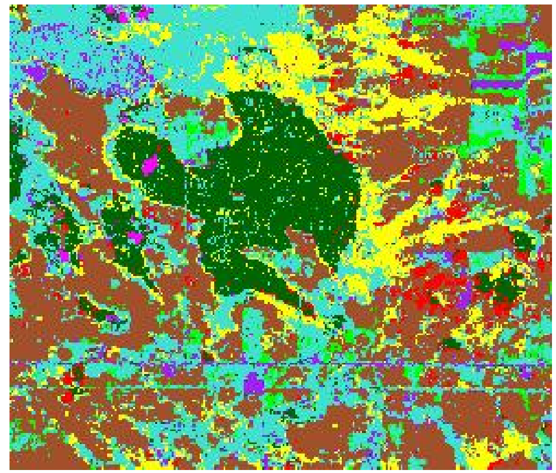
Where

$$I(C(i,j), C(k,l)) = \begin{cases} 1 \text{ if } C(i,j) = C(k,l) \\ 0 \text{ if } C(i,j) \neq C(k,l) \end{cases}$$

Spatial Semi-supervised

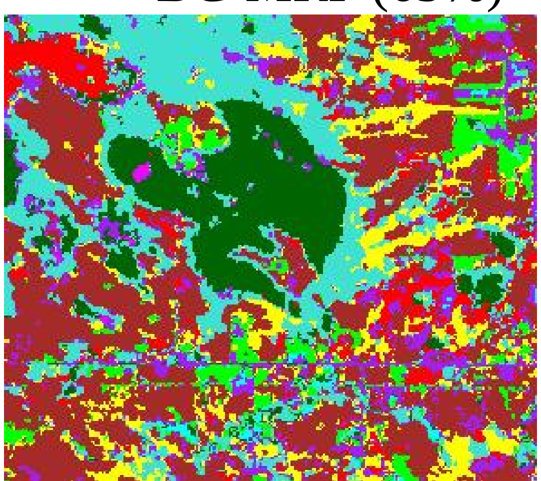


BC (60%)

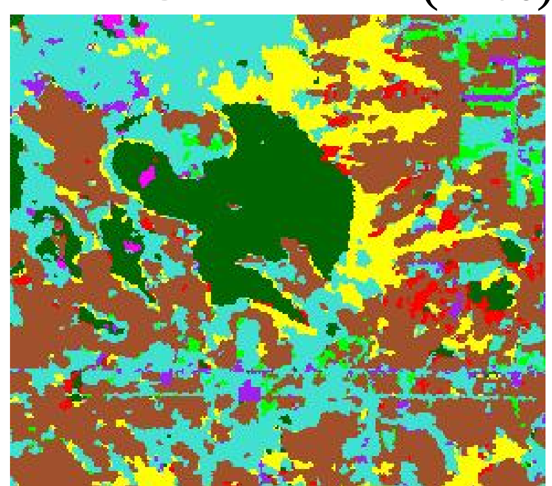


BC-EM (68%)

BC-MRF (65%)



BC-EM-MRF (72%)



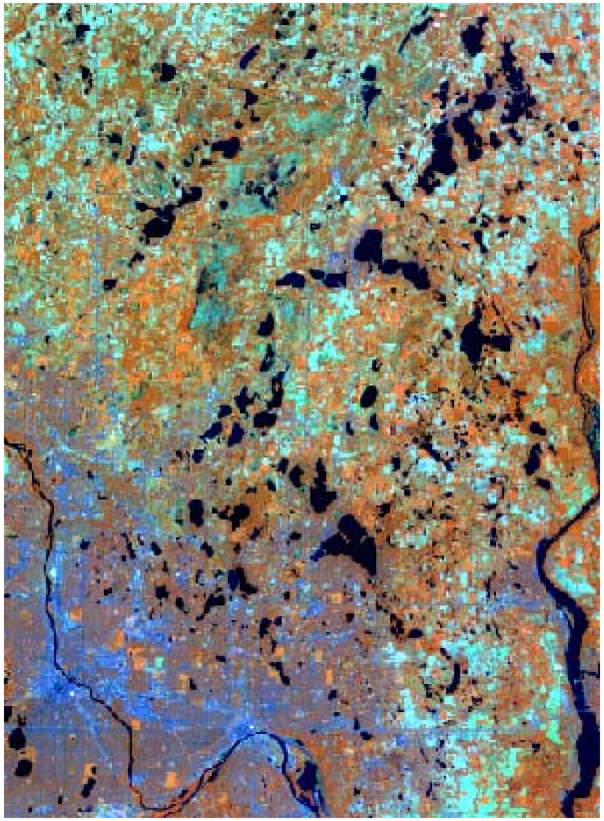
Open Research Problems

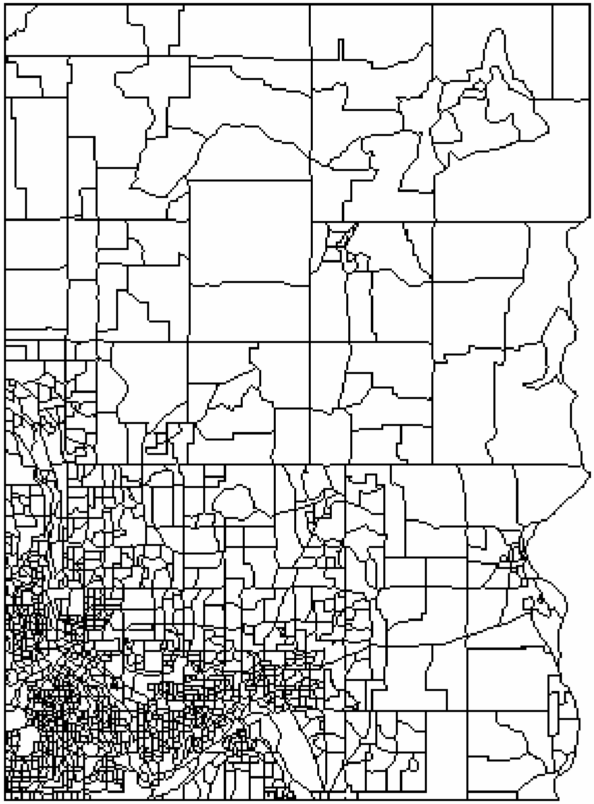
- Care should be exercised when collecting unlabeled samples
  - No samples from small classes
  - Mixed plots
  - Samples from unknown components

- Extend the model for multi-source data

Open Research Problems

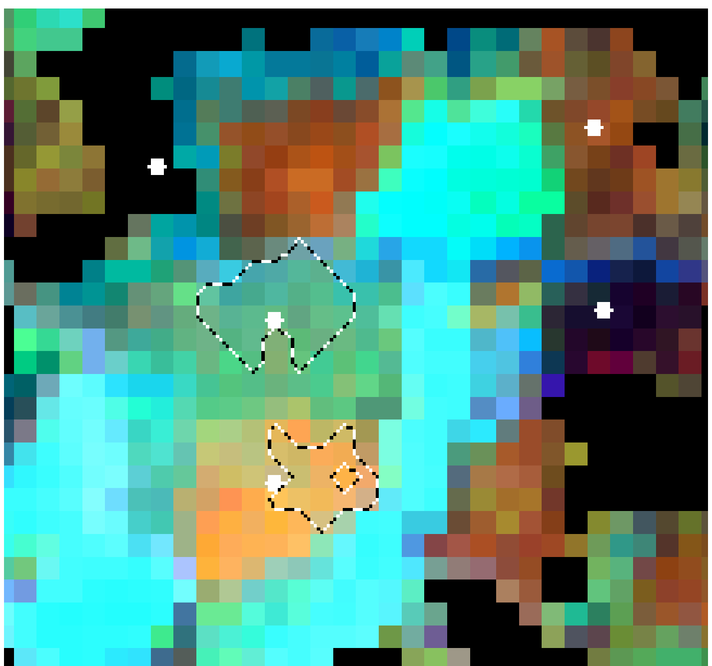
- Extend the model for multi-source data

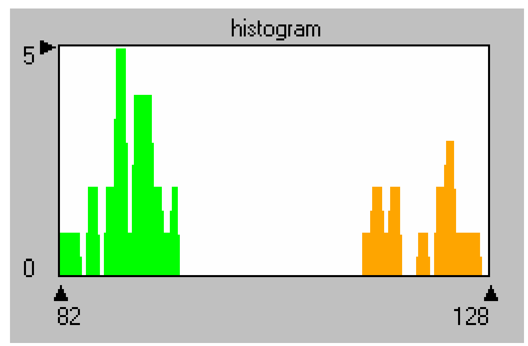


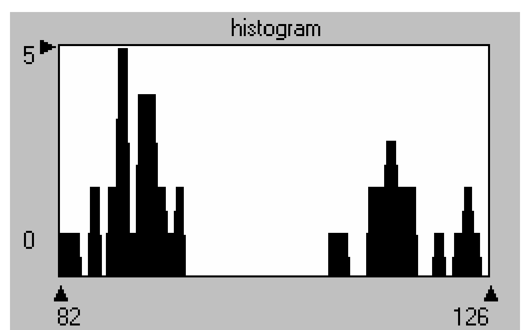


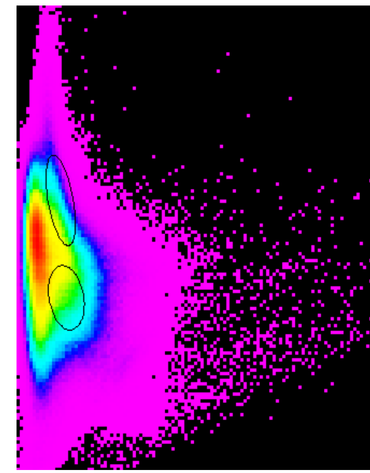
Open Research Problems

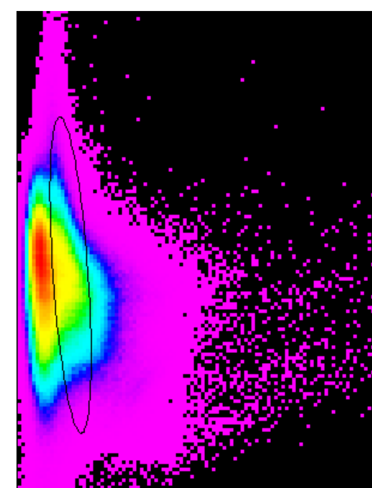
- Discovering sub-components (e.g., Anderson Level 2 and 3 classes)











Open Research Problems

- Efficient Algorithms
- Global Optimization (Stochastic versions)
- Semi-supervised MCC
- Semi-supervised + Active Learning

- Not a research problem, but
  - Can we create bench mark dataset(s) similar to CMU or UCI for Remote Sensing Data Mining.

- Conclusions
  - Results were promising

Acknowledgements

- Prof. Joydeep Ghosh, UT Austin
- Team @ Remote Sensing & Geospatial Analysis Lab
- Team @ Spatial Databases Research Group, C.Sci. Dept. U of Minnesota.