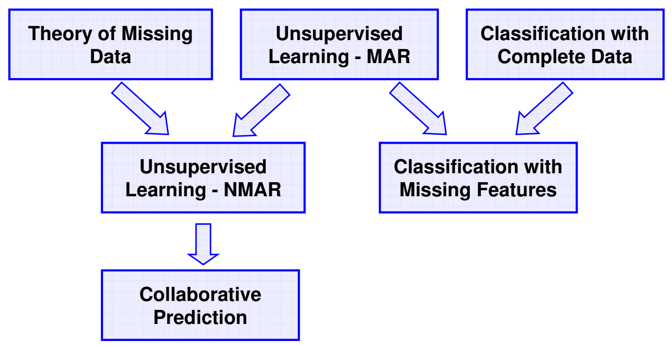
Missing Data Problems in Machine Learning

Senate Thesis Defense

Ben Marlin
Machine Learning Group
Department of Computer Science
University of Toronto
April 8, 2008

Contents:

Overview:



Basic Notation

N	Number of data cases.
D	Number of data dimensions.
C	Number of classes.
V	Number of multinomial values.
K	Number of clusters or hidden units.

Basic Notation for Missing Data

$\mathbf{x}_n$	0.1 | 0.9 | 0.2 | 0.7 | 0.3	Data Vector
$\mathbf{r}_n$	1 | 0 | 0 | 1 | 1	Response Vector
$\mathbf{o}_n$	1 | 4 | 5	Observed Dimensions
$\mathbf{m}_n$	2 | 3	Missing Dimensions
$\mathbf{x}_n^{\mathbf{o}_n}, \mathbf{x}_n^{o}$	0.1 | 0.7 | 0.3	Observed Data
$\mathbf{x}_n^{\mathbf{m}_n}, \mathbf{x}_n^{m}$	0.9 | 0.2	Missing Data

Theory of Missing Data: Overview

Background on the Theory of Missing Data (Little/Rubin):

- Factorizations of the generative process
- Three classes of missing data
  - Missing Completely at Random (MCAR)
  - Missing at Random (MAR)
  - Not Missing at Random (NMAR)
- The effect of each class of missing data on inference

Extensions and Elaborations:

- MAR assumption, multivariate data, and symmetry
- MAR assumption and model misspecification

Theory of Missing Data: Factorizations

Data/Selection Model Factorization:

$$P(\mathbf{X},\mathbf{R},\mathbf{Z}|\theta,\mu) = P(\mathbf{R}|\mathbf{X},\mathbf{Z},\mu)P(\mathbf{X},\mathbf{Z}|\theta)$$

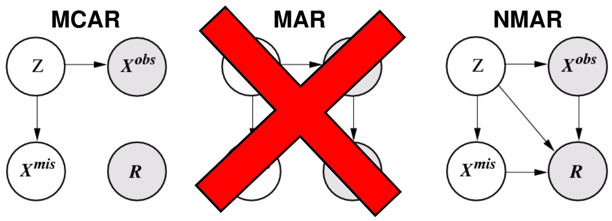
• The probability of selection depends on the true values of the data variables and latent variables.

Pattern Mixture Model Factorization:

$$P(\mathbf{X},\mathbf{R},\mathbf{Z}|\vartheta,\nu) = P(\mathbf{X},\mathbf{Z}|\mathbf{R},\vartheta)P(\mathbf{R}|\nu)$$

• Each response vector defines a different pattern, and each pattern has a different distribution over the data.

Theory of Missing Data: Classification



MCAR: $P(\mathbf{R}|\mathbf{X},\mathbf{Z},\mu) = P(\mathbf{R}|\mu)$

MAR: $P(\mathbf{R}|\mathbf{X},\mathbf{Z},\mu) = P(\mathbf{R}|X^{obs},\mu)$

NMAR: No simplification in general.

Theory of Missing Data: Classification

What Does it mean to be Missing at Random?

• MAR is ***not*** a statement of independence between random variables. MAR requires that particular symmetries hold so that P(R=r|X=x) can be determined from observed data only.

$X\backslash R$	0 0	0 1	1 0	1 1
0 0	α	β	γ	$1-\alpha-\beta-\gamma$
0 1	α	δ	γ	$1-\alpha-\delta-\gamma$
1 0	α	β	λ	$1-\alpha-\beta-\lambda$
1 1	α	δ	λ	$1-\alpha-\delta-\lambda$

Theory of Missing Data: Inference

MCAR/MAR Posterior:

$$P(\theta|\mathbf{x}^o,\mathbf{r}) \propto \int\int\int P(\mathbf{X},\mathbf{Z}|\theta)P(\mathbf{R}|\mathbf{X},\mu)P(\theta|\omega)P(\mu|\eta)d\mu dZ d\mathbf{x}^m$$
$$\propto P(\mathbf{X}^o=\mathbf{x}^o|\theta)P(\theta|\omega)$$

NMAR Posterior:

$$P(\theta|\mathbf{x}^o,\mathbf{r}) \propto \int\int\int P(\mathbf{X},\mathbf{Z}|\theta)P(\mathbf{R}|\mathbf{X},\mathbf{Z},\mu)P(\theta|\omega)P(\mu|\eta)d\mu dZ d\mathbf{x}^m$$

• When data is NMAR, the selection model can not be ignored. Doing so will “bias” inference, learning, and prediction.

Theory of Missing Data: Misspecification

Misspecified Missing Data Model, NMAR Missing Data:

• If missing data is NMAR, it is not sufficient to use any missing data model. Inference is still biased if the wrong missing data model is used.

Misspecified Data Model, MAR Missing Data:

• Even if missing data is MAR with respect to the underlying generative process, inference for the parameters of a simpler data model can still be biased.

Theory of Missing Data: Misspecification

Misspecified Data Model, MAR Missing Data:

• Consider a 2D binary example where the true data model is the full four element CPT, and we approximate it using a product of the two marginal distributions.

• Suppose the missing data model is MAR, and our goal is to estimate the marginal P(X1=1).

x	$P(x)$	$P(R=[0,0]\|x)$	$P(R=[0,1]\|x)$	$P(R=[1,0]\|x)$	$P(R=[1,1]\|x)$
0 0	a	α	β	γ	$1-\alpha-\beta-\gamma$
0 1	b	α	δ	γ	$1-\alpha-\delta-\gamma$
1 0	c	α	β	λ	$1-\alpha-\beta-\lambda$
1 1	d	α	δ	λ	$1-\alpha-\delta-\lambda$

Theory of Missing Data: Misspecification

Misspecified Data Model, MAR Missing Data:

- Suppose we estimate P(X1=1) under the marginal model, and under the true model.
- We can show that Computing P(X1=1) under the marginal model is equal to computing P(X1=1| R1=1).
- We can further prove that P(X1=1) is only equal to P(X1=1| R1=1) if $\beta = \delta$. This corresponds to the MCAR condition.

Theory of Missing Data: Misspecification

Misspecified Data Model, MAR Missing Data:

$$a = 0.1 \quad c = 0.7 \quad \alpha = 0.1 \quad \delta = 0.1$$
$$b = 0.1 \quad d = 0.1 \quad \beta = 0.1 + t0.05 \quad \gamma = 0.2$$

$\beta - \delta$	True $P(X_1 = 1)$	Est. $P(X_1 = 1)$	True $P(X_1 = 1 \mid R_1 = 1)$	Est. $P^M(X_1 = 1)$
0.05	0.8000	0.7999 ± 0.0007	0.7961	0.7961 ± 0.0007
0.10	0.8000	0.8004 ± 0.0006	0.7917	0.7923 ± 0.0006
0.15	0.8000	0.7996 ± 0.0006	0.7868	0.7860 ± 0.0007
0.20	0.8000	0.8011 ± 0.0007	0.7812	0.7826 ± 0.0008
0.25	0.8000	0.7990 ± 0.0007	0.7750	0.7737 ± 0.0008
0.30	0.8000	0.8000 ± 0.0007	0.7679	0.7679 ± 0.0007
0.35	0.8000	0.7994 ± 0.0008	0.7596	0.7582 ± 0.0009
0.40	0.8000	0.7999 ± 0.0009	0.7500	0.7501 ± 0.0010
0.45	0.8000	0.7992 ± 0.0010	0.7386	0.7379 ± 0.0010
0.50	0.8000	0.7986 ± 0.0010	0.7250	0.7241 ± 0.0010

Unsupervised Learning – MAR: Overview

Background on Unsupervised Learning With Random Missing Data:

- Finite Bayesian Multinomial Mixture (MAP EM)
- Dirichlet Process Multinomial Mixture (Gibbs)
- Finite Factor Analysis/PPCA Mixture (ML EM)
- Restricted Boltzmann Machines (Contrastive Divergence)

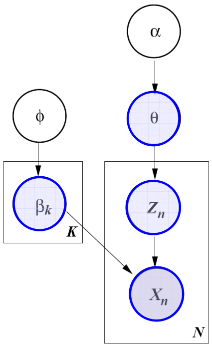
Extensions and Elaborations:

- Collapsed Gibbs sampler for DPMM with missing data
- Derivation of factor analysis mixture with missing data
- New view of RBM models for missing data

Finite Bayesian Mixture: Model

Probability Model:

$$P(Z_n = k|\theta) = \theta_k$$
$$P(\mathbf{X}_n = \mathbf{x}_n | Z_n = k, \beta) = P(\mathbf{x}_n | \beta_k)$$
$$P(\theta, \beta | \alpha, \phi) = P(\theta|\alpha) \prod_k P(\beta_k | \phi)$$



Properties:

- Allows for a fixed, finite number of clusters.
- In the multinomial mixture, $P(x_n|\beta_k)$ is a product of discrete distributions. The prior on β and θ is Dirichlet.

Finite Bayesian Mixture: Model

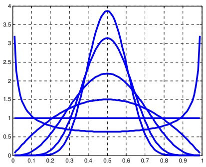
Dirichlet Distribution:

Bayesian mixture modeling becomes much easier when conjugate priors are used for the model parameters. The conjugate prior for the mixture proportions θ is the Dirichlet distribution.

$$P(\theta|\alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1}$$

$$E[\theta_k|\alpha] = \frac{\alpha_k}{\sum_{k=1}^K \alpha_k}$$

$$P(\theta|\alpha, \mathbf{z}) = \frac{\Gamma(N + \sum_k \alpha_k)}{\prod_k \Gamma(C_k + \alpha_k)} \prod_k \theta_k^{C_k + \alpha_k - 1}$$



Finite Bayesian Mixture: Learning

MAP EM Algorithm:

$$\text{E-Step: } q_n(k) \leftarrow \frac{\theta_k \prod_{d=1}^D \prod_{v=1}^V \beta_{vdk}^{[r_{dn}=1][x_{dn}=v]}}{\sum_{k'=1}^K \theta_{k'} \prod_{d=1}^D \prod_{v=1}^V \beta_{vdk'}^{[r_{dn}=1][x_{dn}=v]}}$$

$$\text{M-Step: } \theta_k \leftarrow \frac{\alpha_k - 1 + \sum_{n=1}^N q_n(k)}{N - K + \sum_{k=1}^K \alpha_k}$$

$$\beta_{vdk} \leftarrow \frac{\phi_{vdk} - 1 + \sum_{n=1}^N q_n(k)[r_{dn} = 1][x_{dn} = v]}{\sum_{n=1}^N q_n(k)[r_{dn} = 1] - V + \sum_{v=1}^V \phi_{vdk}}$$

Finite Bayesian Mixture: Prediction

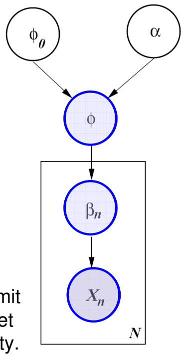
Predictive Distribution:

$$P(x_{dn} = v|\mathbf{x}_n, \mathbf{r}_n, \beta, \theta)$$
$$= \sum_{k=1}^{K} P(x_{dn} = v|z_n = k, \beta)P(z_n = k|\mathbf{x}_n, \mathbf{r}_n, \beta, \theta)$$
$$= \sum_{k=1}^{K} \beta_{vdk} \frac{\theta_k \prod_{d=1}^{D} \prod_{v=1}^{V} \beta_{vdk}^{[r_{dn}=1][x_{dn}=v]}}{\sum_{k'=1}^{K} \theta_{k'} \prod_{d=1}^{D} \prod_{v=1}^{V} \beta_{vdk'}^{[r_{dn}=1][x_{dn}=v]}}$$

Dirichlet Process Mixture: Model

Probability Model:

$$P(\phi|\phi_0, \alpha) = \mathcal{DP}(\alpha, \phi_0)$$
$$P(\beta_n|\phi) = \phi(\beta_n)$$
$$P(\mathbf{X}_n = \mathbf{x}_n|\beta) = P(\mathbf{x}_n|\beta_n)$$



Properties:

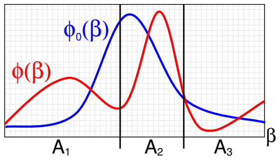
- Since ϕ is discrete, the DPM can be viewed as a countably infinite mixture model.
- Another way to arrive at a DPM is to consider the limit of a Bayesian mixture model with symmetric Dirichlet prior as the number for components K goes to infinity.

Dirichlet Process Mixture: DP

The Dirichlet Process:

Let α be a scalar and ϕ_0 be a distribution on a random variable β. A random distribution ϕ is a draw from DP(α,ϕ_0) if and only if for any K and any K-partition $A_1,\ldots,A_K$ of the space of β, the distribution over elements of the partition induced by ϕ is given by the Dirichlet distribution:

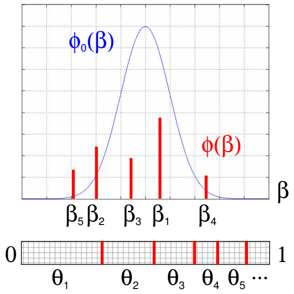
$$(\phi(\beta \in A_1), ..., \phi(\beta \in A_K)) = Dir(\alpha\phi_0(A_1), ..., \alpha\phi_0(A_K))$$



Dirichlet Process Mixture: DP

Understanding ϕ:

- The only ϕ that satisfy the definition of the DP are discrete with probability 1.
- Drawing samples β_n from the DP is easy if ϕ_0 can be sampled easily (Blackwell/McQueen).
- The "stick breaking" view of the DP prior lets us incrementally construct a draw ϕ from $DP(\alpha,\phi_0)$.



$$P(\beta_N|\alpha, \phi_0, \beta_1, ..., \beta_{N-1})$$
$$= \frac{\alpha\phi_0(\beta_N)}{N-1+\alpha} + \frac{\sum_{n=1}^{N} \delta_{\beta_n}(\beta_N)}{N-1+\alpha}$$

Dirichlet Process Mixture: Inference

Collapsed Gibbs Sampler With Missing Data:

$$P(z_n = k, \exists i \neq n \;\; z_i = k|z_{-n}, \mathbf{x}, \alpha, \phi_0)$$
$$\propto \frac{c_k^{-n}}{N-1+\alpha} \prod_{d=1}^{D} \prod_{v=1}^{V} \left(\frac{c_{vdk}^{-n} + \phi_{vd0}}{\sum_{v=1}^{V} c_{vdk}^{-n} + \phi_{vd0}} \right)^{[r_{dn}=1][x_{dn}=v]}$$

Dirichlet Process Mixture: Prediction

Predictive Distribution (Training Cases):

$$\beta_{vdk}^{s} = \frac{\phi_{v0} + c_{vdk}}{\sum_{v=1}^{V} \phi_{v0} + c_{vdk}}$$

$$P(x_{dn} = v|\{\mathbf{x}_n, \mathbf{r}_n\}_{n=1:N}, \phi_0, \alpha) = \frac{1}{S} \sum_{s=1}^{S} \sum_{k=1}^{K^s} [z_n^s = k]\beta_{vdk}^{s}$$

Dirichlet Process Mixture: Prediction

Predictive Distribution (Test Cases):

$$\hat{\theta}_k^s = \begin{cases} \frac{\sum_{n=1}^N [z_n = k]}{N+\alpha} & \ldots\ k \leq K^s \\ \frac{\alpha}{N+\alpha} & \ldots\ k = K^s + 1 \end{cases}$$

$$\hat{\beta}_{vdk}^s = \begin{cases} \beta_{vdk}^s & \ldots\ k \leq K^s \\ \frac{\phi_{vd0}}{\sum_{v=1}^V \phi_{vd0}} & \ldots\ k = K^s + 1 \end{cases}$$

$$P(x_{d*} = v | \mathbf{x}_*, \mathbf{r}_*, \beta, \theta)$$
$$= \frac{1}{S} \sum_{s=1}^{S} \sum_{k=1}^{K^s+1} \hat{\beta}_{vdk}^s \frac{\hat{\theta}_k \prod_{d=1}^D \prod_{v=1}^V \hat{\beta}_{vdk}^{[r_{d*}=1][x_{d*}=v]}}{\sum_{k'=1}^K \hat{\theta}_{k'} \prod_{d=1}^D \prod_{v=1}^V \hat{\beta}_{vdk'}^{[r_{d*}=1][x_{d*}=v]}}$$

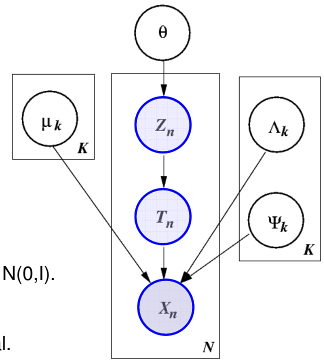
FA/PPCA Mixtures: Model

Probability Model:

$$P(z_n = k) = \theta_k$$

$$P(\mathbf{t}_n) = \frac{1}{(2\pi)^{Q/2}} \exp(-\frac{1}{2}\mathbf{t}_n^T \mathbf{t}_n)$$

$$P(\mathbf{x}_n | z_n = k, \mathbf{t}_n, \Lambda, \Psi, \mu)$$
$$= \mathcal{N}(\mu_k + \Lambda_k \mathbf{t}_n, \Psi_k)$$



Properties:

- T_n is a length Q real-valued vector ~ N(0,I).
- Factor loading matrix Λ is DxQ.
- Covariance matrix Ψ is DxD diagonal.

FA/PPCA Mixtures: Model

Joint Distribution of X and T given Z:

$$P\left([\mathbf{x}_n, \mathbf{t}_n]^T | z_n = k, \mu, \Lambda, \Psi_k\right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_k \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \Psi_k + \Lambda_k \Lambda_k^T & \Lambda_k \\ \Lambda_k^T & I \end{bmatrix}\right)$$

Marginal Distribution of X given Z:

$$P(\mathbf{x}_n | z_n = k, \Lambda, \sigma, \mu) = \mathcal{N}(\mathbf{x}_n; \mu_k, \Psi_k + \Lambda_k \Lambda_k^T)$$

FA/PPCA Mixtures: Learning

M-Step Inference with Missing Data:

$$q_n(\mathbf{t}_n, \mathbf{x}_n^m, z_n) = P(\mathbf{t}_n, \mathbf{x}_n^m | z_n, \mathbf{x}_n^o, \mu, \Lambda, \Psi) P(z_n | \mathbf{x}_n^o, \mu, \Lambda, \Psi, \theta)$$
$$= q_{nz_n}(\mathbf{t}_n, \mathbf{x}_n^m) q_n(z_n)$$

$$q_n(k) \propto \theta_k \mathcal{N}(\mu_k^o, \Psi_k^o + \Lambda_k^o \Lambda_k^{oT})$$

$$q_{nk}(\mathbf{t}_n, \mathbf{x}_n^m) = \mathcal{N}\left(\begin{bmatrix} \mu_{t_n|x_n^o k} \\ \mu_{x_n^m|x_n^o k} \end{bmatrix}, \begin{bmatrix} \Sigma_{t_n|x_n^o k} & \Sigma_{t_n x_n^m|x_n^o k} \\ \Sigma_{t_n x_n^m|x_n^o k}^T & \Sigma_{t_n|x_n^o k} \end{bmatrix}\right)$$

FA/PPCA Mixtures: Learning

E-Step Updates with Missing Data:

$$\theta_k = \frac{\sum_{n=1}^N q_n(k)}{N}$$

$$\mu_k = \frac{1}{\sum_{n=1}^N q_n(k)} \sum_{n=1}^N q_n(k)(E_{q_{nk}}[\mathbf{x}_n] - \Lambda_k E_{q_{nk}}[t_n])$$

$$\Lambda_k = \left(\sum_{n=1}^N q_n(k)(E_{q_{nk}}[\mathbf{x}_n \mathbf{t}_n^T] - \mu_k E_{q_{nk}}[\mathbf{t}_n]^T)\right) \left(\sum_{n=1}^N q_n(k) E_{q_{nk}}[\mathbf{t}_n \mathbf{t}_n^T]\right)^{-1}$$

$$\Psi_{dd} = \frac{1}{\sum_{n=1}^N q_n(k)} \sum_{n=1}^N q_n(k) \left(E_{q_{nk}}[\mathbf{x}_n \mathbf{x}_n^T]_{dd} + \mu_{dk}^2 + \Lambda_{d:k} E_{q_{nk}}[\mathbf{t}_n \mathbf{t}_n^T] \Lambda_{d:k}^T\right.$$
$$\left. -2\mu_{dk} E_{q_{nk}}[\mathbf{x}_n]_d - 2\Lambda_{d:k} E_{q_{nk}}[\mathbf{x}_n \mathbf{t}_n^T]_{d:}^T + 2\mu_{dk} \Lambda_{d:k} E_{q_{nk}}[\mathbf{t}_n]\right)$$

FA/PPCA Mixtures: Prediction

Predictive Distribution:

$$P(\mathbf{x}_n^m | \mathbf{x}_n^o, \Lambda, \mu, \Psi) = \sum_{k=1}^K q_n(k) \mathcal{N}(\mu_{x_n^m|x_n^o k}, \Sigma_{x_n^m|x_n^o k})$$

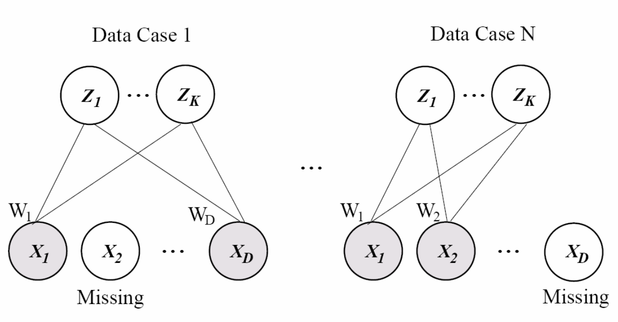
$$q_n(k) \propto \theta_k \frac{1}{|2\pi \Sigma_{xxk}^{oo}|} \exp(-\frac{1}{2}(\mathbf{x}_n^o - \mu_k^o)^T (\Sigma_{xxk}^{oo})^{-1} (\mathbf{x}_n^o - \mu_k^o))$$

$$\mu_{x_n^m|x_n^o k} = \mu_k^m + \Sigma_{xxk}^{mo} (\Sigma_{xxk}^{oo})^{-1} (\mathbf{x}_n^o - \mu_k^o)$$

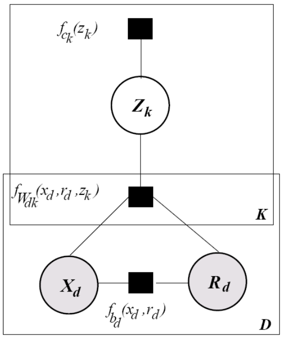
$$\Sigma_{x_n^m|x_n^o k} = \Sigma_{xxk}^{mm} - \Sigma_{xxk}^{mo} (\Sigma_{xxk}^{oo})^{-1} \Sigma_{xxk}^{om}$$

$$\Sigma_{xxk} = \Psi_k + \Lambda_k \Lambda_k^T$$

Conditional RBM's: Model



Conditional RBM's: Model



Conditional RBM's: Energy Function

Probability Model:

$$P(\mathbf{x}_n^o, \mathbf{z}_n | \mathbf{r}_n) = \frac{\exp(-E(\mathbf{x}_n^o, \mathbf{z}_n, \mathbf{r}_n))}{\sum_{\mathbf{x}^o} \sum_{\mathbf{z}} \exp(-E(\mathbf{x}^o, \mathbf{z}, \mathbf{r}_n))}$$

$$\begin{aligned} E(\mathbf{x}_n^o, \mathbf{z}_n, \mathbf{r}_n) = & -\sum_{d=1}^{D} \sum_{v=1}^{V} \sum_{k=1}^{K} W_{vdk}[x_{dn} = v][z_{kn} = 1][r_{dn} = 1] \\ & -\sum_{d=1}^{D} \sum_{v=1}^{V} b_{vd}[x_{dn} = v][r_{dn} = 1] \\ & -\sum_{k=1}^{K} c_k[z_{kn} = 1] \end{aligned}$$

Conditional RBM's: Inference

Gibbs Sampler:

$$\begin{aligned} & P(x_d = v | \mathbf{z}_n, W, b) \\ & = \frac{\exp(\sum_{k=1}^{K} W_{vdk}[z_{kn} = 1] + b_{vd})}{\sum_{v=1}^{V} \exp(\sum_{k=1}^{K} W_{vdk}[z_k = 1] + b_{vd})} \end{aligned}$$

$$\begin{aligned} & P(z_k = 1 | \mathbf{x}_n^o, \mathbf{r}_n, W, c) \\ & = \frac{1}{1 + \exp(-(\sum_{d=1}^{D} \sum_{v=1}^{V} W_{vdk}[x_{dn} = v][r_{dn} = 1] + c_k))} \end{aligned}$$

Conditional RBM's: Learning

Contrastive Divergence Gradients:

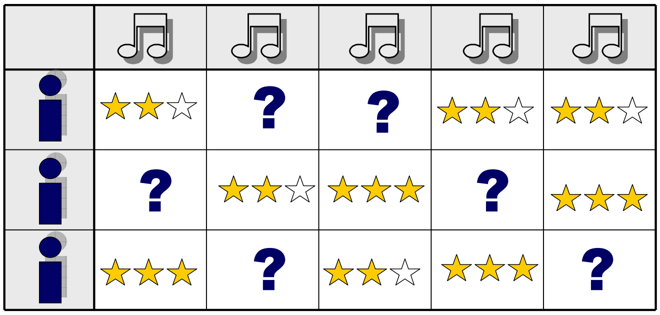
$$\begin{aligned} \frac{\partial \mathcal{C}}{\partial W_{vdk}} \approx & \sum_{n=1}^{N} [x_{dn} = v][r_{dn} = 1] P(\mathbf{z}_k = 1 | \mathbf{x}_n) \\ & - \sum_{n=1}^{N} [x_{dn}^T = v][r_{dn} = 1] P(\mathbf{z}_k = 1 | \mathbf{x}_n^T) \end{aligned}$$

Unsupervised Learning – NMAR: Overview

Data Sets and Experimental Protocols:

- Conducted first ever survey of user rating behavior in a recommender system.
- Collected first collaborative filtering data set that includes both ratings for user-selected items and ratings for randomly selected items.
- Designed new experimental protocols for collaborative prediction to test methods that assume MAR vs methods that model NMAR effects.

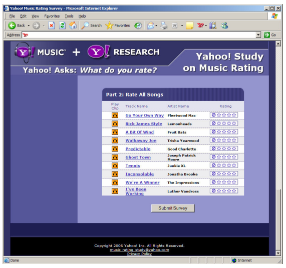
Unsupervised Learning – NMAR:
Collaborative Prediction Problem



Unsupervised Learning – NMAR:
Data Sets: Yahoo!

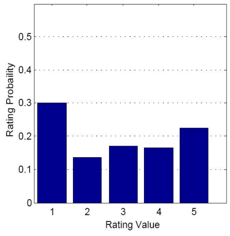
Data was collected through an online survey of Yahoo! Music LaunchCast radio users.

- 1000 songs selected at random.
- Users rate 10 songs selected at random from 1000 songs.
- Answer 16 questions.
- Collected data from 35,000+ users.



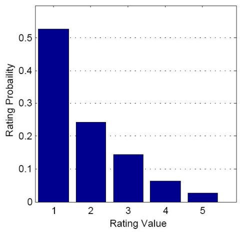
Unsupervised Learning – NMAR:
Data Sets: Yahoo!

User Selected



$$P^U_E(x = v | r = 1)$$

Randomly Selected

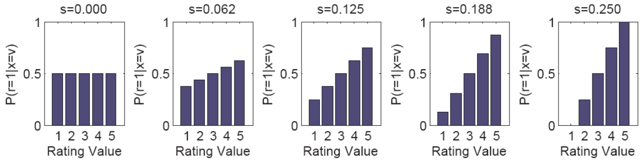


$$P^R_E(x = v)$$

Unsupervised Learning – NMAR:
Data Sets: Jester

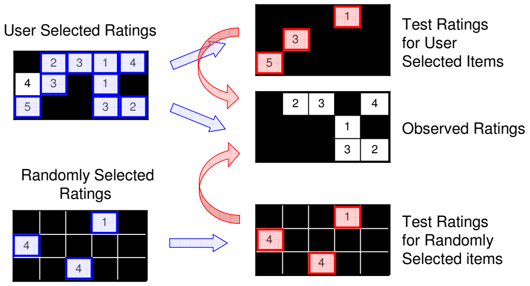
Jester gauge set of 10 jokes used as complete data. Synthetic missing data was added.

- 15,000 users randomly selected
- Missing data model: $\mu_v(s) = s(v-3)+0.5$

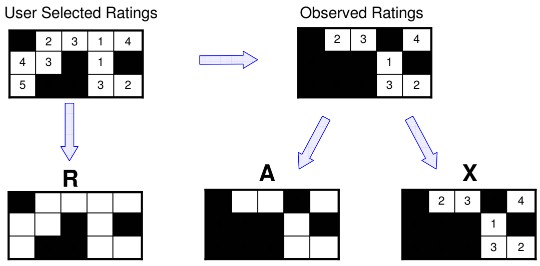


Unsupervised Learning – NMAR:
Data Sets: User Splits

Unsupervised Learning – NMAR:
Data Sets: Rating Splits



Unsupervised Learning – NMAR: Data Sets: Connection to Notation



Unsupervised Learning – NMAR: Experimental Protocols

Weak Generalization:

- Learn on training user observed ratings.
- Evaluate on training user test ratings for user selected items, and training user test ratings for randomly selected items.

Strong Generalization:

- Learn on training user observed ratings.
- Evaluate on test user test ratings for user selected items, and test user test ratings for randomly selected items.

Unsupervised Learning – NMAR: Models

- Follow a modeling strategy based on combining probabilistic models for complete data with simple models of the missing data process.
- Consider complete data models including finite Bayesian mixtures, Dirichlet Process mixtures, and RBM's.
- Consider two basic missing data models: CPT-v and LOGIT-vd.

Finite Mixture/CPT-v: Model

Probability Model:

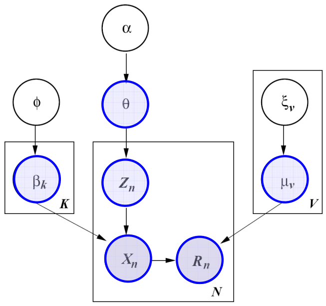
$$P(\theta|\alpha) = \mathcal{D}(\theta|\alpha)$$

$$P(\beta|\phi) = \prod_{k=1}^{K}\prod_{d=1}^{D}\mathcal{D}(\beta_{dk}|\phi_{dk})$$

$$P(\mathbf{X} = \mathbf{x}_n|Z_n = k, \beta) = \prod_{d=1}^{D}\prod_{v=1}^{V}\beta_{vdk}^{[x_{dn}=v]}$$

$$P(\mu|\xi) = \prod_{v}\mathcal{B}(\mu_v|\xi_v)$$

$$P(\mathbf{R} = \mathbf{r}_n|\mathbf{X} = \mathbf{x}_n, \mu) = \prod_{d=1}^{D}\prod_{v=1}^{V}\mu_v^{[r_{dn}=1][x_{dn}=v]}(1-\mu_v)^{[r_{dn}=0][x_{im}=v]}$$



Finite Mixture/CPT-v: Identifiability

Identifiability:

$$w_{dn} = \begin{cases} x_{dn} & r_{dn} = 1 \\ \varnothing & r_{dn} = 0 \end{cases}$$

$$\phi_{\mathbf{w}} = \sum_{k=1}^{K}\theta_k\prod_{d=1}^{D}\left(\prod_{v=1}^{V}(\mu_v\beta_{vdk})^{[w_d=v]}\right)^{[w_d\neq\varnothing]}\left(\sum_{v=1}^{V}(1-\mu_v)\beta_{vdk}\right)^{[w_d=\varnothing]}$$

2D Binary Example:

$$\phi_{11} = \sum_{k=1}^{K}\theta_k\mu_1\beta_{11k}\mu_1\beta_{12k} \qquad \phi_{21} = \sum_{k=1}^{K}\theta_k\mu_2\beta_{21k}\mu_1\beta_{12k}$$

$$\phi_{1\varnothing} = \sum_{k=1}^{K}\theta_k\mu_1\beta_{11k}(\sum_{v=1}^{V}(1-\mu_v)\beta_{v2k}) \qquad \phi_{2\varnothing} = \sum_{k=1}^{K}\theta_k\mu_2\beta_{21k}(\sum_{v=1}^{V}(1-\mu_v)\beta_{v2k})$$

Finite Mixture/CPT-v: Identifiability

2D Binary Example:

$$\begin{bmatrix} (\phi_{11}+\phi_{12}+\phi_{1\varnothing}) & (\phi_{21}+\phi_{22}+\phi_{2\varnothing}) \\ (\phi_{11}+\phi_{21}+\phi_{\varnothing 1}) & (\phi_{12}+\phi_{22}+\phi_{\varnothing 2}) \end{bmatrix}\begin{bmatrix} \frac{1}{\mu_1} \\ \frac{1}{\mu_2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- This system will have a unique solution for μ_1 and μ_2 if both are greater than 0, and the matrix Φ of sums of ϕ_w coefficients is non-singular.
- This result is easily extended to the general case of D dimensions and V multinomial values.

Finite Mixture/CPT-v: Learning

MAP EM Algorithm (E-Step):

$$q_n(k) = P(z_n = k|\mathbf{x}_n^o, \mathbf{r}_n, \mathbf{a}_n, \theta, \beta, \mu) = \frac{\theta_k \prod_{d=1}^D \gamma_{dkn}}{\sum_{k=1}^K \theta_k \prod_{d=1}^D \gamma_{dkn}}$$

$$q_n(k, v, d) = P(z_n = k, x_{dn} = v|\mathbf{x}_n^o, \mathbf{r}_n, \mathbf{a}_n, \theta, \beta, \mu)$$

$$= q_n(k) \left(\frac{\mu_v \beta_{vdk}}{\sum_{v'=1}^V \mu_{v'} \beta_{v'dk}} \right)^{[r_{dn}=1][a_{dn}=0]} \left(\frac{(1-\mu_v)\beta_{vdk}}{\sum_{v'=1}^V (1-\mu_{v'})\beta_{v'dk}} \right)^{[r_{dn}=0][a_{dn}=0]}$$

$$\gamma_{dkn} = \left(\prod_{v=1}^V (\beta_{vdk}\mu_v)^{[x_{dn}=v]} \right)^{[r_{dn}=1][a_{dn}=1]} \left(\prod_{v=1}^V (\beta_{vdk}(1-\mu_v))^{[x=v]} \right)^{[r_{dn}=0][a_{dn}=1]}$$

$$\cdot \left(\sum_{v=1}^V \beta_{vdk}\mu_v \right)^{[r_{dn}=1][a_{dn}=0]} \left(\sum_{v=1}^V \beta_{vdk}(1-\mu_v) \right)^{[r_{dn}=0][a_{dn}=0]}$$

Finite Mixture/CPT-v: Learning

MAP EM Algorithm (M-Step):

$$\theta_k = \frac{\alpha_k - 1 + \sum_{n=1}^N q_n(k)}{N - K + \sum_{k=1}^K \alpha_k}$$

$$\beta_{vdk} = \frac{\phi_{vdk} - 1 + \sum_{n=1}^N q_n(k)[a_{dn}=1][x_{dn}=v] + q_n(k,v,d)[a_{dn}=0]}{\sum_{n=1}^N q_n(k) - V + \sum_{v=1}^V \phi_{vdk}}$$

$$\mu_v = \frac{\xi_{1v} - 1 + \sum_{n=1}^N \sum_{d=1}^D [r_{dn}=1][x_{dn}=v] + q_n(v,d)[r_{dn}=1][a_{dn}=0]}{\xi_{1v} + \xi_{0v} - 2 + \sum_{n=1}^N \sum_{d=1}^D [r_{dn}=1][x_{dn}=v] + q_n(v,d)[a_{dn}=0]}$$

Finite Mixture/CPT-v: Prediction

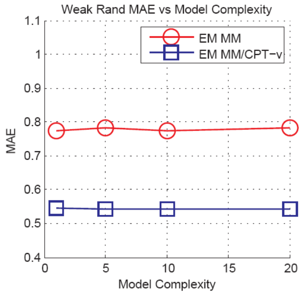
Predictive Distribution

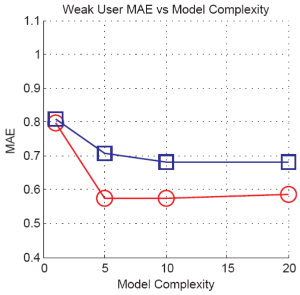
$$P(X_{dn} = v|\mathbf{x}_n, \mathbf{r}_n, \mathbf{a}_n, \theta, \beta, \mu)$$

$$= \sum_{k=1}^K \frac{\theta_k \prod_{d=1}^D \gamma_{dkn}}{\sum_{k=1}^K \theta_k \prod_{d=1}^D \gamma_{dkn}} \left(\frac{\mu_v \beta_{vdk}}{\sum_{v'=1}^V \mu_{v'} \beta_{v'dk}} \right)^{[r_{dn}=1]} \left(\frac{(1-\mu_v)\beta_{vdk}}{\sum_{v'=1}^V (1-\mu_{v'})\beta_{v'dk}} \right)^{[r_{dn}=0]}$$

Finite Mixture/CPT-v: Results

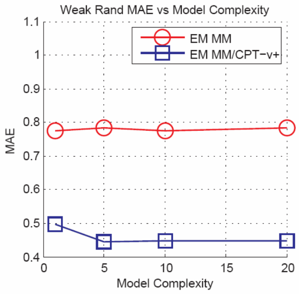
Yahoo! Weak Generalization Results: MM vs MM/CPT-v

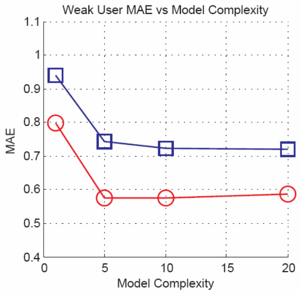




Finite Mixture/CPT-v: Results

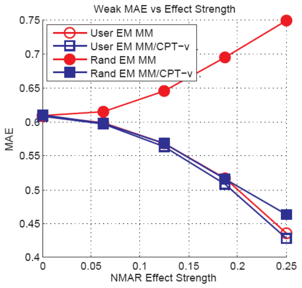
Yahoo! Weak Generalization Results: MM vs MM/CPT-v+

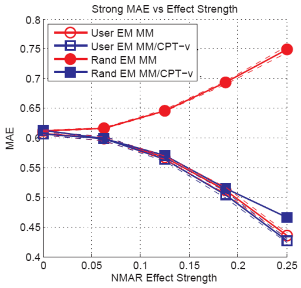




Finite Mixture/CPT-v: Results

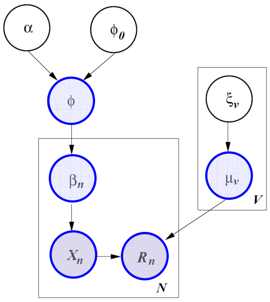
Jester Results: MM vs MM/CPT-v





DP Mixture/CPT-v: Model

Probability Model:



$$P(\phi|\phi_0,\alpha) = \mathcal{DP}(\alpha\phi_0)$$

$$P(\beta|\phi_0) = \prod_{d=1}^{D} \mathcal{D}(\beta_d|\phi_{d0})$$

$$P(\mathbf{X}=\mathbf{x}_n|\beta_n) = \prod_{d=1}^{D}\prod_{v=1}^{V} \beta_{vdk}^{[x_{dn}=v]}$$

$$P(\mu|\xi) = \prod_{v} \mathcal{B}(\mu_v|\xi_v)$$

$$P(\mathbf{R}=\mathbf{r}_n|\mathbf{X}=\mathbf{x}_n,\mu) = \prod_{d=1}^{D}\prod_{v=1}^{V} \mu_v^{[r_{dn}=1][x_{dn}=v]}(1-\mu_v)^{[r_{dn}=0][x_{dn}=v]}$$

DP Mixture/CPT-v: Inference

Auxiliary Variable Gibbs: Mixture Indicator Update

$$P(z_n=k, \exists j\neq n \;\; z_j=k|z_{-n},\mathbf{x}_n^o,\mathbf{r}_n,\mathbf{a}_n,\beta,\alpha,\mu)$$
$$\propto \frac{\sum_{i\neq n}^{N}[z_i=k]}{N-1+\alpha}\prod_{d=1}^{D}\left(\prod_{v=1}^{V}\beta_{vdk}{}^{[x_{dn}=v]}\right)^{[r_{dn}=1][a_{dn}=1]}\left(\sum_{v=1}^{V}\mu_v\beta_{vdk}\right)^{[r_{dn}=1][a_{dn}=0]}$$
$$\cdot\left(\sum_{v=1}^{V}(1-\mu_v)\beta_{vdk}\right)^{[r_{dn}=0][a_{dn}=0]}$$

$$P(\forall j\neq n \;\; z_n\neq z_j|z_{-n},\mathbf{x}_n^o,\mathbf{r}_n,\mathbf{a}_n,\alpha,\mu,\phi_0)$$
$$\propto \frac{\alpha}{N-1+\alpha}\prod_{d=1}^{D}\left(\prod_{v=1}^{V}\frac{\phi_{vd0}}{\sum_v\phi_{vd0}}^{[x_{dn}=v]}\right)^{[r_{dn}=1][a_{dn}=1]}\left(\sum_{v=1}^{V}\mu_v\frac{\phi_{vd0}}{\sum_v\phi_{vd0}}\right)^{[r_{dn}=1][a_{dn}=0]}$$
$$\cdot\left(\sum_{v=1}^{V}(1-\mu_v)\frac{\phi_{vd0}}{\sum_v\phi_{vd0}}\right)^{[r_{dn}=0][a_{dn}=0]}$$

DP Mixture/CPT-v: Inference

Auxiliary Variable Gibbs: Auxiliary Count Update

$$P(C_{dk00}=c_{dk00}|\mathbf{z},\mathbf{x}^o,\mathbf{r},\mathbf{a},\beta,\mu) = \frac{c_{dk00}!}{\prod_{v=1}^{V}c_{vdk00}!}\prod_{v=1}^{V}P(x_d=v|r_d=0,a_d=0,z_n=k)^{c_{vdk00}}$$
$$= \frac{c_{dk00}!}{\prod_{v=1}^{V}c_{vdk00}!}\prod_{v=1}^{V}\left(\frac{(1-\mu_v)\beta_{vmk}}{\sum_{v=1}^{V}(1-\mu_v)\beta_{vmk}}\right)^{c_{vdk00}}$$

$$P(C_{dk10}=c_{dk10}|\mathbf{z},\mathbf{x}^o,\mathbf{r},\mathbf{a},\beta,\mu) = \frac{c_{dk10}!}{\prod_{v=1}^{V}c_{vdk10}!}\prod_{v=1}^{V}P(x_d=v|r_d=1,a_d=0,z_n=k)^{c_{vdk10}}$$
$$= \frac{c_{dk10}!}{\prod_{v=1}^{V}c_{vdk10}!}\prod_{v=1}^{V}\left(\frac{\mu_v\beta_{vmk}}{\sum_{v=1}^{V}\mu_v\beta_{vmk}}\right)^{c_{vdk10}}$$

DP Mixture/CPT-v: Inference

Auxiliary Variable Gibbs: Parameter Updates

$$P(\beta_{vdk}|\mathbf{z},\mathbf{x}^o,\mathbf{x}^m,\mathbf{r},\phi_0) \propto \prod_{n=1}^{N}P(x_{dn}|\beta_{dk})P(\beta_{dk}|\phi_{d0})$$
$$= \mathcal{D}(c_{vdk11}+c_{vdk10}+c_{vdk00}+\phi_{vd0})$$

$$P(\mu_v|\mathbf{x}^o,\mathbf{x}^m,\mathbf{r},\xi) \propto \prod_{n=1}^{N}\prod_{d=1}^{D}\mu_v^{[r_{dn}=1][x_{dn}=v]}(1-\mu_v)^{[r_{dn}=0][x_{dn}=v]}\cdot\mu_v^{\eta_{1v}-1}(1-\mu_v)^{\eta_{0v}-1}$$
$$= \mathcal{B}(\eta_{1v}+\sum_{k=1}^{K}\sum_{d=1}^{D}c_{vdk11}+c_{vdk10}, \eta_{0v}+\sum_{k=1}^{K}\sum_{d=1}^{D}c_{vdk00})$$

DP Mixture/CPT-v: Prediction

Predictive Distribution

$$P(x_{dn}=v|\{\mathbf{x}_n,\mathbf{r}_n,\mathbf{a}_n\}_{n=1:N},\alpha,\phi_0,\eta) \approx \frac{1}{S}\sum_{s=1}^{S}P(x_{dn}=v|r_{dn},z_n^s,\beta^s,\mu^s)$$
$$= \frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{K^s}[z_n^s=k]\left(\frac{\mu_v\beta_{vdk}}{\sum_{v=1}^{V}\mu_v\beta_{vdk}}\right)^{[r_{dn}=0]}\left(\frac{(1-\mu_v)\beta_{vdk}}{\sum_{v=1}^{V}(1-\mu_v)\beta_{vdk}}\right)^{[r_{dn}=0]}$$

DP Mixture/CPT-v: Results

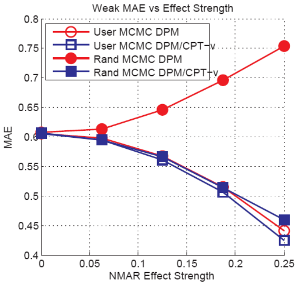
Yahoo! Results: DP vs DP/CPT-v and DP/CPT-v+

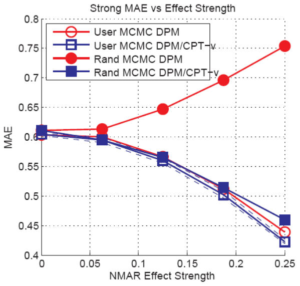
	Weak Rand	Weak User
MCMC DP	0.7658 ± 0.0031	0.5735 ± 0.0004
MCMC DP/CPT-v	0.5548 ± 0.0037	0.6798 ± 0.0049
MCMC DP/CPT-v+	0.4421 ± 0.0008	0.7814 ± 0.0082

	Strong Rand	Strong User
MCMC DP	0.7624 ± 0.0063	0.5767 ± 0.0077
MCMC DP/CPT-v	0.5549 ± 0.0026	0.6670 ± 0.0071
MCMC DP/CPT-v+	0.4428 ± 0.0027	0.7537 ± 0.0026

DP Mixture/CPT-v: Results

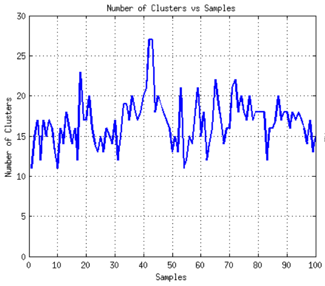
Jester Results: DP vs DP/CPT-v

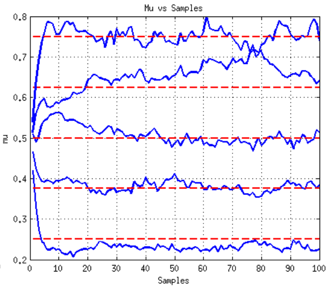




DP Mixture/CPT-v: MCMC Diagnostics

Example Parameter Traces on Jester





Finite Mixture/LOGIT-vd: Model

Probability Model:

$$P(\theta|\alpha) = \mathcal{D}(\theta|\alpha)$$

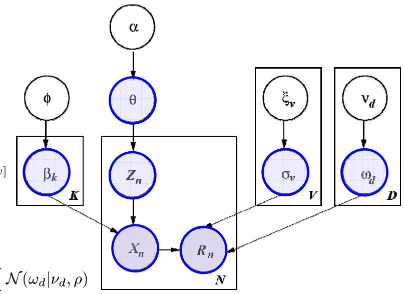
$$P(\beta|\phi) = \prod_{k=1}^{K}\prod_{d=1}^{D}\mathcal{D}(\beta_{dk}|\phi_{dk})$$

$$P(X_d = x_{dn}|Z_n = k, \beta) = \prod_{v=1}^{V}\beta_{vdk}^{[x_{dn}=v]}$$

$$\mu_{vd} = \frac{1}{1+\exp(-(\sigma_v + \omega_d))}$$

$$P(\sigma, \omega|\xi, \tau, \nu, \rho) = \prod_{v=1}^{V}\mathcal{N}(\sigma_v|\xi_v, \tau)\prod_{d=1}^{D}\mathcal{N}(\omega_d|\nu_d, \rho)$$

$$P(\mathbf{R} = \mathbf{r}_n|\mathbf{X} = \mathbf{x}_n, \mu) = \prod_{d=1}^{D}\prod_{v=1}^{V}\mu_{vd}^{[r_{dn}=1][x_{dn}=v]}(1-\mu_{vd})^{[r_{dn}=0][x_{dn}=v]}$$



Finite Mixture/LOGIT-vd: Learning

MAP GEM Algorithm (E-Step):

$$q_n(k) = P(z_n = k|\mathbf{x}_n^o, \mathbf{r}_n, \mathbf{a}_n, \theta, \beta, \mu) = \frac{\theta_k \prod_{d=1}^{D}\gamma_{dkn}}{\sum_{k=1}^{K}\theta_k\prod_{d=1}^{D}\gamma_{dkn}}$$

$$q_n(k,v,d) = P(z_n = k, x_{dn} = v|\mathbf{x}_n^o, \mathbf{r}_n, \mathbf{a}_n, \theta, \beta, \mu)$$

$$= q_n(k)\left(\frac{\mu_{vd}\beta_{vdk}}{\sum_{v'=1}^{V}\mu_{v'd}\beta_{v'dk}}\right)^{[r_{dn}=1][a_{dn}=0]}\left(\frac{(1-\mu_{vd})\beta_{vdk}}{\sum_{v'=1}^{V}(1-\mu_{v'd})\beta_{v'dk}}\right)^{[r_{dn}=0][a_{dn}=0]}$$

$$\gamma_{dkn} = \left(\prod_{v=1}^{V}(\beta_{vdk}\mu_{vd})^{[x_{dn}=v]}\right)^{[r_{dn}=1][a_{dn}=1]}\left(\prod_{v=1}^{V}(\beta_{vdk}(1-\mu_{vd}))^{[x=v]}\right)^{[r_{dn}=0][a_{dn}=1]}$$

$$\cdot\left(\sum_{v=1}^{V}\beta_{vdk}\mu_{vd}\right)^{[r_{dn}=1][a_{dn}=0]}\left(\sum_{v=1}^{V}\beta_{vdk}(1-\mu_{vd})\right)^{[r_{dn}=0][a_{dn}=0]}$$

Finite Mixture/LOGIT-vd: Learning

MAP GEM Algorithm (M-Step):

$$\theta_k = \frac{\alpha_k - 1 + \sum_{n=1}^{N} q_n(k)}{N - K + \sum_{k=1}^{K}\alpha_k}$$

$$\beta_{vdk} = \frac{\phi_{vdk} - 1 + \sum_{n=1}^{N} q_n(k)[a_{dn}=1][x_{dn}=v] + q_n(k,v,d)[a_{dn}=0]}{\sum_{n=1}^{N} q_n(k) - V + \sum_{v=1}^{V}\phi_{vdk}}$$

$$\sigma_v = \sigma_v - \lambda\frac{\partial E[\log\mathcal{P}]}{\partial\sigma_v}$$

$$\omega_d = \omega_d - \lambda\frac{\partial E[\log\mathcal{P}]}{\partial\omega_d}$$

Finite Mixture/LOGIT-vd: Prediction

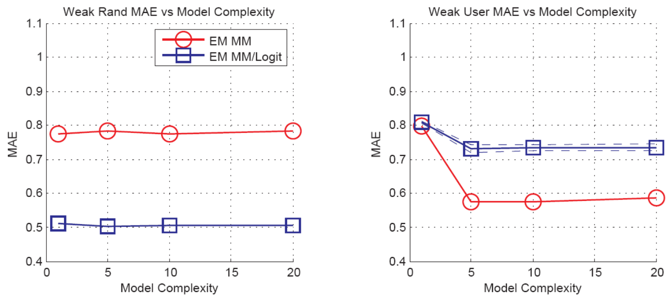
Predictive Distribution

$$P(X_{dn} = v|\mathbf{x}_n, \mathbf{r}_n, \mathbf{a}_n, \theta, \beta, \mu)$$

$$= \sum_{k=1}^{K}\frac{\theta_k\prod_{d=1}^{D}\gamma_{dkn}}{\sum_{k=1}^{K}\theta_k\prod_{d=1}^{D}\gamma_{dkn}}\left(\frac{\mu_{vd}\beta_{vdk}}{\sum_{v'=1}^{V}\mu_{v'd}\beta_{v'dk}}\right)^{[r_{dn}=1]}\left(\frac{(1-\mu_{vd})\beta_{vdk}}{\sum_{v'=1}^{V}(1-\mu_{v'd})\beta_{v'dk}}\right)^{[r_{dn}=0]}$$

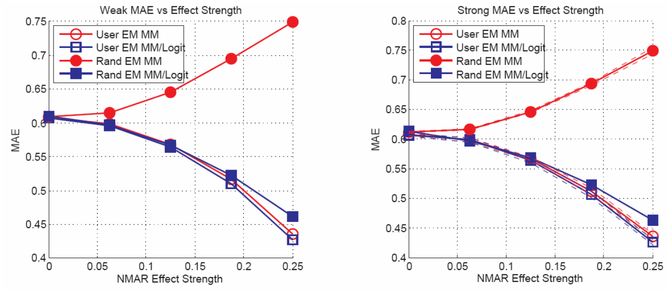
Finite Mixture/LOGIT-vd: Results

Yahoo! Weak Generalization Results: MM vs MM/LOGIT-vd



Finite Mixture/LOGIT-vd: Results

Jester Results: MM vs MM/LOGIT-vd



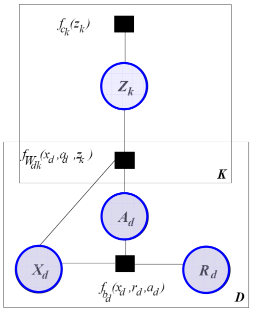
Conditional RBM: Model

$$f_{c_k}(z_k) = \exp(c_k[z_k = 1])$$

$$f_{W_{dk}}(x_{dn}, a_{dn}, z_k) = \exp(\sum_{v=1}^{V} W_{vdk}[x_{dn} = v][a_{dn} = 1][z_k = 1])$$

$$f_{b_d}(x_{dn}, r_{dn}, a_{dn}) = \exp(\sum_{v=1}^{V} b^1_{vd}[x_{dn} = v][r_{dn} = 1][a_{dn} = 1])$$

$$\exp(\sum_{v=1}^{V} b^0_{v}[x_{dn} = v][r_{dn} = 0][a_{dn} = 1])$$



Conditional RBM: Model

Probability Model:

$$P(\mathbf{x}^o_n, \mathbf{z}_n | \mathbf{r}_n, \mathbf{a}_n) = \frac{\exp(-E(\mathbf{x}^o_n, \mathbf{z}_n, \mathbf{r}_n, \mathbf{a}_n))}{\sum_{\mathbf{x}^o} \sum_{\mathbf{z}} \exp(-E(\mathbf{x}^o, \mathbf{z}, \mathbf{r}_n), \mathbf{a}_n)}$$

$$\begin{aligned} E(\mathbf{x}^o_n, \mathbf{z}_n, \mathbf{r}_n, \mathbf{a}_n) = &- \sum_{d=1}^{D} \sum_{v=1}^{V} \sum_{k=1}^{K} W_{vdk}[x_d = v][z_k = 1][a_{dn} = 1] \\ &- \sum_{d=1}^{D} \sum_{v=1}^{V} b^1_{vd}[x_d = v][r_{dn} = 1][a_{dn} = 1] \\ &- \sum_{d=1}^{D} \sum_{v=1}^{V} b^0_{v}[x_d = v][r_{dn} = 0][a_{dn} = 1] \\ &- \sum_{k=1}^{K} c_k[z_k = 1] \end{aligned}$$

Conditional RBM: Inference

Gibbs Sampler:

$$\begin{aligned} &P(x_{dn} = v | r_{dn}, a_{dn}, \mathbf{z}, W, b) \\ &= \frac{\exp(\sum_{k=1}^{K} W_{vdk}[z_k = 1] + b^1_{vd}[r_{dn} = 1] + b^0_v[r_{dn} = 0])}{\sum_{v=1}^{V} \exp(\sum_{k=1}^{K} W_{vdk}[z_k = 1] + b^1_{vd}[r_{dn} = 1] + b^0_v[r_{dn} = 0])} \end{aligned}$$

$$\begin{aligned} &P(z_k = 1 | \mathbf{x}^o_n, \mathbf{r}_n, \mathbf{a}_n, W, c) \\ &= \frac{1}{1 + \exp(-(\sum_{d=1}^{D} \sum_{v=1}^{V} W_{vdk}[x_{dn} = v][a_{dn} = 1] + c_k))} \end{aligned}$$

Conditional RBM: Learning

Contrastive Divergence Gradients:

$$\begin{aligned} \frac{\partial \mathcal{C}(\mathbf{x}|\mathbf{r}, \mathbf{a}, W, b, c)}{\partial W_{vdk}} &\approx \sum_{n=1}^{N} [x_{dn} = v][a_{dn} = 1] P(\mathbf{z}_{kn} = 1 | \mathbf{x}_n) \\ &- \sum_{n=1}^{N} [x^T_{dn} = v][a_{dn} = 1] P(\mathbf{z}_{kn} = 1 | \mathbf{x}^T_n) \end{aligned}$$

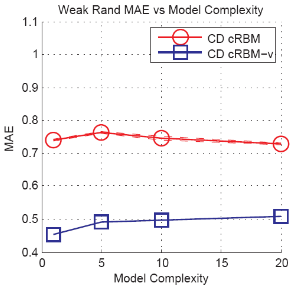
$$\frac{\partial \mathcal{C}(\mathbf{x}|\mathbf{r}, \mathbf{a}, W, b, c)}{\partial c_k} \approx \sum_{n=1}^{N} \left(P(\mathbf{z}_{kn} = 1 | \mathbf{x}_n) - P(\mathbf{z}_{kn} = 1 | \mathbf{x}^T_n) \right)$$

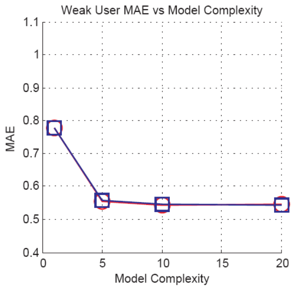
$$\frac{\partial \mathcal{C}(\mathbf{x}|\mathbf{r}, \mathbf{a}, W, b, c)}{\partial b^1_{vd}} \approx \sum_{n=1}^{N} [r_{dn} = 1][a_{dn} = 1] \left([x_{dn} = v] - P(x^T_{dn} = v) \right)$$

$$\frac{\partial \mathcal{C}(\mathbf{x}|\mathbf{r}, \mathbf{a}, W, b, c)}{\partial b^0_{v}} \approx \sum_{n=1}^{N} \sum_{d=1}^{D} [r_{dn} = 0][a_{dn} = 1] \left([x_{dn} = v] - P(x^T_{dn} = v) \right)$$

Conditional RBM: Results

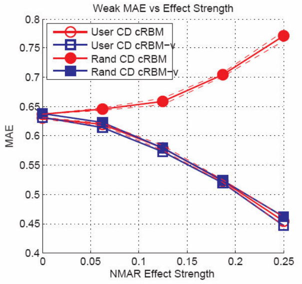
Yahoo! Weak Generalization Results: cRBM vs cRBM-v

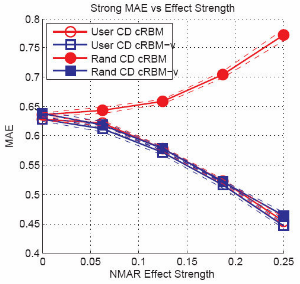




Conditional RBM: Results

Jester Results: cRBM vs cRBM-v





Unsupervised Learning – NMAR: Results

- Methods that model NMAR effects perform significantly better than methods that don't on synthetic and real data.
- Differences between methods that model NMAR effects are small by comparison, but still significant.
- Results show a big win for rating prediction when a small number of ratings for randomly selected items is available at training time.

Unsupervised Learning – NMAR: Comparison of Results on Yahoo! Data

	Complexity	Rand MAE	Complexity	User MAE
EM MM	1	0.7725 ± 0.0024	5	0.5779 ± 0.0066
EM MM/CPT-v	20	0.5431 ± 0.0012	10	0.6661 ± 0.0025
EM MM/Logit	5	0.5038 ± 0.0030	5	0.7029 ± 0.0186
EM MM/CPT-v+	5	0.4456 ± 0.0033	20	0.7088 ± 0.0087
MCMC DP	N/A	0.7624 ± 0.0063	N/A	0.5767 ± 0.0077
MCMC DP/CPT-v	N/A	0.5549 ± 0.0026	N/A	0.6670 ± 0.0071
MCMC DP/CPT-v+	N/A	0.4428 ± 0.0027	N/A	0.7537 ± 0.0026
CD RBM	20	0.7179 ± 0.0025	10	0.5513 ± 0.0077
CD cRBM/E-v	1	0.4553 ± 0.0031	20	0.5506 ± 0.0085

Unsupervised Learning – NMAR: NEW: Ranking Results

$$NDCG(n) = \frac{\sum_{i=1}^{T} \frac{2^{x_{ni}^t} - 1}{\log(1 + \hat{\pi}(i,n))}}{\sum_{i=1}^{T} \frac{2^{x_{ni}^t} - 1}{\log(1 + \pi(i,n))}}$$

- $\hat{x}_{ni}^t$: mean of posterior predictive distribution for test item i.
- $\hat{\pi}(i,n)$: rank of test item i according to $\hat{x}_{ni}^t$.
- $\pi(i,n)$: rank of test item i according to x_{ni}^t .

Unsupervised Learning – NMAR: NEW: Comparison of Yahoo! Ranking Results

Weak Generalization:

	K=1	K=5	K=10	K=20
EM MM	0.8153 ± 0.0007	0.8135 ± 0.0006	0.8106 ± 0.0005	0.8073 ± 0.0006
EM MM/CPT-v	0.8257 ± 0.0006	0.8325 ± 0.0006	0.8353 ± 0.0006	0.8356 ± 0.0008
EM MM/Logit	0.8251 ± 0.0005	0.8385 ± 0.0003	0.8384 ± 0.0005	0.8381 ± 0.0010
EM MM/CPT-v+	0.8282 ± 0.0003	0.8337 ± 0.0007	0.8355 ± 0.0008	0.8367 ± 0.0007
MCMC DP	0.8167 ± 0.0007	0.8167 ± 0.0007	0.8167 ± 0.0007	0.8167 ± 0.0007
MCMC DP/CPT-v	0.8259 ± 0.0010	0.8259 ± 0.0010	0.8259 ± 0.0010	0.8259 ± 0.0010
MCMC DP/CPT-v+	0.8320 ± 0.0011	0.8320 ± 0.0011	0.8320 ± 0.0011	0.8320 ± 0.0011
CD cRBM	0.8104 ± 0.0007	0.8154 ± 0.0012	0.8174 ± 0.0010	0.8183 ± 0.0011
CD cRBM/E-v	0.8211 ± 0.0007	0.8185 ± 0.0010	0.8220 ± 0.0011	0.8210 ± 0.0009

Unsupervised Learning – NMAR:
NEW: Comparison of Yahoo! Ranking Results

Strong Generalization:

	Complexity	Rand NDCG
EM MM	1	0.8162 ± 0.0022
EM MM/CPT-v	20	0.8352 ± 0.0023
EM MM/Logit	5	0.8398 ± 0.0012
EM MM/CPT-v+	20	0.8377 ± 0.0012
MCMC DP	N/A	0.8167 ± 0.0025
MCMC DP/CPT-v	N/A	0.8248 ± 0.0020
MCMC DP/CPT-v+	N/A	0.8319 ± 0.0011
CD cRBM	20	0.8207 ± 0.0011
CD cRBM/E-v	10	0.8244 ± 0.0017

Classification with Missing Data:

Background on Classification with Complete Data:

- Linear/Regularized Discriminant Analysis
- Logistic Regression
- Perceptrons and SVMs
- Kernel Methods and Kernel Logistic Regression
- Multi-Layer Neural Networks

Frameworks for Classification with Missing Features:

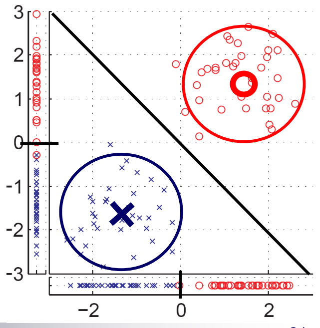
- Generative Classifiers
- Single and Multiple Imputation
- Reduced Models/Classification in Subspaces
- Response Indicator Augmentation

Generative Framework
Linear Discriminant Analysis

$$P(\mathbf{X}_n^o = \mathbf{x}_n^o | Y_n = c) = \mathcal{N}(\mathbf{x}_n^o | \mu_c^o, \Sigma^{oo})$$

$$P(Y = c | \mathbf{X}_n^o = \mathbf{x}_n^o) = \frac{\theta_c \mathcal{N}(\mathbf{x}_n^o | \mu_c^o, \Sigma^{oo})}{\sum_c \theta_c \mathcal{N}(\mathbf{x}_n^o | \mu_c^o, \Sigma^{oo})}$$

Factor Analysis Covariance Model: $\Sigma = \Lambda\Lambda^T + \Psi$



Generative Framework
Linear Discriminant Analysis

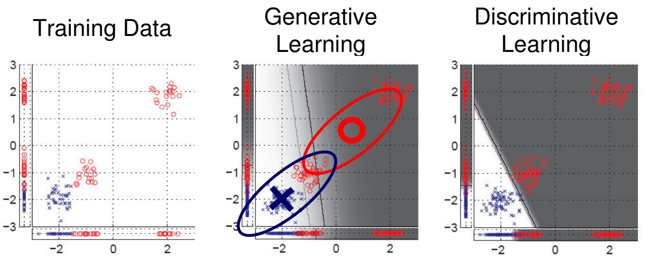
Generative Training:

- Estimate class means from incomplete data
- Run EM for Factor analysis with missing data to estimate pooled covariance parameters

Discriminative Training:

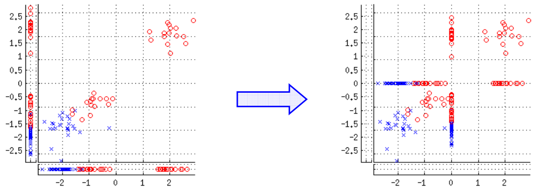
- Directly maximize the conditional likelihood of the labels given incomplete features.
- Non-linear gradient descent in the negative log conditional likelihood.

Generative Framework
Linear Discriminant Analysis



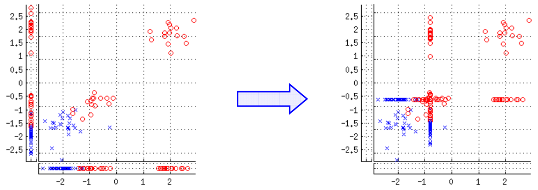
Imputation Framework

Zero Imputation: Replace missing feature values with zeros.



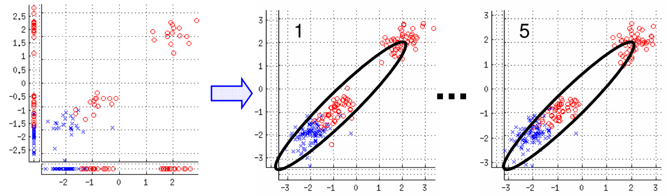
Imputation Framework

Mean Imputation: Replace missing feature values with mean of observed values for each feature.



Imputation Framework

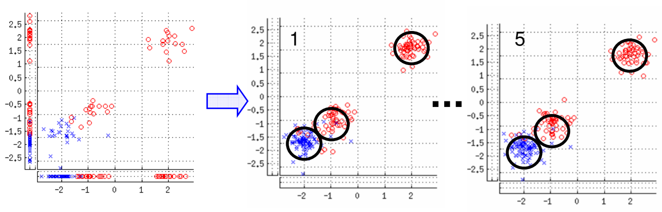
Multiple Imputation: Replace missing feature values with samples of $\mathbf{x}^m$ given $\mathbf{x}^o$ drawn from several imputation models.



Mixture of Factor Analyzers
K=1, Q=1

Imputation Framework

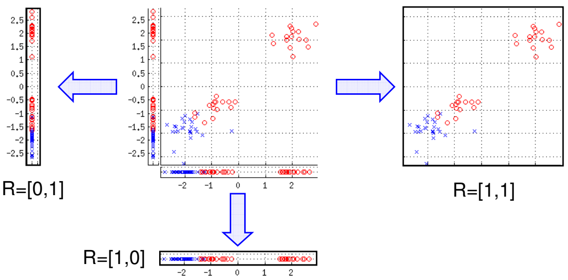
Multiple Imputation: Replace missing feature values with samples of $\mathbf{x}^m$ given $\mathbf{x}^o$ drawn from several imputation models.



Mixture of Factor Analyzers
K=3, Q=1

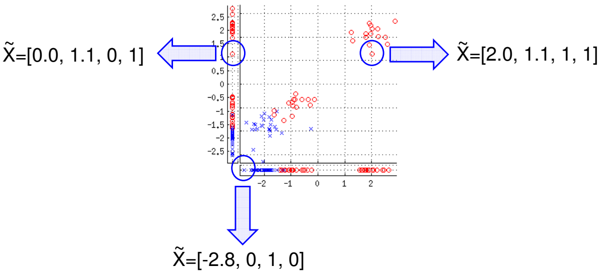
Reduced Models Framework

Reduced Models: Each observed data subspace defined by a pattern of missing data gives a separate classification problem.



Response Augmentation Framework

Response Augmentation: Set missing features to zero and augment feature representation with response indicators.



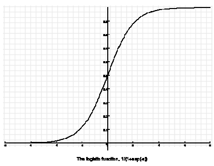
Logistic Regression: Model

Linear logistic regression optimizes the conditional likelihood of the class labels given the features using gradient methods.

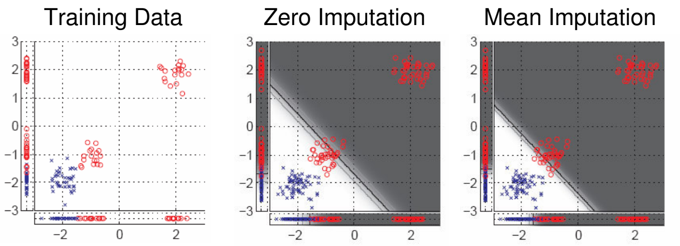
- Can exactly represent the class posterior of exponential family class conditional models with shared dispersion.

$$P(Y=1|\mathbf{X}=\mathbf{x}) = \frac{1}{1+\exp(-(\mathbf{w}^T\mathbf{x}+b))}$$

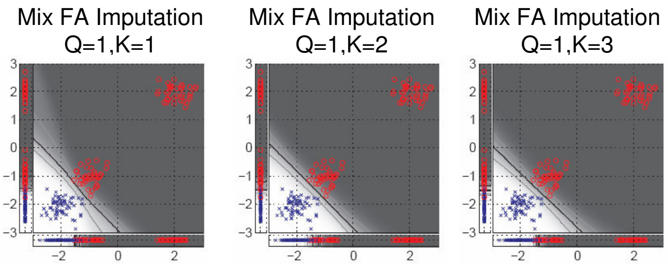
$$P(Y=c|\mathbf{X}=\mathbf{x}) = \frac{\exp(\mathbf{w}_c^T\mathbf{x}+b_c)}{\sum_{c'=1}^{C}\exp(\mathbf{w}_{c'}^T\mathbf{x}+b_{c'})}$$



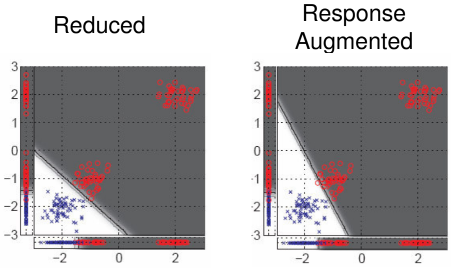
Logistic Regression: Synthetic Data



Logistic Regression: Synthetic Data



Logistic Regression: Synthetic Data



Kernel Logistic Regression: Model

Kernel logistic regression optimizes the conditional likelihood of the class labels given training data and a kernel function.

$$P(Y=c|\mathbf{X}=\mathbf{x}) = \frac{\exp(\sum_{n'=1}^{N} \alpha_{n'c}\mathcal{K}(\mathbf{x}_{n'},\mathbf{x})+b_c)}{\exp(\sum_{c'=1}^{C}\sum_{n'=1}^{N} \alpha_{n'c'}\mathcal{K}(\mathbf{x}_{n'},\mathbf{x})+b_{c'})}$$

$$F(\alpha) = \min_{\alpha} \sum_{n=1}^{N}\sum_{c=1}^{C} -[y_n=c]\log\left(\frac{\exp(\sum_{n'=1}^{N} \alpha_{n'c}\mathcal{K}(\mathbf{x}_{n'},\mathbf{x}_n)+b_c)}{\exp(\sum_{c'=1}^{C}\sum_{n'=1}^{N} \alpha_{n'c'}\mathcal{K}(\mathbf{x}_{n'},\mathbf{x}_n)+b_{c'})}\right) + \gamma\sum_{c=1}^{C}\alpha_c^T\mathbf{K}\alpha_c$$

Kernel Logistic Regression: Basic Kernels for Missing Data

Linear: $K_l^o(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j) = \sum_d r_{di}r_{dj}x_{di}x_{dj}$

Polynomial: $K_p^o(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j) = \left(\kappa + \sum_d r_{di}r_{dj}x_{di}x_{dj}\right)^{\delta}$

Gaussian: $K_g^o(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j) = \exp(-\frac{1}{\sigma^2}D^o(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j)^2)$

$$D^o(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j) = \left(\sum_d r_{di}r_{dj}(x_{di}-x_{dj})^2\right)^{\frac{1}{2}}$$

Kernel Logistic Regression: Response Augmented Kernels for Missing Data

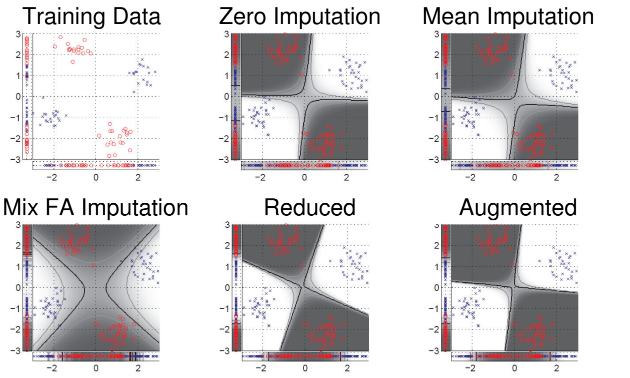
Linear: $K_l^{o+r}(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j) = \sum_d r_{di}r_{dj}x_{di}x_{dj} + \gamma r_{di}r_{dj}$

Polynomial: $K_p^{o+r}(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j) = \left(\kappa + \sum_d r_{di}r_{dj}x_{di}x_{dj} + \gamma r_{di}r_{dj}\right)^{\delta}$

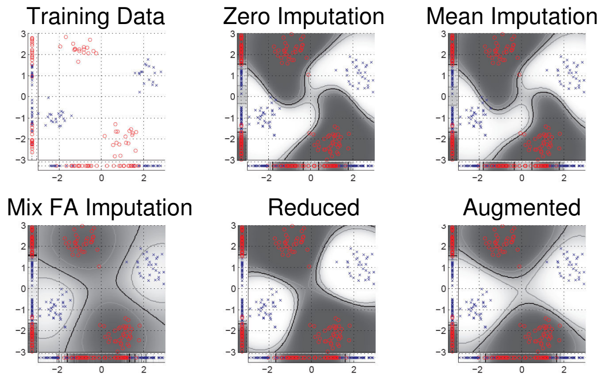
Gaussian: $K_g^{o+r}(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j) = \exp(-\frac{1}{\sigma^2}D^{o+r}(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j)^2)$

$$D^{o+r}(\mathbf{x}_i,\mathbf{r}_i,\mathbf{x}_j,\mathbf{r}_j) = \left(\sum_d r_{di}r_{dj}(x_{di}-x_{dj})^2 + \gamma(r_{di}-r_{dj})^2\right)^{\frac{1}{2}}$$

Polynomial Kernel Logistic Regression



Gaussian Kernel Logistic Regression

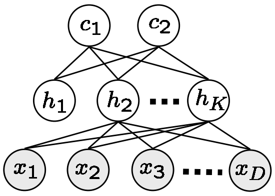


Neural Networks: Model

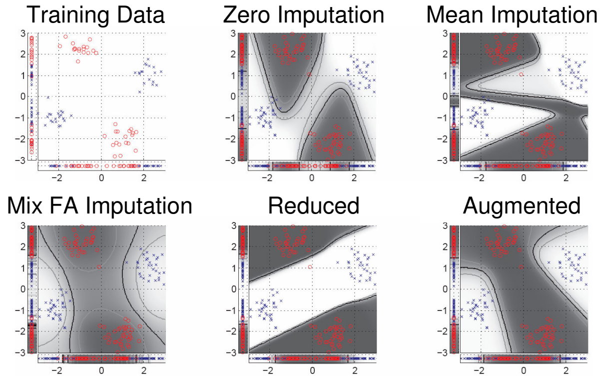
Multi-Layer Sigmoid neural network with cross entropy loss optimizes the conditional likelihood of the class labels given the features using backpropagation.

$$h_k = \frac{1}{1+\exp(-(\mathbf{w}_k^1\mathbf{x}+b_k^1))}$$

$$P(Y=c|\mathbf{X}=\mathbf{x}) = \frac{\exp(\mathbf{w}_c^2\mathbf{h}+b_c^2)}{\sum_{c'=1}^{C}\exp(\mathbf{w}_c^2\mathbf{h}+b_{c'}^2)}$$



Neural Networks:



Classification with Missing Data:
UCI Hepatitis

	Hepatitis	
	Loss	Err(%)
LR Zero	0.4012 ± 0.0439	20.67 ± 2.71
LR Mean	0.4064 ± 0.0576	18.00 ± 2.82
LR MixFA	0.3517 ± 0.0506	13.33 ± 3.44
LR Reduced	0.4443 ± 0.0720	19.33 ± 3.78
LR Augmented	0.5812 ± 0.1258	19.33 ± 4.27
LDA-FA Dis	0.4312 ± 0.0720	20.00 ± 3.98

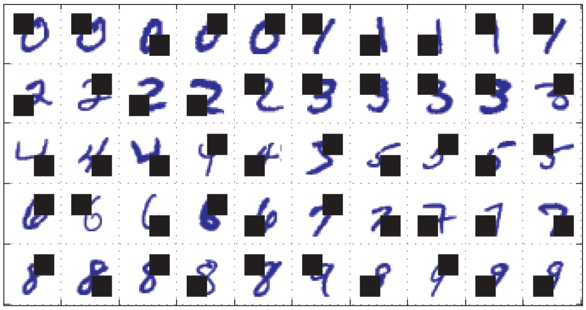
Classification with Missing Data:
UCI Thyroid-AllHypo

	Thyroid: AllHypo	
	Loss	Err(%)
LR Zero	0.1284 ± 0.0002	3.62 ± 0.02
LR Mean	0.1274 ± 0.0001	3.43 ± 0.00
LR MixFA	0.1273 ± 0.0020	3.88 ± 0.15
LR Reduced	0.1281 ± 0.0008	3.53 ± 0.06
LR Augmented	0.1246 ± 0.0003	3.49 ± 0.03
NN Mean	0.0630 ± 0.0007	2.51 ± 0.08
NN MixFA	0.0673 ± 0.0002	2.72 ± 0.03
NN Reduced	0.0650 ± 0.0004	2.55 ± 0.07
NN Augmented	0.0612 ± 0.0003	2.57 ± 0.10
LDA-FA Dis	0.1246 ± 0.0003	3.55 ± 0.02

Classification with Missing Data:
UCI Thyroid-Sick

	Thyroid: Sick	
	Loss	Err(%)
LR Zero	0.2123 ± 0.0005	6.75 ± 0.00
LR Mean	0.1112 ± 0.0000	5.25 ± 0.00
LR MixFA	0.1270 ± 0.0009	6.21 ± 0.11
LR Reduced	0.1263 ± 0.0000	5.35 ± 0.00
LR Augmented	0.1166 ± 0.0024	5.35 ± 0.06
NN Mean	0.1892 ± 0.0036	6.42 ± 0.00
NN MixFA	0.1118 ± 0.0012	5.03 ± 0.15
NN Reduced	0.1069 ± 0.0022	3.81 ± 0.09
NN Augmented	0.1065 ± 0.0025	4.95 ± 0.19
LDA-FA Dis	0.1092 ± 0.0011	5.16 ± 0.02

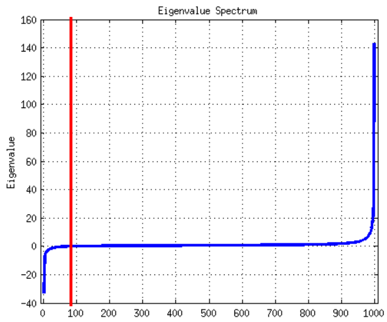
Classification with Missing Data:
MNIST Digit Classification with Missing Data



Classification:
MNIST Digit Classification with Missing Data

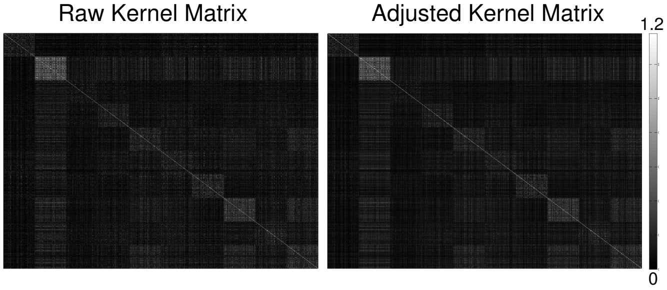
	MNIST Digits	
	Loss	Err(%)
LR Zero	0.6350 ± 0.0110	19.75 ± 0.41
LR Mean	0.6150 ± 0.0112	19.15 ± 0.34
LR Reduced	0.7182 ± 0.0135	22.62 ± 0.45
LR Augmented	0.6160 ± 0.0112	19.35 ± 0.36
LDA-FA Dis	0.6355 ± 0.0051	19.95 ± 0.25
NN Mean	0.6235 ± 0.0541	18.34 ± 0.42
NN Reduced	0.6944 ± 0.0088	21.51 ± 0.27
NN Augmented	0.5925 ± 0.0161	17.76 ± 0.18
gKLR Mean	0.4147 ± 0.0075	13.02 ± 0.24
gKLR Reduced	0.5694 ± 0.0079	18.32 ± 0.49
gKLR Augmented	0.3896 ± 0.0101	12.34 ± 0.46

Classification:
gKLR Augmented Kernel Details



Classification:
gKLR Augmented Kernel Details

Raw Kernel Matrix   Adjusted Kernel Matrix



The End