Latent Feature Learning

Jun Zhu
dcszj@mail.tsinghua.edu.cn
http://bigml.cs.tsinghua.edu.cn/~jun
State Key Lab of Intelligent Technology & Systems
Tsinghua University

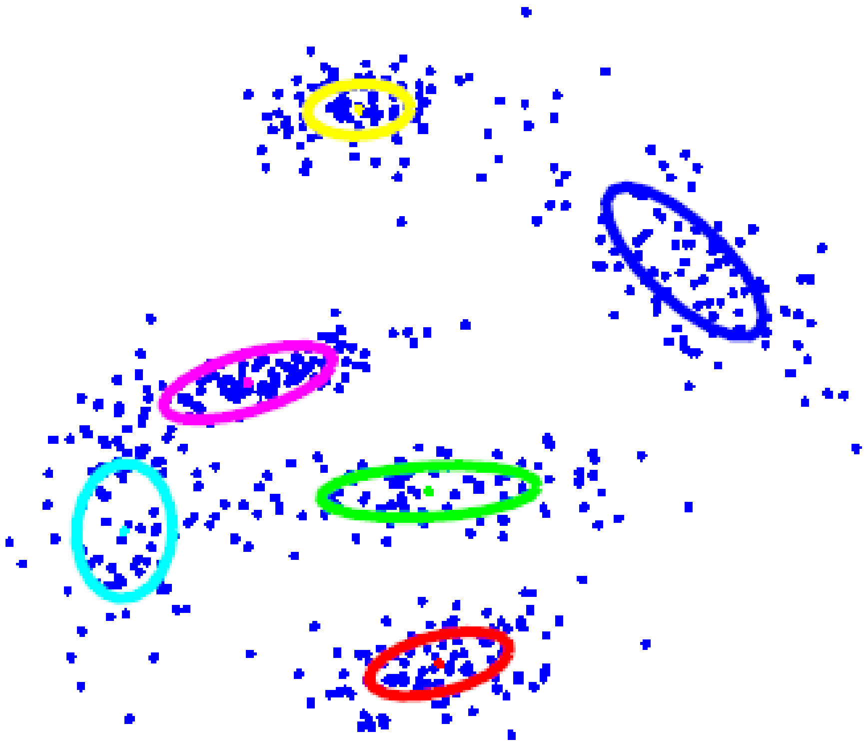
May 19, 2015

Outline

- Finite latent feature models
  - PCA as the particular example
- Infinite latent feature models
  - Indian buffet process

Clustering

- **Basic idea**: each data point belongs to a cluster

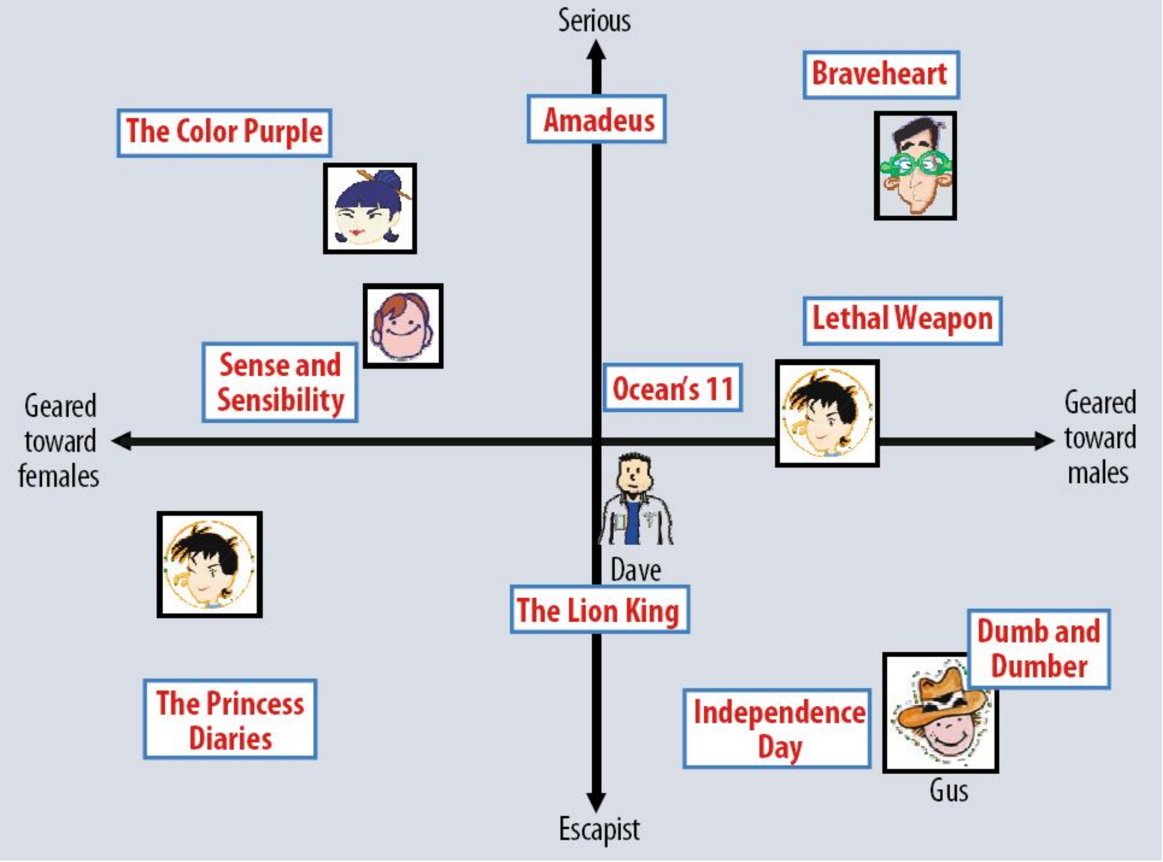


Why latent features?

- Many statistical models can be thought of as modeling data in terms of hidden or latent variables.
- Clustering algorithms (e.g. using mixture models) represent data in terms of which cluster each data point belongs to.
- But clustering models are restrictive…
- Consider modeling people's movie preferences (the "Netflix" problem). A movie might be described using features such as "is science fiction", "has Charlton Heston", "was made in the US", "was made in 1970s", "has apes in it"… these features may be unobserved (latent).
- The number of potential latent features for describing a movie (or person, news story, image, gene, speech waveform, etc) is unlimited.

Example: Latent Feature/Factors

- Characterize both items & users on say 20 to 100 factors inferred from the rating patterns



[Y. Koren, R. Bell & C. Volinsky, IEEE, 2009]

Latent Feature Models are not New …

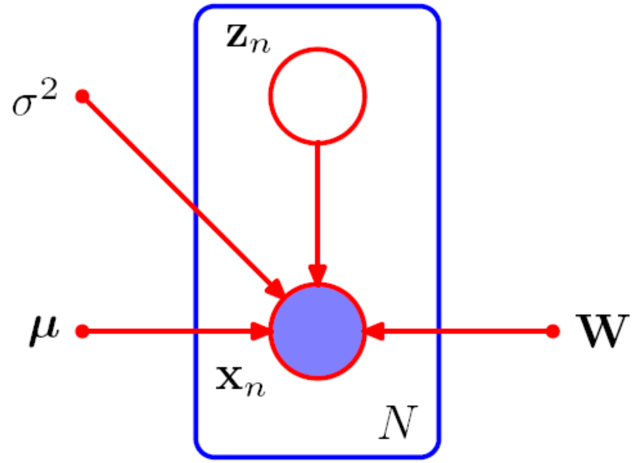
- PCA
- ICA
- LDA (latent discriminant analysis)
- LSI
- Neural networks

- Topic models
  - A special case with some constraints (e.g., conservation of belief constraint)

Probabilistic PCA

- A simple linear-Gaussian model
- Let z be a latent feature vector $\mathbf{z} \in \mathbb{R}^M$
  - In Bayesian, we assume it's prior $\mathbf{z} \sim \mathcal{N}(0, I)$
- A linear-Gaussian model

$$\mathbf{x} = W\mathbf{z} + \mu + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I)$$



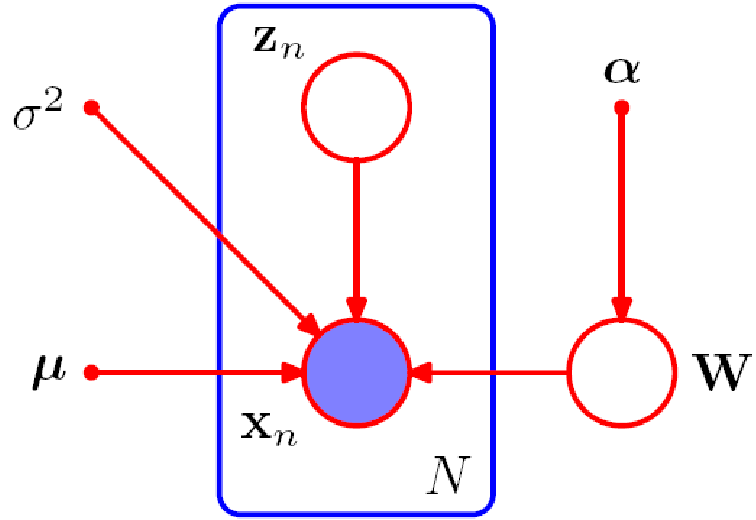
  - this gives the likelihood

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|W\mathbf{z} + \mu, \sigma^2 I)$$

  - the columns of W span a linear subspace

Bayesian PCA

- A prior is assumed on the parameters W



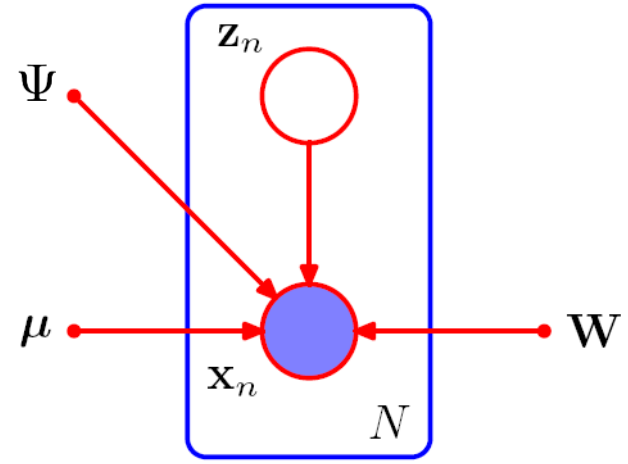
$$p(W|\alpha) = \prod_{i=1}^{M} \left(\frac{\alpha_i}{2}\right)^{D/2} \exp\left\{ -\frac{1}{2}\alpha_i \mathbf{w}_i^\top \mathbf{w}_i \right\}$$

- Inference can be done in closed-form, as in GP regression
- Fully Bayesian treatment put priors on $\mu, \sigma^2,\ \alpha$

Factor Analysis

- Another simple linear-Gaussian model
- Let z be a latent feature vector $\mathbf{z} \in \mathbb{R}^M$
  - In Bayesian, we assume it's prior $\mathbf{z} \sim \mathcal{N}(0, I)$
- A linear-Gaussian model

$$\mathbf{x} = W\mathbf{z} + \mu + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \Psi)$$



  - Ψ is a diagonal matrix
  - this gives the likelihood

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|W\mathbf{z} + \mu, \Psi)$$

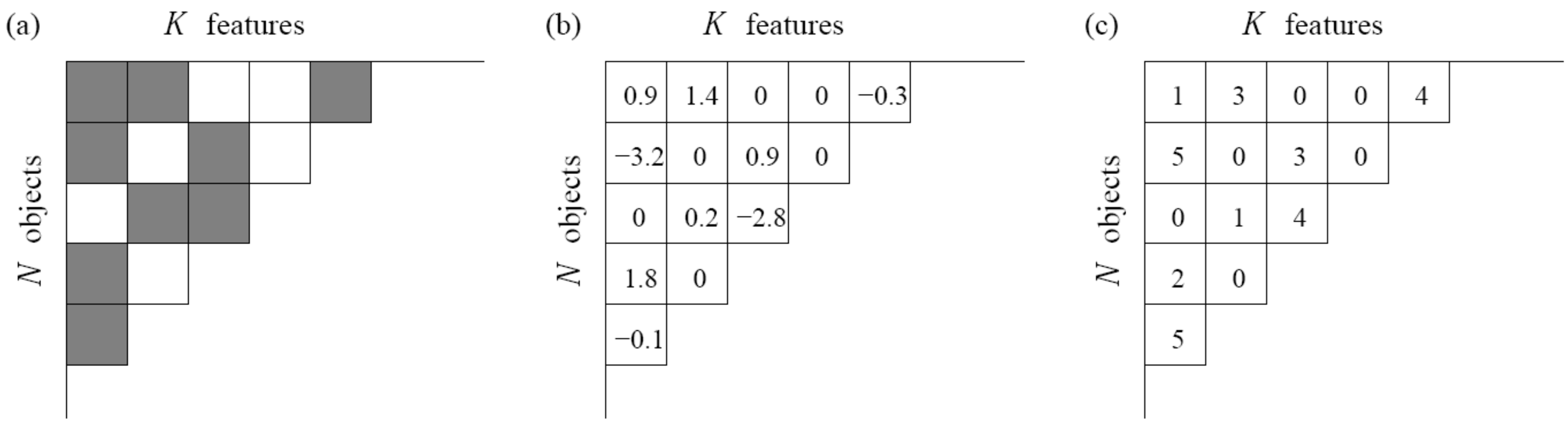
  - the columns of W span a linear subspace

Model Selection Issue

- How to decide the latent dimension?
- We will present a non-parametric technique to automatically infer the latent dimension

Latent Feature Models

- Consider N objects, the latent features form a matrix
- The feature matrix can be decomposed into two components
  - A binary matrix Z indicating which features possessed by each object
  - A matrix V indicating the value of each feature for each object



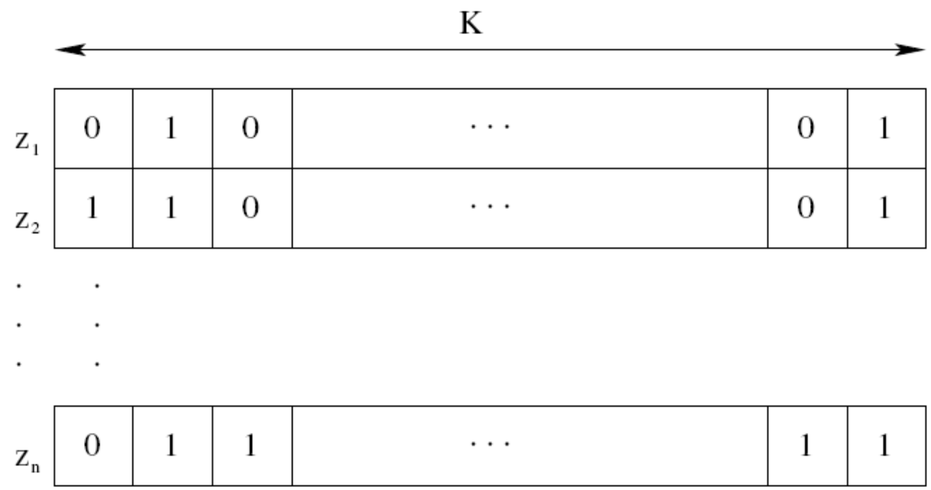
  - Sparsity is imposed on the binary matrix Z
  - For Bayesian, the prior can be imposed as $p(F) = p(Z)p(V)$
  - We will focus on $p(Z)$, which determines the effective dimensionality of latent features

A Finite Latent Feature Model

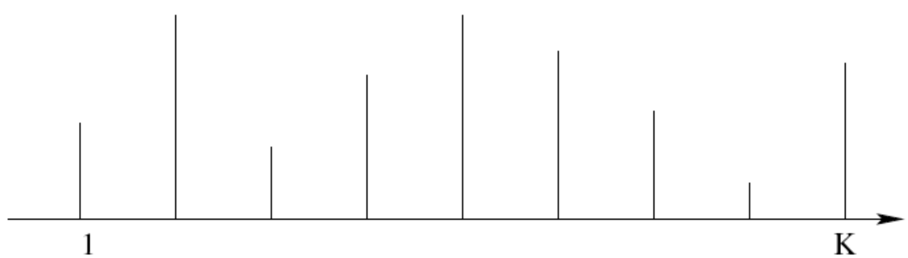
- A random finite binary latent feature model

$$\pi_k | \alpha \sim \mathrm{Beta}(\frac{\alpha}{K}, 1)$$

$$z_{ik} | \pi_k \sim \mathrm{Bernoulli}(\pi_k)$$



  - π_k is the relative probability of each feature being on, e.g.,



A Finite Latent Feature Model

- The marginal probability of a binary matrix Z is

$$p(Z) = \prod_{k=1}^{K} \int \Big(\prod_{i=1}^{N} p(z_{ik}|\pi_k) \Big) p(\pi_k) d\pi_k$$

$$= \prod_{k=1}^{K} \int \Big(\prod_{i=1}^{N} \pi_k^{z_{ik}} (1-\pi_k)^{1-z_{ik}} \Big) p(\pi_k) d\pi_k$$

$$m_k = \sum_{i=1}^{N} z_{ik}$$

$$= \prod_{k=1}^{K} \int \pi_k^{m_k} (1-\pi_k)^{N-m_k} p(\pi_k) d\pi_k$$

$$= \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}$$

Features are independent!

$$\boxed{\int_0^1 \pi^{r-1}(1-\pi)^{s-1} d\pi = \frac{\Gamma(r)\Gamma(s)}{\Gamma(r+s)}}$$

A Finite Latent Feature Model

- The conditional probability of each feature assignment

$$p(z_{ik}=1|Z_{-(i,k)}) = p(z_{ik}=1|\mathbf{z}_{-(i,k)}) = \frac{p(z_{ik}=1,\mathbf{z}_{-(i,k)})}{p(\mathbf{z}_{-(i,k)})}$$

$$p(z_{ik}=1,\mathbf{z}_{-(i,k)}) = \frac{\Gamma\Big((m_{-(i,k)}+1)+\frac{\alpha}{K}\Big)\Gamma\Big(N-(m_{-(i,k)}+1)+1\Big)}{\Gamma(N+\frac{\alpha}{K}+1)}$$

$$p(\mathbf{z}_{-(i,k)}) = \frac{\Gamma(m_{-(i,k)}+\frac{\alpha}{K})\Gamma\Big((N-1)-m_{-(i,k)+1}\Big)}{\Gamma\Big((N-1)+\frac{\alpha}{K}+1\Big)}$$

$$\boxed{p(z_{ik}=1|Z_{-(i,k)}) = \frac{m_{-(i,k)}+\frac{\alpha}{K}}{N+\frac{\alpha}{N}}}$$

$$p(\mathbf{z}_k) = \frac{\frac{\alpha}{K}\Gamma(m_k+\frac{\alpha}{K})\Gamma(N-m_k+1)}{\Gamma(N+1+\frac{\alpha}{K})}$$

$$m_k = \sum_{i=1}^{N} z_{ik} \qquad \Gamma(x+1) = x\Gamma(x)$$

A Finite Latent Feature Model

- Expectation of the number of non-zero features

$$\mathbb{E}[1^\top Z 1] = \mathbb{E}\Big[\sum_{ik} z_{ik}\Big] = K\mathbb{E}[1^\top \mathbf{z}_k]$$

  - the last equality is due to the independence of the features

- For feature k, we have

$$\mathbb{E}[1^\top \mathbf{z}_k] = \sum_{i=1}^{N} \mathbb{E}[z_{ik}] = \sum_{i=1}^{N} \int_0^1 \pi_k p(\pi_k) d\pi_k = N \frac{\frac{\alpha}{K}}{1 + \frac{\alpha}{K}}$$

  - The last equality is due to the fact that expectation of Beta(r, s) is $\dfrac{r}{r+s}$

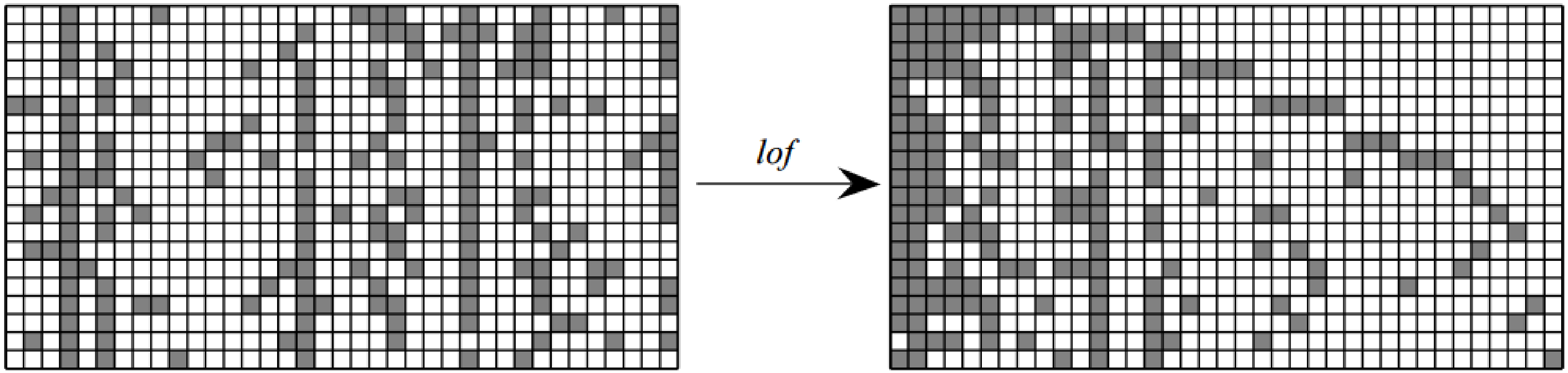
  - Thus, $$\mathbb{E}[1^\top Z 1] = \frac{N\alpha}{1 + \frac{\alpha}{K}} < N\alpha$$

From Finite to Infinite

- A technical difficulty: the probability for any particular matrix goes to zero as $K \to \infty$

$$\lim_{K \to \infty} p(Z|\alpha) = 0$$

- However, if we consider equivalence classes of matrices in left-ordered form obtained by reordering the columns:



A **many to one** mapping! Order the columns from left to right by the magnitude of the binary numbers expressed by that column, taking the first row as the most significant bit.

From Finite to Infinite

- A technical difficulty: the probability for any particular matrix goes to zero as $K \to \infty$

$$\lim_{K \to \infty} p(Z|\alpha) = 0$$

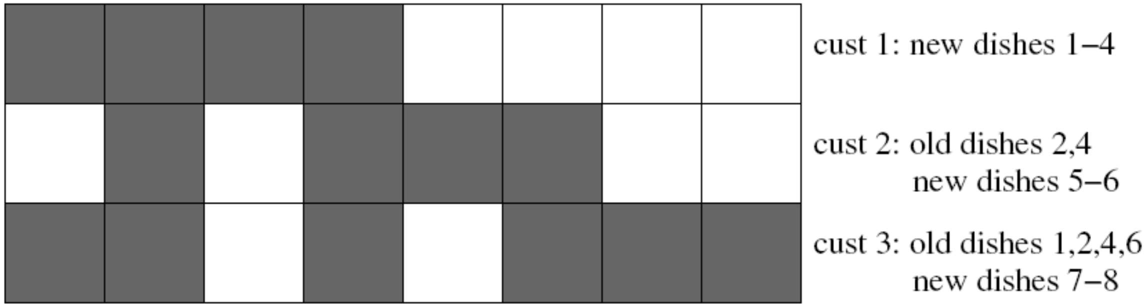
- However, if we consider equivalence classes of matrices in left-ordered form obtained by reordering the columns:

$$\lim_{K \to \infty} p([Z]|\alpha) = \exp\{-\alpha H_N\} \frac{\alpha^{K_+}}{\prod_{h>0} K_h!} \prod_{k \leq K_+} \frac{(N - m_k)!(m_k - 1)!}{N!}$$

  - K_+ is the number of features assigned (i.e. non-zero columns).
  - $H_N = \sum_{n=1}^{N} \frac{1}{n}$ is the Nth harmonic number.
  - K_h are the number of features with history h (a technicality).

Indian Buffet Process

- A stochastic process on infinite binary feature matrices
- Generative procedure:
  - Customer 1 chooses the first K_1 dishes: $K_1 \sim \text{Poisson}(\alpha)$
  - Customer i chooses:
    - Each of the existing dishes with probability $\dfrac{m_k}{i}$
    - K_i additional dishes, where $K_i \sim \text{Poisson}(\dfrac{\alpha}{i})$



$$Z_{i\cdot} \sim \mathcal{IBP}(\alpha)$$

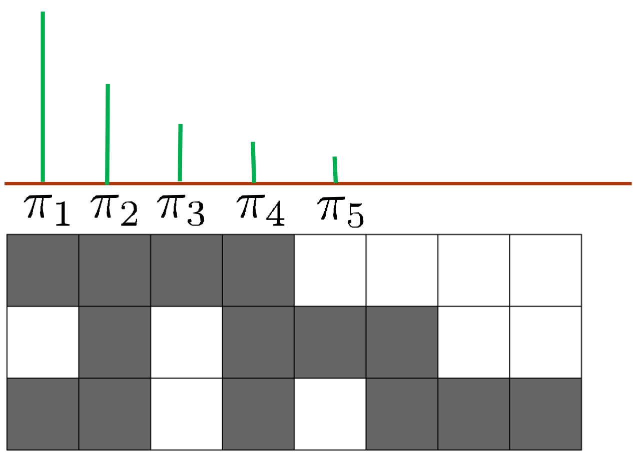
Indian Buffet Process

- A stochastic process on infinite binary feature matrices
- Stick-breaking construction: $Z_i \sim \mathcal{IBP}(\alpha)$

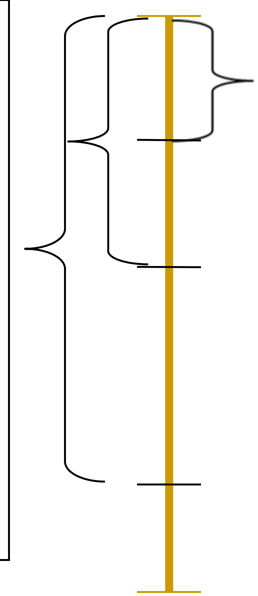
$$z_{nk} \sim \text{Bernoulli}(\pi_k)$$

$$\pi_i(\mathbf{v}) = v_i \pi_{i-1}(\mathbf{v}) = \prod_{j=1}^{i} v_j$$

$$v_i \sim \text{Beta}(\alpha, 1)$$



$\prod_{j=1}^{i-1} v_j$	v_i	π_i
0	0.8	0.8
0.8	0.5	0.4
0.4	0.4	0.16



Inference by Gibbs Sampling

- In the finite Beta-Bernoulli model, we have

$$\boxed{p(z_{ik}=1|Z_{-(i,k)}) = \frac{m_{-(i,k)} + \frac{\alpha}{K}}{N + \frac{\alpha}{N}}}$$

- Set limit $K \to \infty$, we have the conditional for infinite model

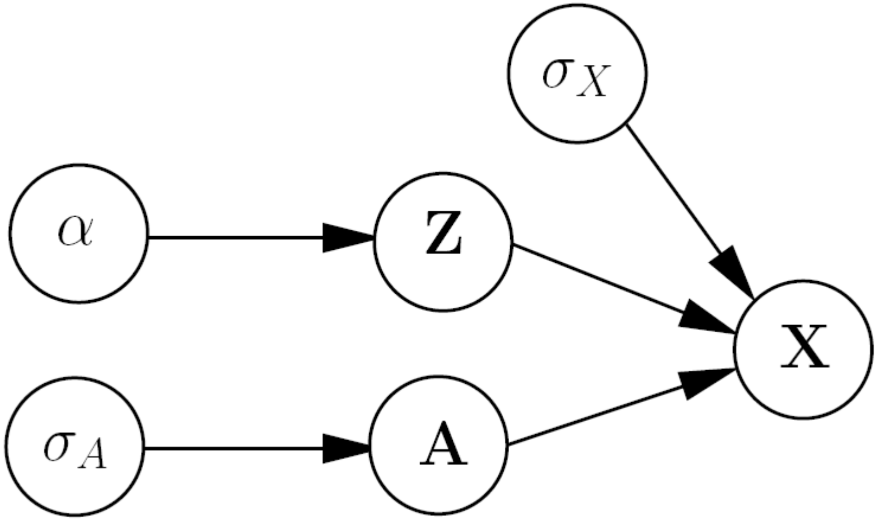
$$p(z_{ik}=1|Z_{-(i,k)}) = \frac{m_{-(i,k)}}{N}$$

  - for any k such that $m_{-(i,k)} > 0$
  - The number of new features should be drawn from

$$\mathrm{Poisson}\left(\frac{\alpha}{K}\right)$$

Use with Data

- A linear-Gaussian model with binary features



  - Gaussian likelihood $p(X|Z, A, \sigma_X) = \mathcal{N}\left(ZA, \sigma_X^2 I\right)$
  - Gaussian prior $p(A|\sigma_A) = \mathcal{N}\left(0, \sigma_A^2 I\right)$

Inference with Gibbs Sampling

- The posterior is

$$p(Z, A|X, \alpha) \propto p(X|Z)p(Z|\alpha)$$

- The conditional for each feature assignment

$$p(z_{nk} = 1|Z_{-(n,k)}, X, \alpha) \propto p(z_{nk} = 1|Z_{-(n,k)}, \alpha)p(X|Z)$$

  - If $m_{-(i,k)} > 0$, $p(z_{ik} = 1|Z_{-(i,k)}) = \frac{m_{-(i,k)}}{N}$
  - For infinitely many k such that $m_{-(i,k)} = 0$: Metropolis steps with truncation to sample from the number of new features for each object

- For linear-Gaussian model, $p(X|Z)$ can be computed

Other Issues

- Sampling methods for non-conjugate models
- Variational inference with the stick-breaking representation of IBP

$$z_{nk} \sim \text{Bernoulli}(\pi_k)$$

$$\pi_i(\mathbf{v}) = v_i \pi_{i-1}(\mathbf{v}) = \prod_{j=1}^{i} v_j$$

$$v_i \sim \text{Beta}(\alpha, 1)$$

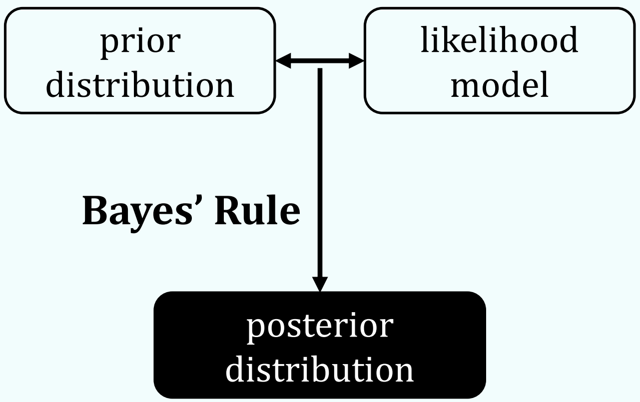
- Applications to various types of data
  - Graph structures, overlapping clusters, time series models

Applications of IBP

Discriminative Bayesian Learning

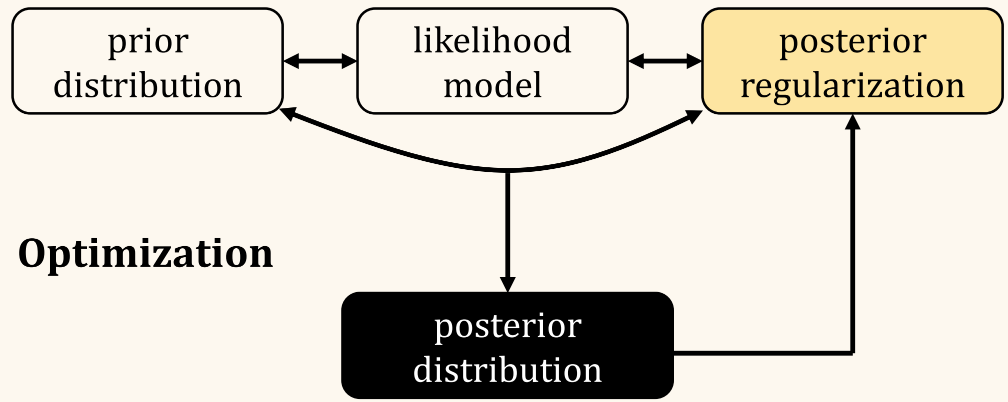
- **Regularized Bayesian inference:**

A paradigm to perform Bayesian inference with rich posterior regularization:



$$p(\mathcal{M}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{M})\pi(\mathcal{M})}{\int p(\mathbf{x}|\mathcal{M})\pi(\mathcal{M})d\mathcal{M}}$$

Thomas Bayes (1763)



$$\min_{q(\mathcal{M})} \quad \mathrm{KL}(q(\mathcal{M})\|p(\mathcal{M}|\mathbf{x})) + \Omega(q(\mathcal{M}))$$
$$\mathrm{s.t.}: \quad q(\mathcal{M}) \in \mathcal{P}_{\mathrm{prob}},$$

posterior regularization

[Zhu, Chen, & Xing, NIPS 2011; JMLR, 2014]

Ways to Derive Posterior Regularization

- **From learning objectives**
  - Performance of posterior distribution can be evaluated when applying it to a learning task
  - Learning objective can be formulated as Pos. Reg.

- **From domain knowledge** (ongoing & future work)
  - Elicit expert knowledge
  - E.g., logic rules

- **Others …** (ongoing & future work)
  - E.g., decision making, cognitive constraints, etc.

PAC-Bayesian Theory

- Basic Setup:
  - Binary classification: $\mathbf{x} \in \mathbb{R}^d \quad y \in \mathcal{Y} = \{-1, +1\}$
  - Unknown, true data distribution: $(\mathbf{x}, y) \sim D$
  - Hypothesis space: $\mathcal{H}$
  - Risk, & Empirical Risk:

$$R(h) = \mathbb{E}_{(\mathbf{x},y)\sim D} I(h(\mathbf{x}) \neq y) \qquad R_S(h) = \frac{1}{N}\sum_{n=1}^{N} I(h(\mathbf{x}_i) \neq y_i)$$

- Learn a posterior distribution Q
- Bayes/majority-vote classifier:

$$B_Q(\mathbf{x}) = \mathrm{sgn}\left[\mathbb{E}_{h\sim Q} h(\mathbf{x})\right]$$

- Gibbs classifier
  - sample an $h \sim Q$, perform prediction

$$R(G_Q) = \mathbb{E}_{h\sim Q} R(h) \qquad R_S(G_Q) = \mathbb{E}_{h\sim Q} R_S(h)$$

PAC-Bayes Theory

- Theorem (Germain et al., 2009):
  - for any distribution D ; for any set $\mathcal{H}$ of classifiers, for any prior P , for any convex function

$$\phi : \; [0,1] \times [0,1] \to \mathbb{R}$$

  - for any posterior Q , for any $\delta \in (0,1]$, the following inequality holds with a high probability ($\geq 1-\delta$)

$$\phi\left(R_S(G_Q), R(G_Q)\right) \leq \frac{1}{N}\left[\mathrm{KL}(Q\|P) + \ln\left(\frac{C(N)}{\delta}\right)\right]$$

  - where $C(N) = \mathbb{E}_{S\sim D^N}\mathbb{E}_{h\sim P}\left[e^{N\phi(R_S(h),R(h))}\right]$

RegBayes Classifiers

- PAC-Bayes theory

$$\phi\left(R_S(G_Q), R(G_Q)\right) \leq \frac{1}{N}\left[\mathrm{KL}(Q\|P) + \ln\left(\frac{C(N)}{\delta}\right)\right]$$

- RegBayes inference

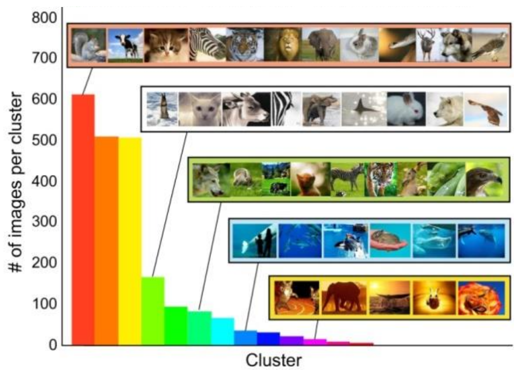
$$\begin{aligned} \min_{q(\mathcal{H})} \quad & \mathrm{KL}(q(\mathcal{H})\|p(\mathcal{H}|\mathbf{x})) + \Omega(q(\mathcal{H})) \\ \mathrm{s.t.}: \quad & q(\mathcal{H}) \in \mathcal{P}_{\mathrm{prob}}, \end{aligned}$$

- **Observations**:
  - when the posterior regularization equals to (or upper bounds) the empirical risk

$$\Omega(q(\mathcal{H})) \geq R_S(G_q)$$

  - the RegBayes classifiers tend to have PAC-Bayes guarantees.

RegBayes with Max-margin Posterior Regularization

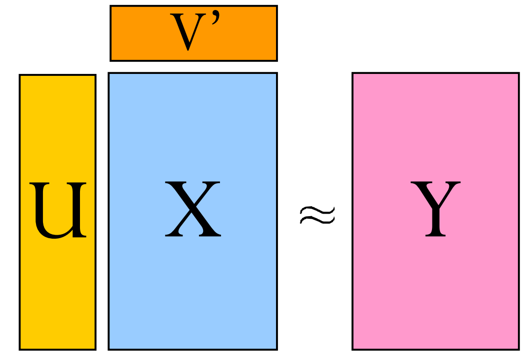


Infinite SVMs

(Zhu, Chen & Xing, ICML'11)

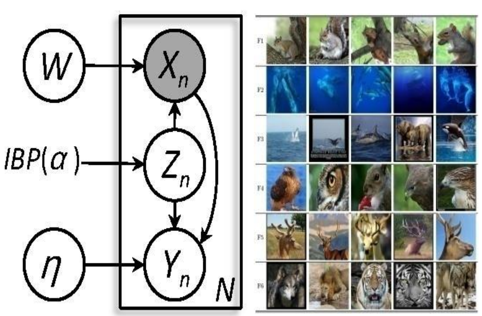
Nonparametric Max-margin Relational Models for Social Link Prediction

(Zhu, ICML'12)



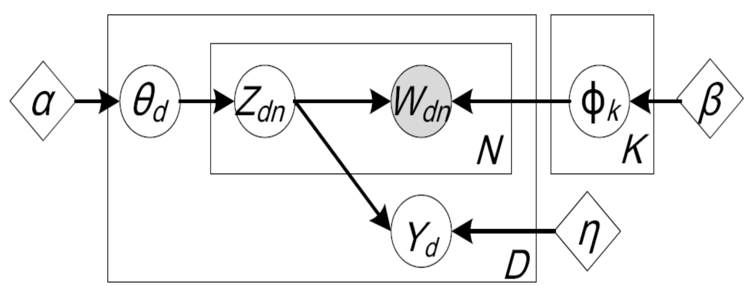
Nonparametric Max-margin Matrix Factorization

(Xu, Zhu, & Zhang, NIPS'12;
Xu, Zhu, & Zhang, ICML'13)



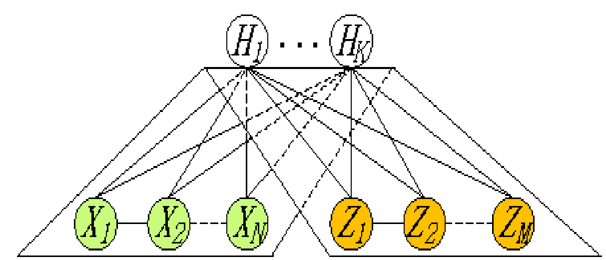
Infinite Latent SVMs

(Zhu, Chen & Xing, NIPS'11;
Zhu, Chen, & Xing, JMLR'14)



Max-margin Topics and Fast Inference

(Zhu, Ahmed & Xing, JMLR'12;
Jiang, Zhu, Sun & Xing, NIPS'12;
Zhu, Chen, Perkins & Zhang, ICML'13;
Zhu, Chen, Perkins & Zhang, JMLR'14)



Multimodal Representation Learning

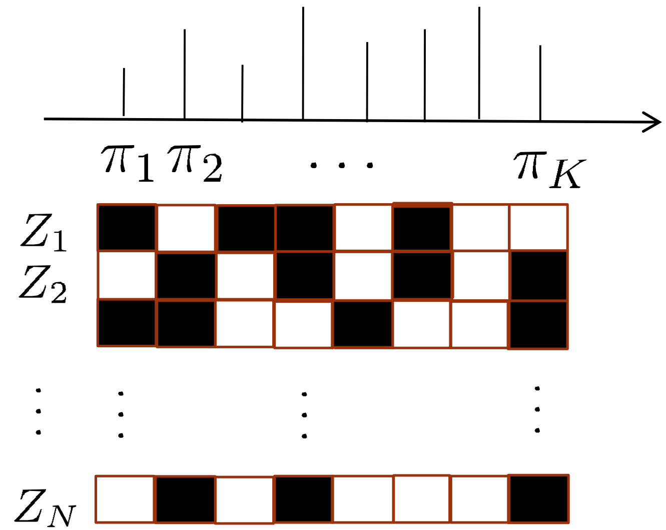
(Chen, Zhu & Xing, NIPS'10,
Chen, Zhu, Sun & Xing, PAMI'12;
Chen, Zhu, Sun, & Zhang, TNNLS'13)

Bayesian Latent Feature Models (finite)

- A finite Beta-Bernoulli latent feature model

$$\pi_k | \alpha \sim \mathrm{Beta}(\frac{\alpha}{K}, 1)$$

$$z_{ik} | \pi_k \sim \mathrm{Bernoulli}(\pi_k)$$

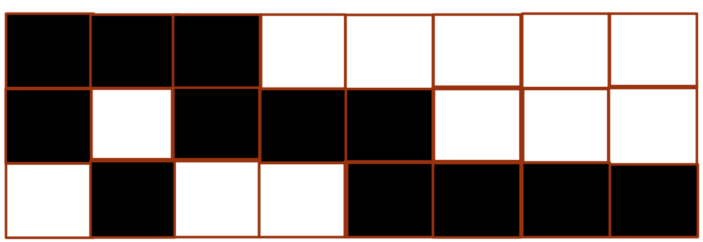


  - π_k is the relative probability of each feature being on
  - $z_{i.}$ are binary vectors, giving the latent structure that's used to generate the data, e.g.,

$$\mathbf{x}_i \sim \mathcal{N}(\eta^\top z_{i.}, \delta^2)$$

Indian Buffet Process

- A stochastic process on infinite binary feature matrices
- Generative procedure:
  - Customer 1 chooses the first K_1 dishes: $K_1 \sim \text{Poisson}(\alpha)$
  - Customer i chooses:
    - Each of the existing dishes with probability $\frac{m_k}{i}$
    - K_i additional dishes, where $K_i \sim \text{Poisson}(\frac{\alpha}{i})$



cust 1: new dishes 1-3

cust 2: old dishes 1,3; new dishes 4-5

cust 3: old dishes 2,5; new dishes 6-8

$$Z \sim \mathcal{IBP}(\alpha)$$

(Griffiths & Ghahramani, 2005)

Posterior Constraints – classification

- Suppose latent features $\mathbf{z}$ are given, we define *latent discriminant function*:

$$f(\mathbf{x}; \mathbf{z}, \boldsymbol{\eta}) = \boldsymbol{\eta}^\top \mathbf{z}$$

- Define *effective discriminant function* (reduce uncertainty):

$$f(\mathbf{x}; q(\mathbf{Z}, \boldsymbol{\eta})) = \mathbb{E}_{q(\mathbf{Z}, \boldsymbol{\eta})}[f(\mathbf{x}, \mathbf{z}; \boldsymbol{\eta})] = \mathbb{E}_{q(\mathbf{Z}, \boldsymbol{\eta})}[\boldsymbol{\eta}^\top \mathbf{z}]$$

- Posterior constraints with max-margin principle

$$\forall n \in \mathcal{I}_{\text{tr}} : y_n f(\mathbf{x}_n; q(\mathbf{Z}, \boldsymbol{\eta})) \geq 1 - \xi_n$$

- Convex *U* function

$$U(\xi) = C \sum_{n \in \mathcal{I}_{\text{tr}}} \xi_n$$

The RegBayes Problem

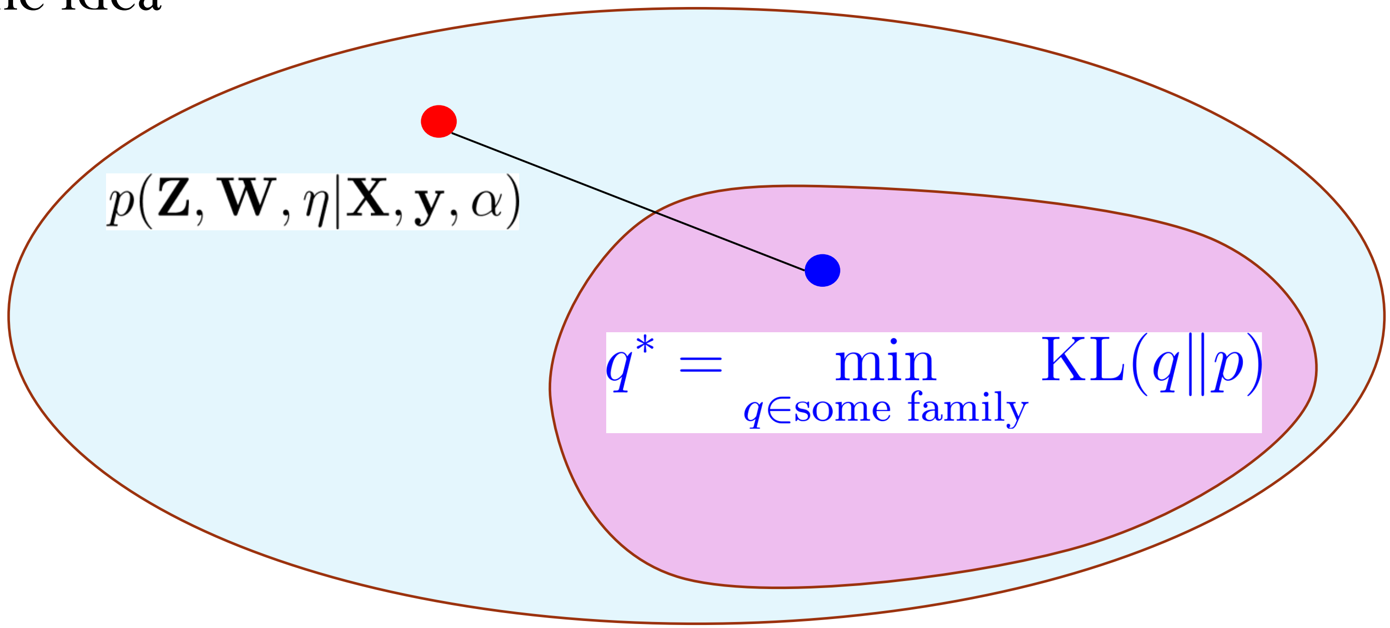
$$\min_{q(\mathbf{Z},\mathbf{W},\boldsymbol{\eta})} \quad \mathcal{L}(q(\mathbf{Z},\mathbf{W},\boldsymbol{\eta})) + 2c \cdot \mathcal{R}(q(\mathbf{Z},\mathbf{W},\boldsymbol{\eta}))$$

- where $\mathcal{L}(q) = \mathrm{KL}(q \| \pi(\mathbf{Z},\mathbf{W},\eta)) - \mathbb{E}_q[\log p(\mathbf{x}|\mathbf{Z},\mathbf{W})]$
- the hinge loss (posterior regularization) is

$$\mathcal{R}(q) = \sum_n \max(0, 1 - y_n f(\mathbf{x}_n; q(\mathbf{Z},\boldsymbol{\eta}))$$

Truncated Variational Inference

- The idea



  - Depends on a stick-breaking representation of IBP (Teh et al., 2007)
  - Truncated mean-field inference with an upper bound of features
  - Works reasonably well in practice

Posterior Regularization with a Gibbs Classifier

- Posterior distribution to learn

$$q(\mathbf{Z}, \boldsymbol{\eta})$$

- Gibbs classifier randomly draws a sample to make prediction

$$(\mathbf{Z}, \boldsymbol{\eta}) \sim q(\mathbf{Z}, \boldsymbol{\eta})$$

  - For classification, we measure the loss of classifier $(\mathbf{Z}, \boldsymbol{\eta})$

$$\mathcal{R}(\mathbf{Z}, \boldsymbol{\eta}) = \sum_n \max(0, 1 - y_n f(\mathbf{x}_n; \mathbf{Z}, \boldsymbol{\eta}))$$

  - It minimizes the expected loss

$$\mathcal{R}'(q) = \mathbb{E}_q \left[\sum_n \max(0, 1 - y_n f(\mathbf{x}_n; \mathbf{Z}, \boldsymbol{\eta})) \right]$$

Comparison

- Expected hinge-loss is an upper bound

$$\mathcal{R}'(q) \geq \mathcal{R}(q)$$

- For averaging classifier, the RegBayes problem is suitable for variational inference with truncation (Zhu et al., arXiv, 2013)

- For Gibbs classifier, the RegBayes problem is suitable for MCMC without truncation

Multi-task Learning (MTL)

- **[Wikipedia]** MTL is an approach to machine learning that learns a problem together with other related problems, using a ***shared representation***

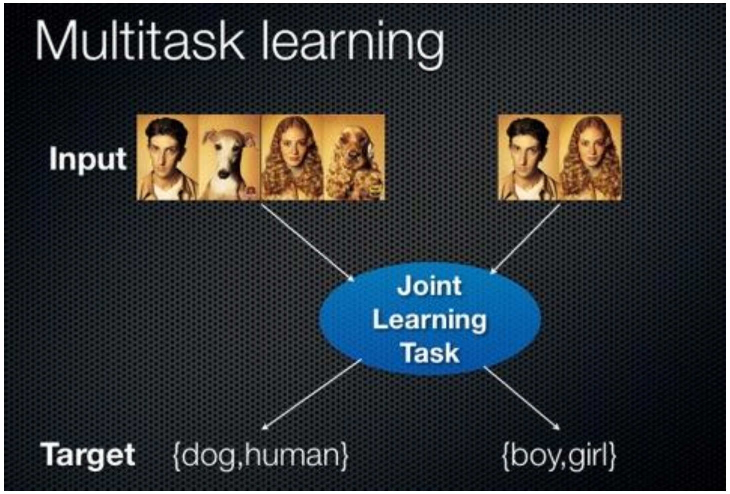


Figure from Wikipedia
Author: Kilian Weinberger

- The goal of MTL is to improve the performance of learning algorithms by learning classifiers for multiple tasks jointly

- It works particularly well if these tasks have some commonality and are generally slightly under sampled

Multi-task Representation Learning

- Assumption:
  - common underlying representation across tasks

- Representative works:
  - ASO (alternating structure optimization): learn a small set of shared features across tasks [Ando & Zhang, 2005]
  - Convex feature learning via sparse norms [Argyriou et al., 2006]

Basic Setup of the Learning Paradigm

- Tasks: $m = 1, \cdots, M$
- N examples per task

$$(\mathbf{x}_{m1}, y_{m1}), \cdots, (\mathbf{x}_{mN}, y_{mN}) \in \mathbb{R}^D \times \mathbb{R}$$

- Estimate

$$f_m : \mathbb{R}^D \to \mathbb{R}, \ \forall m = 1, \cdots, M$$

- Consider features

$$h_1(\mathbf{x}), \ \cdots, h_K(\mathbf{x})$$

- Predict using functions

$$f_m(\mathbf{x}) = \sum_{k=1}^{K} \eta_{mk} h_k(\mathbf{x})$$

Learning a Projection Matrix

- Tasks: $m = 1, \cdots, M$
- N examples per task

$$(\mathbf{x}_{m1}, y_{m1}), \cdots, (\mathbf{x}_{mN}, y_{mN}) \in \mathbb{R}^D \times \mathbb{R}$$

- Estimate

$$f_m : \mathbb{R}^D \to \mathbb{R}, \ \forall m = 1, \cdots, M$$

- Consider features

$$h_k(\mathbf{x}) = \mathbf{z}_k^\top \mathbf{x}, \ \ k = 1, \cdots, \infty$$

- Predict using functions ($\mathbf{Z}$ is a $D \times \infty$ projection matrix)

$$f_m(\mathbf{x}; \mathbf{Z}, \boldsymbol{\eta}) = \sum_{k=1}^{\infty} \eta_{mk} (\mathbf{z}_k^\top \mathbf{x}) = \boldsymbol{\eta}_m^\top (\mathbf{Z}^\top \mathbf{x})$$

Max-margin Posterior Regularizations

- Similar as in infinite latent SVMs
  - Averaging classifier

$$y_{mn}\mathbb{E}_q[f_m(\mathbf{x}_{mn}; \mathbf{Z}, \boldsymbol{\eta})] \geq 1 - \xi_{mn}$$

    - The hinge loss

$$\mathcal{R} = \sum_{m,n \in \mathcal{I}_{\text{tr}}^m} \max\left(0, 1 - y_{mn}\mathbb{E}_q[f_m(\mathbf{x}_{mn}; \mathbf{Z}\boldsymbol{\eta})]\right)$$

  - Gibbs classifier

$$\mathcal{R}' = \mathbb{E}_q\left[\sum_{m,n \in \mathcal{I}_{\text{tr}}^m} \max\left(0, 1 - y_{mn}f_m(\mathbf{x}_{mn}; \mathbf{Z}\boldsymbol{\eta})\right)\right]$$

Experimental Results

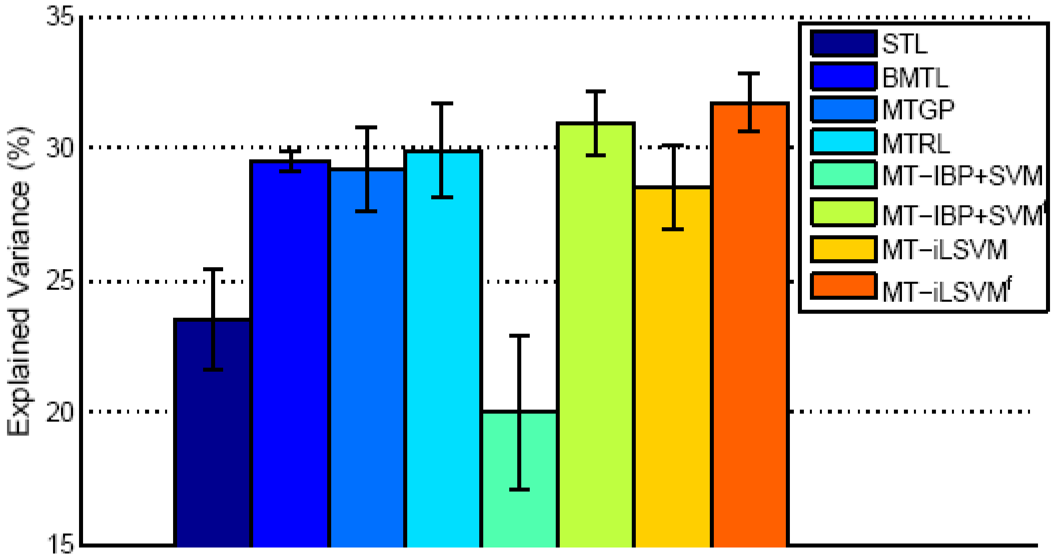
- Multi-label Classification (multiple binary classification)
  - Accuracy and F1 scores (Micro & Macro) on Yeast and Scene datasets

Model	Acc	F1-Macro	F1-Micro
YaXue [Xue et al., 2007]	0.5106	0.3897	0.4022
Piyushrai [Piyushrai et al., 2010]	0.5424	0.3946	0.4112
MT-iLSVM	0.5792 ± 0.003	0.4258 ± 0.005	0.4742 ± 0.008
Gibbs MT-iLSVM	0.5851 ± 0.005	0.4294 ± 0.005	0.4763 ± 0.006

Model	Acc	F1-Macro	F1-Micro
YaXue [Xue et al., 2007]	0.7765	0.2669	0.2816
Piyushrai [Piyushrai et al., 2010]	0.7911	0.3214	0.3226
MT-iLSVM	0.8752 ± 0.004	0.5834 ± 0.026	0.6148 ± 0.020
Gibbs MT-iLSVM	0.8855 ± 0.004	0.6494 ± 0.011	0.6458 ± 0.011

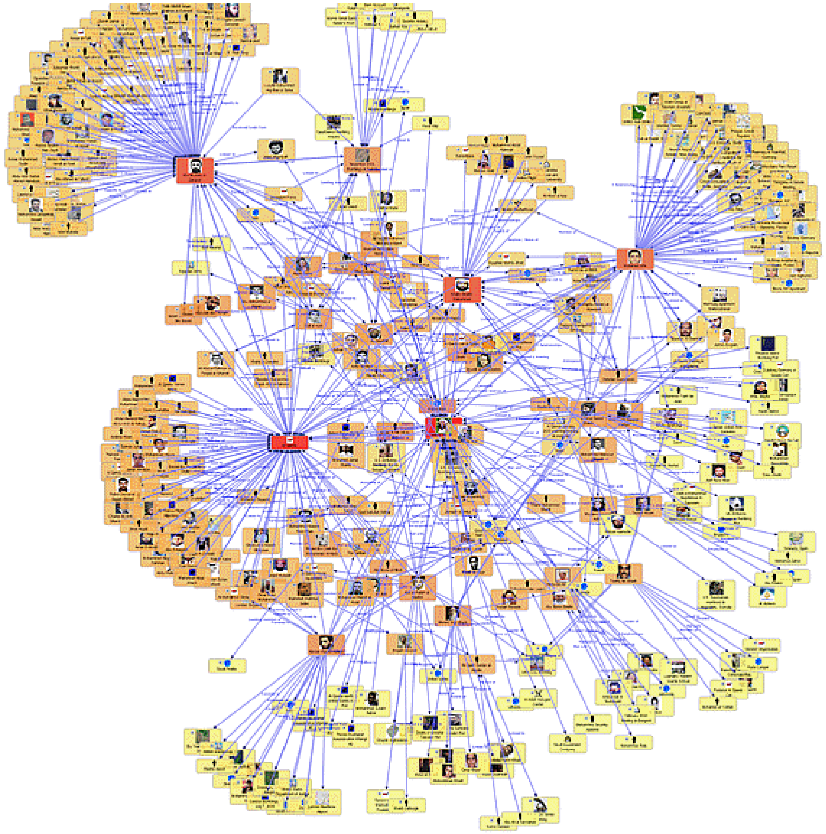
Experimental Results

- Multi-task Regression
  - School dataset (139 regression tasks) – a standard dataset for evaluating multi-task learning
  - Percentage of *explained variance* (higher, better)



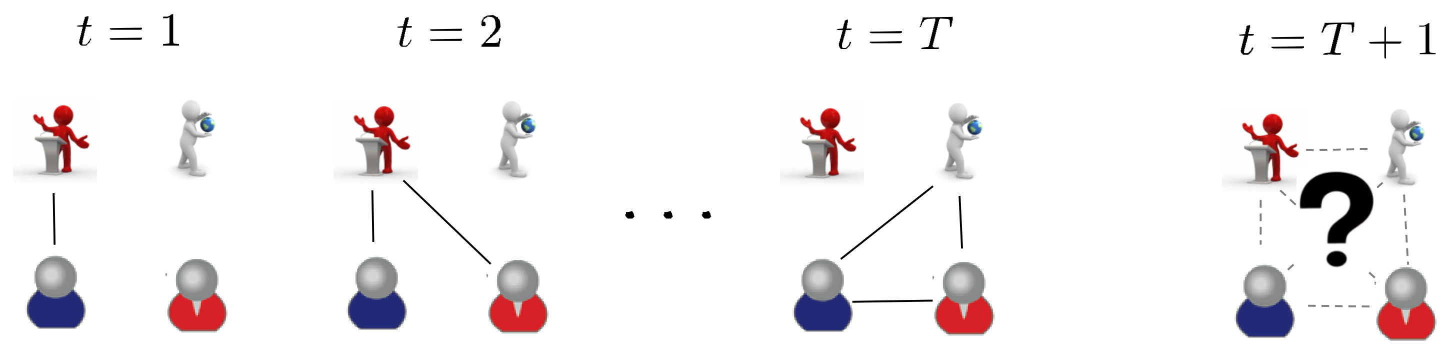
Link Prediction

- Network structures are usually unclear, unobserved, or corrupted with noise

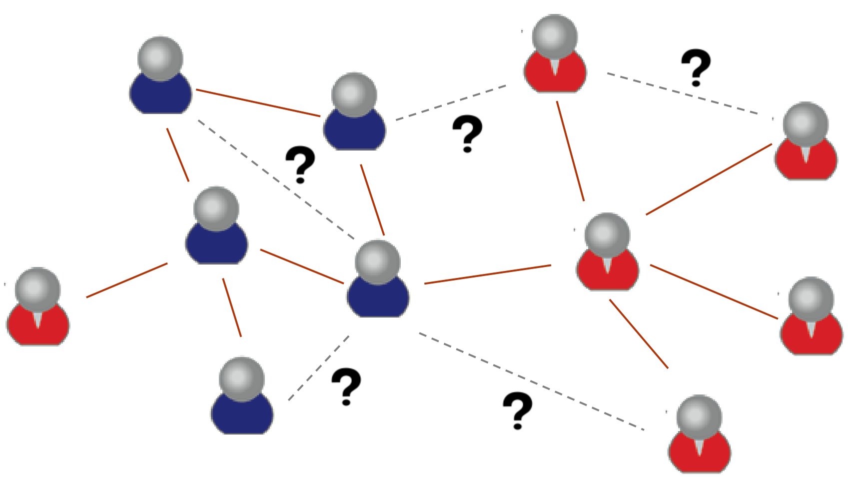


Link prediction – task

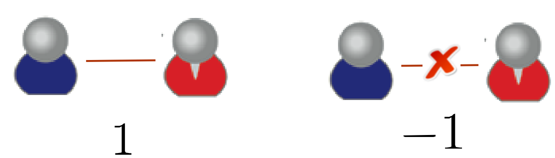
- Dynamic networks



- Static networks



We treat it as a supervised learning task with 1/-1 labels



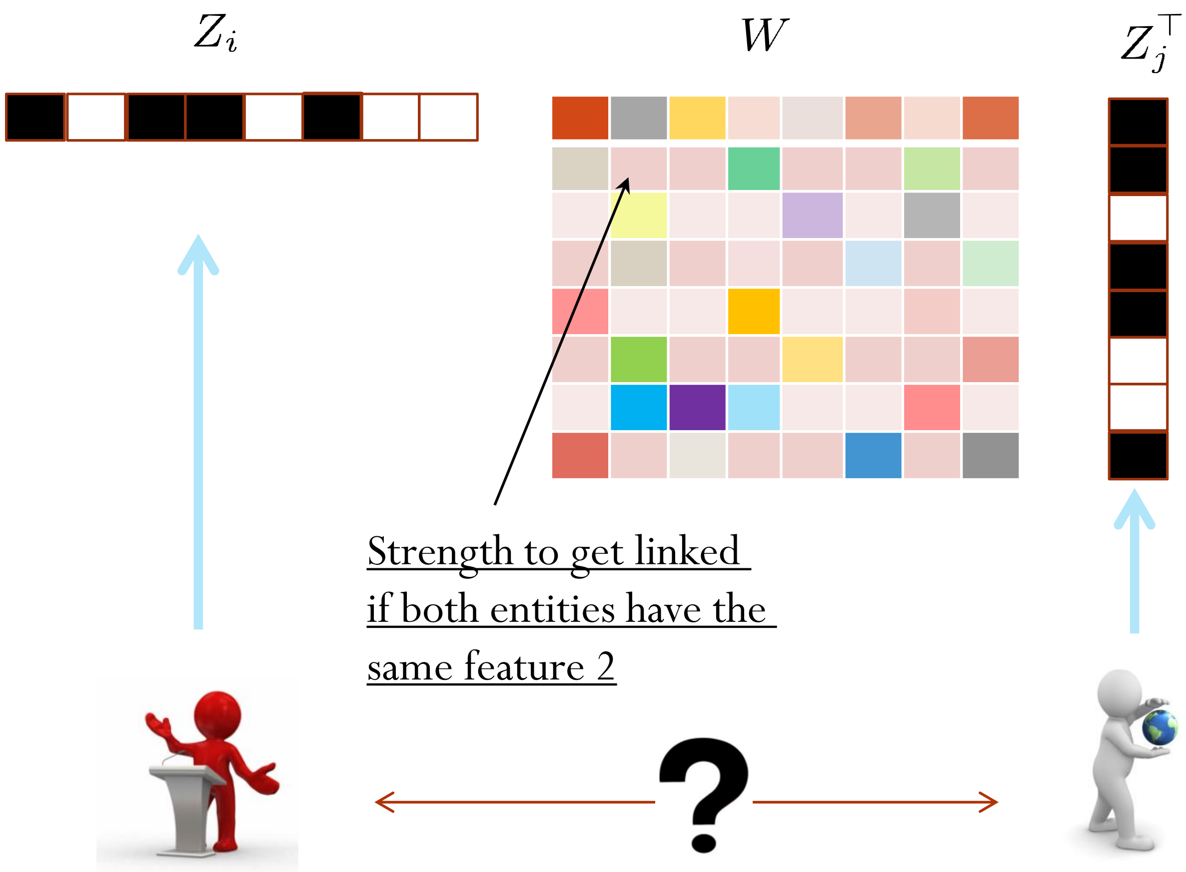
Link Prediction as Supervised Learning

- Building classifiers with manually extracted features from networks
  - Topological features
    - Shortest distance, number of common neighbors, Jaccard's coefficient, etc.
  - Attributes about individual entities
    - E.g., the papers an authors has published
    - * an aggregation function is needed to combine attributes for each pair
  - Proximity features
    - E.g., two authors are close, if their research work evolves around a large set of identical keywords

[Hasan et al., 2006]

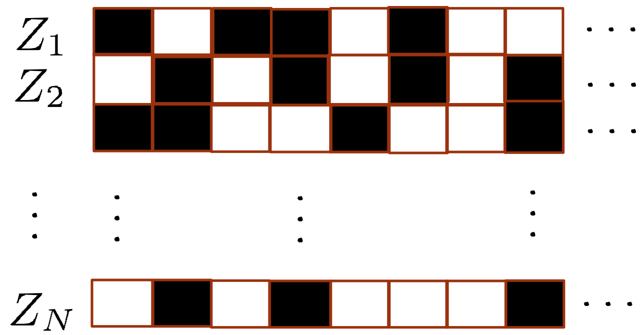
Discriminant Function with Latent Features

$$f(Z_i, Z_j; X_{ij}, W, \eta) = Z_i W Z_j^\top$$



Two Key Issues

- N entities → a latent feature matrix Z



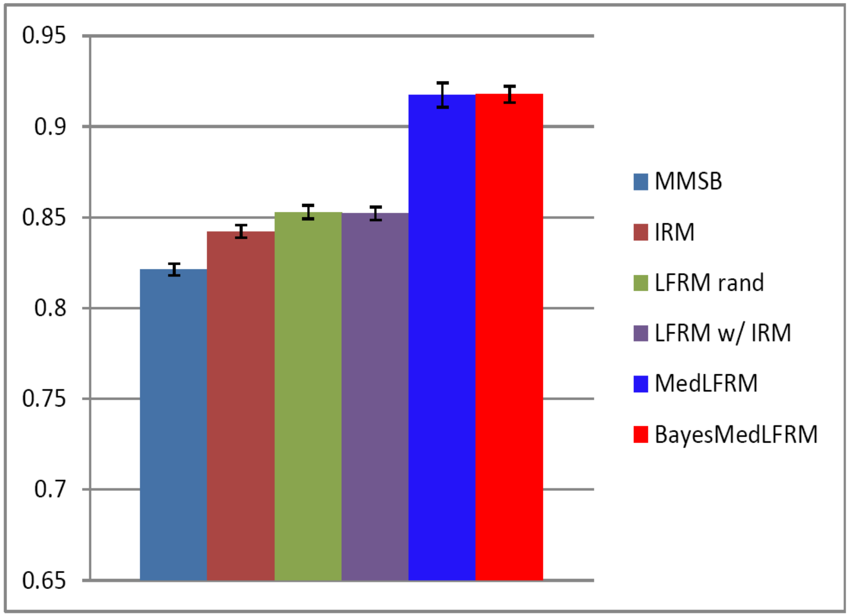
- How many columns (i.e., features) are sufficient?
  → a stochastic process to infer it from data
- What learning principle is good?
  → large-margin principle to learn classifiers

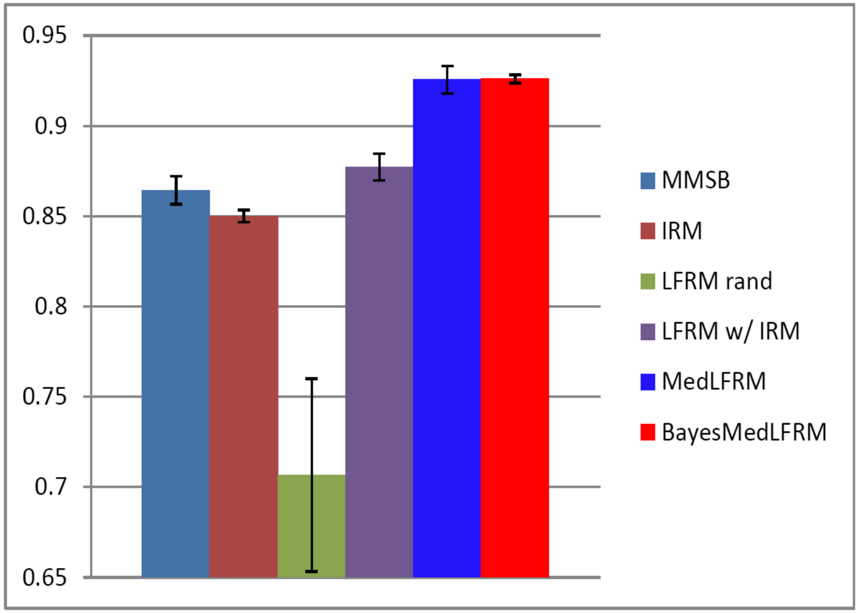
Max-Margin Nonparametric Latent Feature Models for Link Prediction.

[Zhu, ICML 2012]

Some Results

- AUC – area under ROC curve (higher, better)
- Two evaluation settings
  - Single – learn separate models for different relations, and average the AUC scores;
  - Global – learn one common model (i.e., features) for all relations



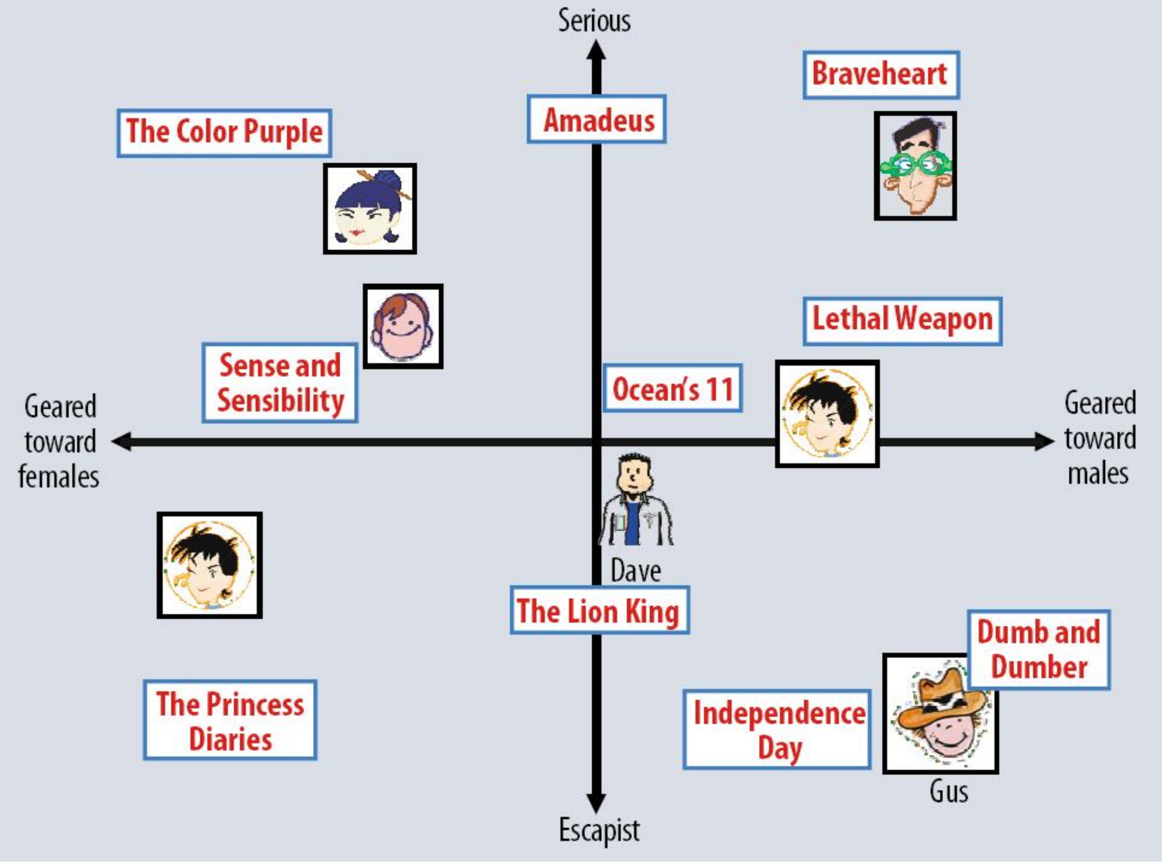


Country Relationships

Collaborative Filtering in Our Life

Latent Factor Methods

- Characterize both items & users on say 20 to 100 factors inferred from the rating patterns

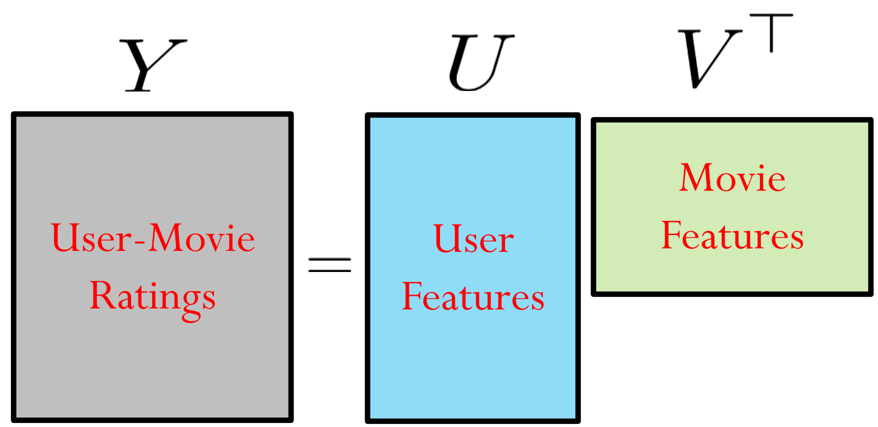


[Y. Koren, R. Bell & C. Volinsky, IEEE, 2009]

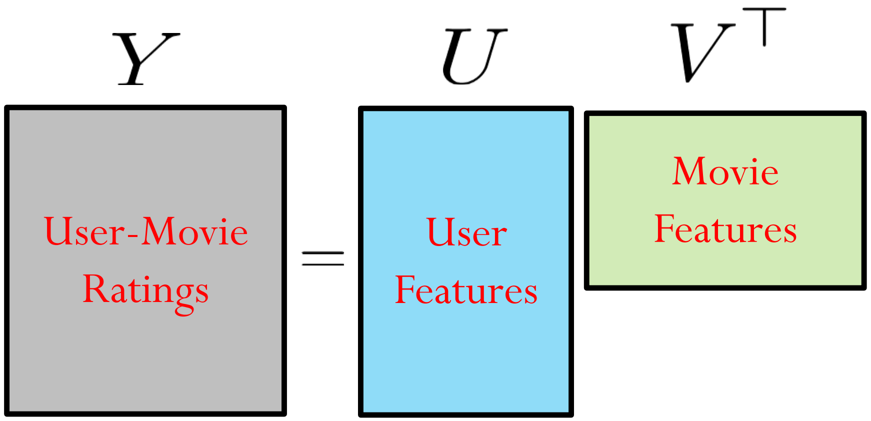
Matrix Factorization

- Some of the most successful latent factor models are based on matrix factorization

item / user	1	2	3	4 …
1	★★★★	?	★★★	★
2	★★★	?	?	★★
3	★★★★	★	?	?
⋮				



Two Key Issues



- How many columns (i.e., features) are sufficient?
  - → a stochastic process to infer it from data
- What learning principle is good?
  - → large-margin principle to learn classifiers

Nonparametric Max-margin Matrix Factorization for Collaborative Prediction

[Xu, Zhu, & Zhang, NIPS 2012]

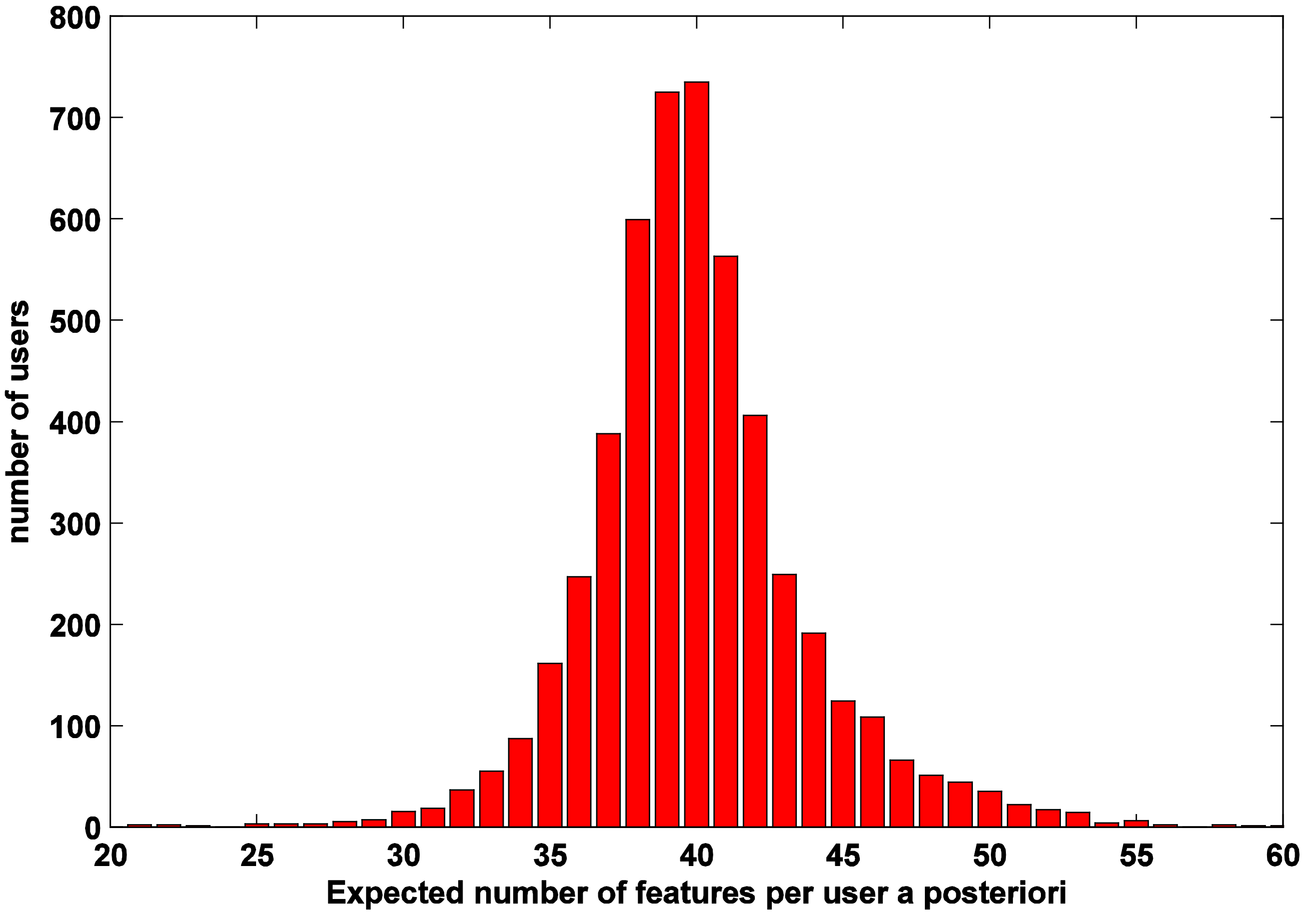
Experiments

- Data sets:
  - **MovieLens**: 1M anonymous ratings of 3,952 movies made by 6,040 users
  - **EachMovie**: 2.8M ratings of 1,628 movies made by 72,916 users
- Overall results on **Normalized Mean Absolute Error (NMAE)** (**the lower, the better**)

Table 1: NMAE performance of different models on MovieLens and EachMovie.

	MovieLens		**EachMovie**	
Algorithm	**weak**	**strong**	**weak**	**strong**
M^3F [11]	$.4156 \pm .0037$	$.4203 \pm .0138$	$.4397 \pm .0006$	$.4341 \pm .0025$
PMF [13]	$.4332 \pm .0033$	$.4413 \pm .0074$	$.4466 \pm .0016$	$.4579 \pm .0016$
BPMF [12]	$.4235 \pm .0023$	$.4450 \pm .0085$	$.4352 \pm .0014$	$.4445 \pm .0005$
M^3F^*	$.4176 \pm .0016$	$.4227 \pm .0072$	$.4348 \pm .0023$	$.4301 \pm .0034$
iPM^3F	$\mathbf{.4031} \pm .0030$	$.4135 \pm .0109$	$\mathbf{.4211} \pm .0019$	$\mathbf{.4224} \pm .0051$
$iBPM^3F$	$.4050 \pm .0029$	$\mathbf{.4089} \pm .0146$	$.4268 \pm .0029$	$.4403 \pm .0040$

Expected Number of Features per User



Fast Sampling Algorithms

- See our paper [Xu, Zhu, & Zhang, ICML2013] for details

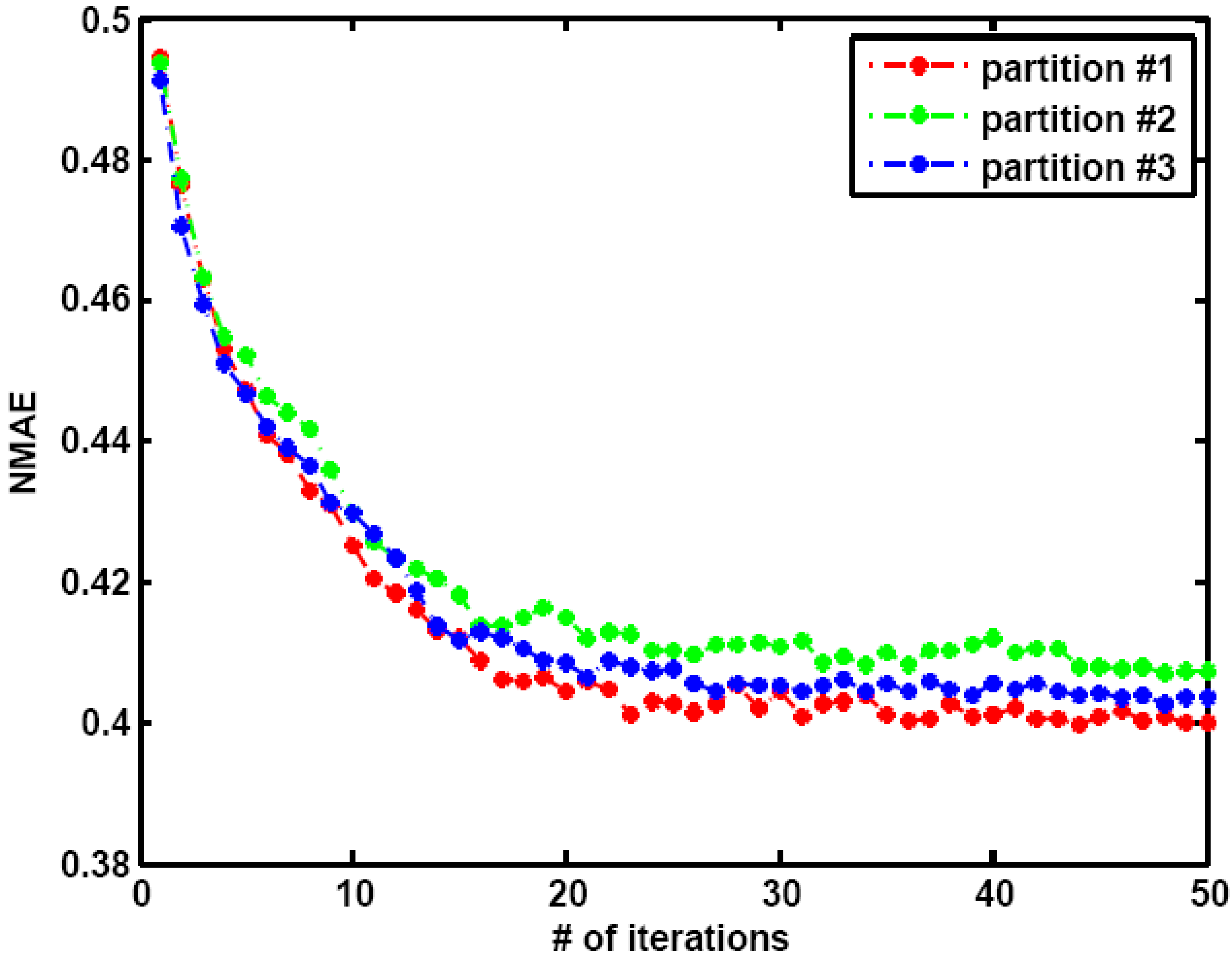
	MovieLens		EachMovie	
Algorithm	**weak**	**strong**	**weak**	**strong**
M^3F	$.4156 \pm .0037$	$.4203 \pm .0138$	$.4397 \pm .0006$	$.4341 \pm .0025$
bcd M^3F	$.4176 \pm .0016$	$.4227 \pm .0072$	$.4348 \pm .0023$	$.4301 \pm .0034$
Gibbs M^3F	$.4037 \pm .0005$	$.4040 \pm .0055$	$.4134 \pm .0017$	$.4142 \pm .0059$
iPM^3F	$.4031 \pm .0030$	$.4135 \pm .0109$	$.4211 \pm .0019$	$.4224 \pm .0051$
Gibbs iPM^3F	$.4080 \pm .0013$	$.4201 \pm .0053$	$.4220 \pm .0003$	$.4331 \pm .0057$

Algorithm	MovieLens	EachMovie	Iters
M^3F	5h	15h	100
bcd M^3F	4h	10h	50
Gibbs M^3F	0.11h	0.35h	50
iPM^3F	4.6h	5.5h	50
Gibbs iPM^3F	0.68h	0.70h	50

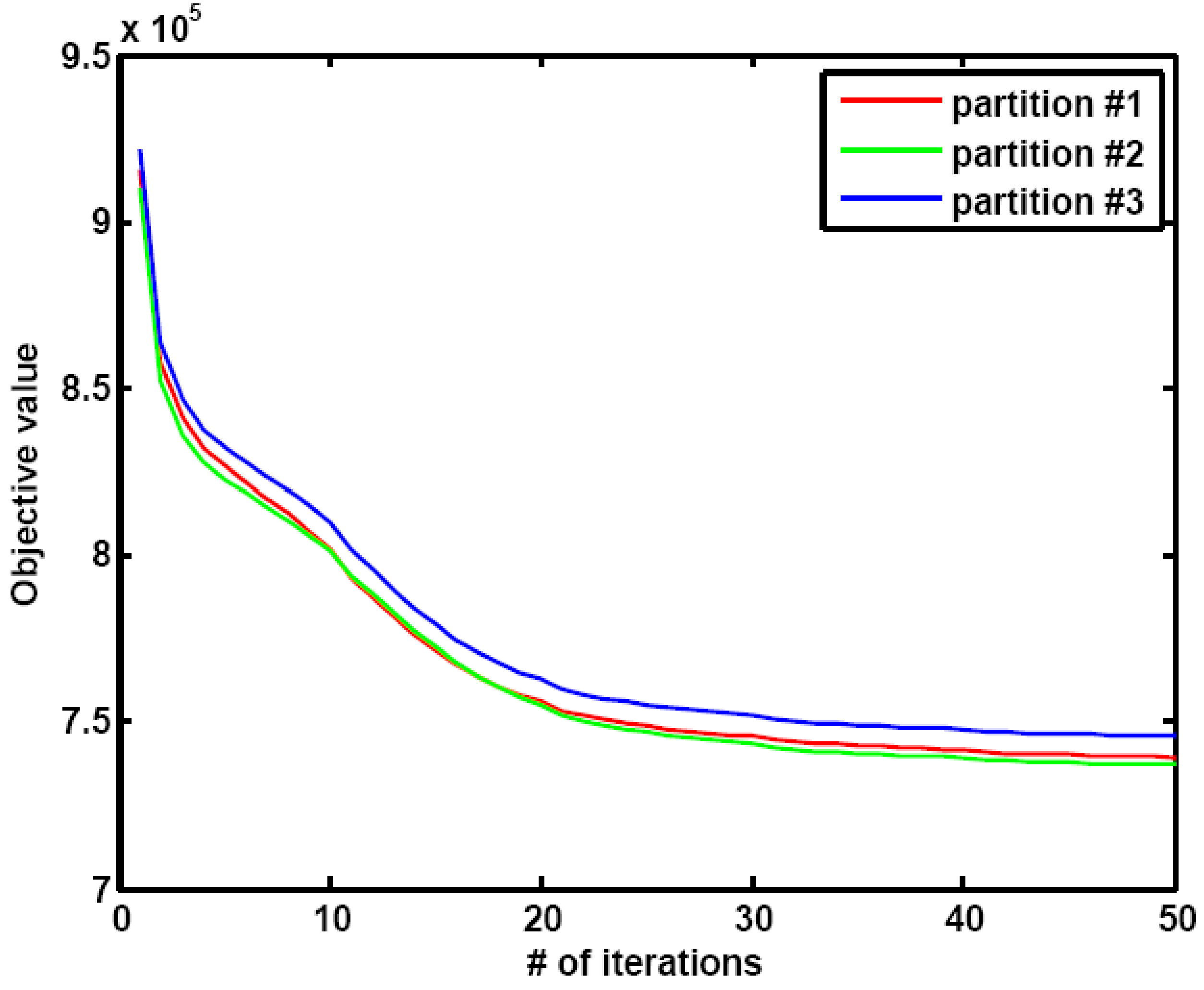
30 times faster!

8 times faster!

Prediction Performance during Iterations



Objective Value during Iterations



Markov IBP and Time Series

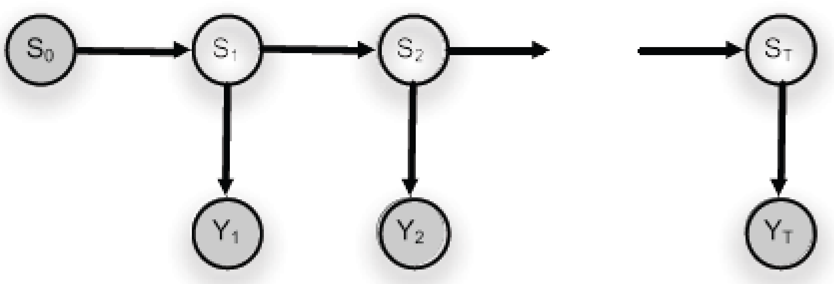


Figure 1: The Hidden Markov Model

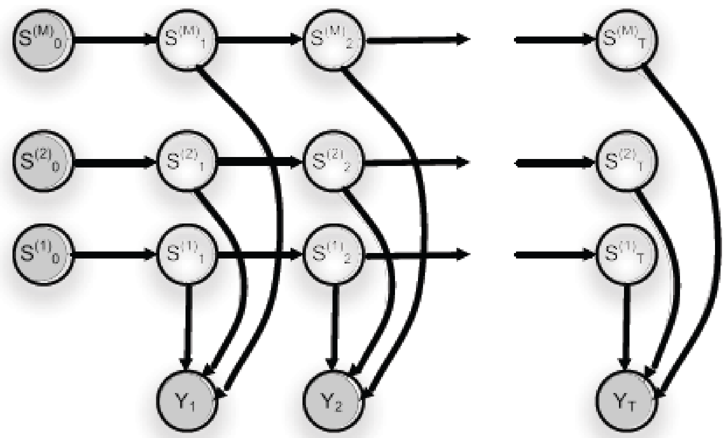


Figure 2: The Factorial Hidden Markov Model

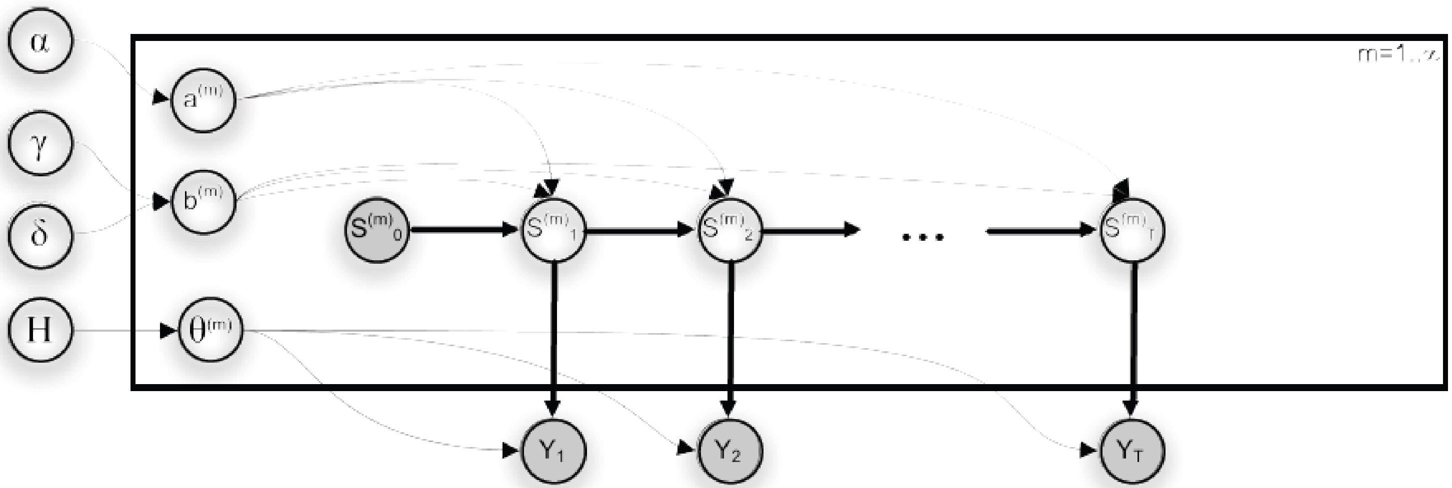
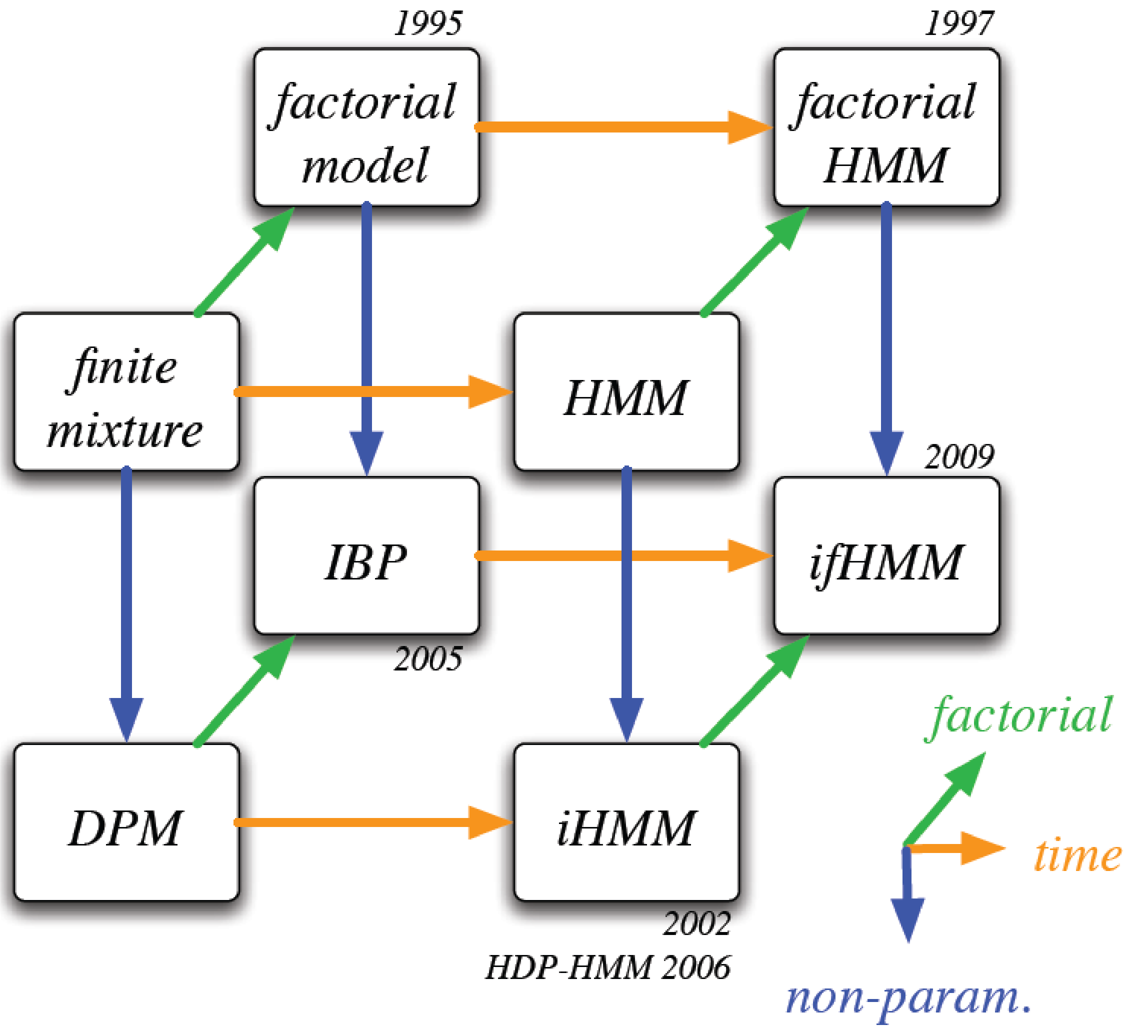


Figure 3: The Infinite Factorial Hidden Markov Model

Big Picture


1995
factorial model
1997
factorial HMM
finite mixture
HMM
2009
IBP
ifHMM
2005
DPM
iHMM
2002
HDP-HMM 2006
factorial
time
non-param.

References

- Chap. 13 of Pattern Recognition and Machine Learning, Bishop, 2006
- Probabilistic Principal Component Analysis. M. Tipping, C. Bishop, 1999.
- Infinite Latent Feature Models and the Indian Buffet Process. T. Griffiths, Z. Ghahramani, 2005.
- The Indian Buffet Process: Scalable Inference and Extensions. Finale Doshi-Velez, 2009.
- Max-margin Nonparametric Latent Feature Models for Link Prediction. J. Zhu, 2012
- Infinite Latent SVMs for Classification and Multi-task Learning, J. Zhu, N. Chen, & E. Xing, 2011
- Bayesian Inference with Posterior Regularizations and its Applications to Infinite Latent SVMs. J. Zhu, N. Chen, & E. Xing, 2014