Machine Learning 1.13: Latent Variables

Tom S. F. Haines
T.S.F.Haines@bath.ac.uk

UNIVERSITY OF BATH

Knowing everything

- Previously: We know everything! (during training)
- Now: We don't

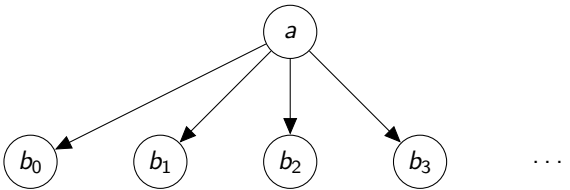
UNIVERSITY OF BATH

Knowing everything

- Previously: We know everything! (during training)
- Now: We don't

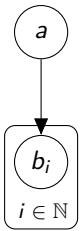
- Structure: Unknown, incomplete, best guess
- RVs missing when training – **latent**
  (often hypothetical RVs)

Aside: Plate notation



- Representing repetition: Tree = confusing

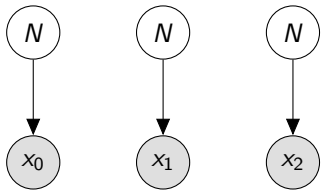
Aside: Plate notation

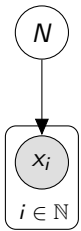


- Representing repetition: Tree = confusing
- **Plate notation** solves this
- Adds indices
- Can contain multiple RVs

One model

- Parameter sharing – explicit with plates:





(e.g. Gaussian distribution N over x)

- Much easier for complex models
- Important for fully Bayesian

UNIVERSITY OF BATH

This lecture

- Two examples:
  - Latent Semantic Analysis (recommender system)
  - Gaussian Mixture Model (clustering / density estimation)

UNIVERSITY OF BATH

Example I: Recommender system

- Estimates what you like/dislike, e.g. films
- Amazon, Netflix etc. all use these

UNIVERSITY OF BATH

Example I: Recommender system

- Estimates what you like/dislike, e.g. films
- Amazon, Netflix etc. all use these

- Two approaches:
  - Content-based filtering:
    Each product has features. Match user to products with features similar to previously liked (e.g. per user logistic regression model).
  - Collaborative filtering:
    Identify users with similar likes. Assume they have similar opinions for products where one has given an opinion and the other has not (e.g. clustering users).

Hybrids used in practise

Latent semantic analysis

- Also called *Latent Semantic Indexing*
- Mathematically similar to *Principal Component Analysis*

UNIVERSITY OF BATH

Latent semantic analysis

- Also called *Latent Semantic Indexing*
- Mathematically similar to *Principal Component Analysis*

- Usually described for text analysis
- Recommender systems people sometimes call it *SVD*.
  (as in the matrix factorisation – seriously stupid name)

UNIVERSITY OF BATH

Latent semantic analysis

- Also called *Latent Semantic Indexing*
- Mathematically similar to *Principal Component Analysis*

- Usually described for text analysis
- Recommender systems people sometimes call it *SVD*.
  (as in the matrix factorisation – seriously stupid name)

- Has multiple equivalent graphical models (this is normal)
- Not probabilistic (though easy to fix)

UNIVERSITY OF BATH

Unknown attributes

- Have unknown attributes
- Each film has attributes
- Each user likes or dislikes each attribute

UNIVERSITY OF BATH

Unknown attributes

- Have unknown attributes
- Each film has attributes
- Each user likes or dislikes each attribute

- Seems reasonable:
  e.g. Robert likes action films. Star Wars is an action film.
  $\therefore$ Robert likes Star Wars

- With multiple attributes:
  e.g. Susan likes action films but really hates fantasy. Star Wars is a fantasy action film.
  $\therefore$ Susan dislikes Star Wars a little bit

UNIVERSITY OF BATH

Unknown attributes

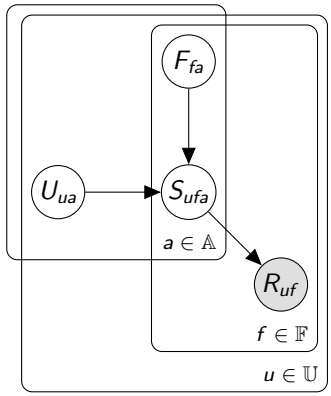
- Have unknown attributes
- Each film has attributes
- Each user likes or dislikes each attribute

- Seems reasonable:
  e.g. Robert likes action films. Star Wars is an action film.
  $\therefore$ Robert likes Star Wars

- With multiple attributes:
  e.g. Susan likes action films but really hates fantasy. Star Wars is a fantasy action film.
  $\therefore$ Susan dislikes Star Wars a little bit

- **Attributes unknown** $\therefore$ Inferred from data
- May not have human interpretation. Does this matter?

Graphical model



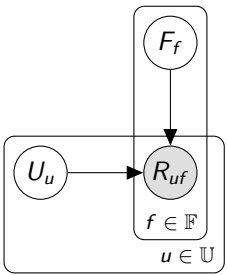
- a – Attribute index, $\in \mathbb{A}$
- u – User index, $\in \mathbb{U}$
- f – Film index, $\in \mathbb{F}$

- F_{fa} – Strength of attribute a for film f
- U_{ua} – How much user u likes attribute a

- S_{ufa} – Score: Effect of attribute a for user u on film f (multiplication)
- R_{uf} – Rating given by user u of film f (summation)

- Training data: R_{uf} **only**, extremely sparse

Simplified



- Usually simplify by hiding attributes (RVs now vectors indexed by attribute)
- This is the PCA graphical model!

Fitting to data

Analytic:

1. Calculate A, where $A_{uf} = R_{uf}$ when known, zero otherwise

Fitting to data

Analytic:

1. Calculate A, where $A_{uf} = R_{uf}$ when known, zero otherwise
2. Perform SVD, obtain $A = U\Sigma V^T$. Assume Σ (singular values) sorted high to low

Fitting to data

Analytic:

1. Calculate A, where $A_{uf} = R_{uf}$ when known, zero otherwise
2. Perform SVD, obtain $A = U\Sigma V^T$. Assume Σ (singular values) sorted high to low
3. Keep the top N columns of U and V, where N is the number of attributes (hyper-parameter)

Fitting to data

Analytic:

1. Calculate A, where $A_{uf} = R_{uf}$ when known, zero otherwise
2. Perform SVD, obtain $A = U\Sigma V^T$. Assume Σ (singular values) sorted high to low
3. Keep the top N columns of U and V, where N is the number of attributes (hyper-parameter)
4. Scale column n by $\sqrt{\Sigma_n}$ (U and V)

U contains user attribute weights; V film attribute weights

Fitting to data

Analytic:

1. Calculate A, where $A_{uf} = R_{uf}$ when known, zero otherwise
2. Perform SVD, obtain $A = U\Sigma V^T$. Assume Σ (singular values) sorted high to low
3. Keep the top N columns of U and V, where N is the number of attributes (hyper-parameter)
4. Scale column n by $\sqrt{\Sigma_n}$ (U and V)

U contains user attribute weights; V film attribute weights

In practise:
Too much data, use gradient descent instead

Results

From Netflix challenge:

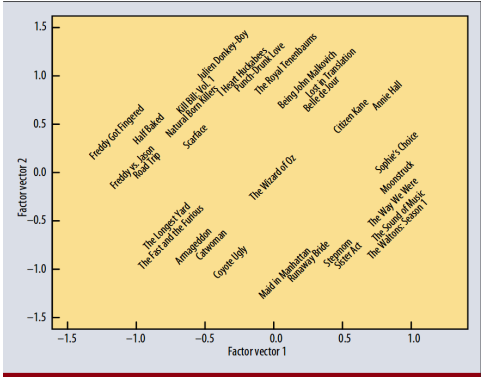


Figure 3. The first two vectors from a matrix decomposition of the Netflix Prize data. Selected movies are placed at the appropriate spot based on their factor vectors in two dimensions. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky independent films.

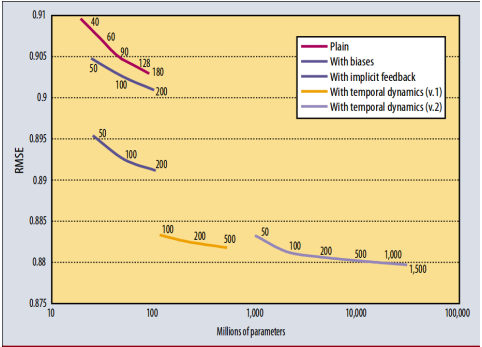


Figure 4. Matrix factorization models' accuracy. The plots show the root-mean-square error of each of four individual factor models (lower is better). Accuracy improves when the factor model's dimensionality (denoted by numbers on the charts) increases. In addition, the more refined factor models, whose descriptions involve more distinct sets of parameters, are more accurate. For comparison, the Netflix system achieves RMSE = 0.9514 on the same dataset, while the grand prize's required accuracy is RMSE = 0.8563.

UNIVERSITY OF BATH

Latent semantic analysis notes

- Identical to PCA except:
  - Subtracting means optional (recommended for numerical stability)
  - Outputs divided by standard deviation (doesn't preserve Euclidean distance)
  - Trained with sparse data

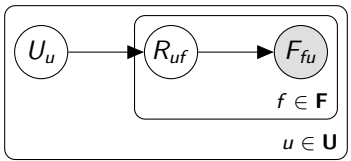
- Needs lots of data – does badly for users/films with few ratings (cold start problem)

Latent semantic analysis notes

- Identical to PCA except:
  - Subtracting means optional (recommended for numerical stability)
  - Outputs divided by standard deviation (doesn't preserve Euclidean distance)
  - Trained with sparse data

- Needs lots of data – does badly for users/films with few ratings (cold start problem)

Traditional graphical model: (weird conditional structure for recommender system)

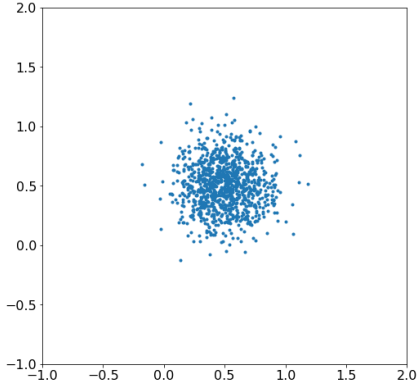


(user → document, rating → topic, film → word)

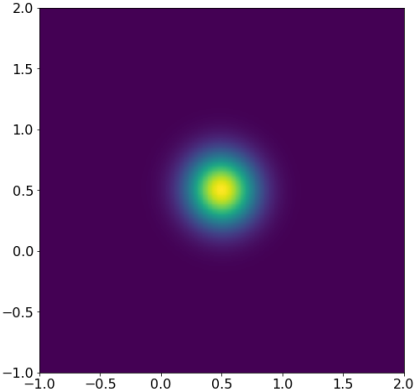
Example II: Density estimation

- Density estimation:
  - Input: Samples, x, drawn from unknown probability distribution D
  - Output: Estimate of D

Input:



Output: (Gaussian)



UNIVERSITY OF BATH

Example II: Density estimation

- Density estimation:
  - Input: Samples, x, drawn from unknown probability distribution D
  - Output: Estimate of D

- Examples:
  - Histogram (normalised)
  - Fitting a Gaussian
  - Gaussian mixture model (GMM)

UNIVERSITY OF BATH

Example II: Density estimation

- Density estimation:
  - Input: Samples, x, drawn from unknown probability distribution D
  - Output: Estimate of D

- Examples:
  - Histogram (normalised)
  - Fitting a Gaussian
  - Gaussian mixture model (GMM)

- Uses:
  - Visualisation
  - Clustering (it's probabilistic k-means)
  - Abnormality detection

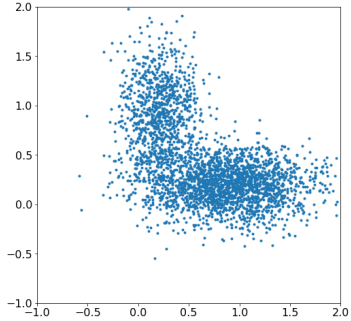
  - Classification via Bayes rule
  - Regression via marginalisation
  - Part of larger models, e.g. voice recognition

Gaussian mixture model

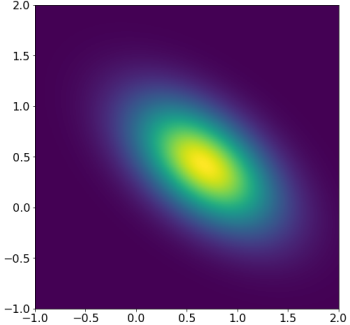
- Sum of Gaussian distributions:

$$P(x|\alpha,\mu,\Sigma) = \sum_{z=1}^{K} \alpha_z \mathcal{N}(x|\mu_z,\Sigma_z)$$

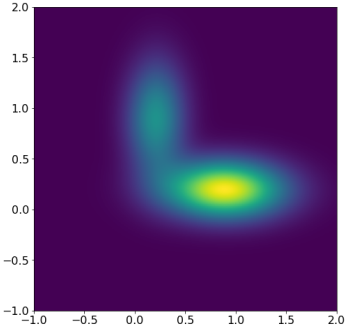
Input:



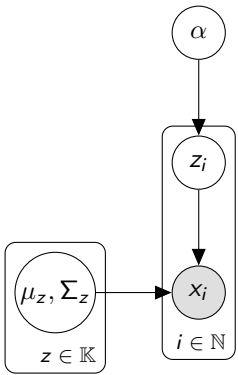
Gaussian:



GMM with 2 components:



Graphical model



- i – Exemplar index, $\in \mathbb{N}$
- z – Component index, $\in \mathbb{K}$
- μ_z, Σ_z – Mean and covariance for component z
- α – Categorical distribution over component membership
- z_i – Which component feature vector i belongs to
- x_i – Feature vector for exemplar i

Fitting to data

- No analytical solution
- Cliques too large for belief propagation

- New algorithm: **Expectation maximisation** (EM)

UNIVERSITY OF BATH

Expectation maximisation

- Maximum likelihood and maximum a posteriori only
- Comparable to gradient descent:
  - Initialised
  - Series of steps to improve model fit to data
  - Get stuck in local minima

Expectation maximisation

- Maximum likelihood and maximum a posteriori only
- Comparable to gradient descent:
  - Initialised
  - Series of steps to improve model fit to data
  - Get stuck in local minima
- Split RVs into three groups:
  - **Model RVs**: Latent variables shared between all data
  - **Missing RVs**: Latent variables associated with data (confusingly, often called latent variables)
  - **Observed RVs**: The data

Expectation maximisation

- Maximum likelihood and maximum a posteriori only
- Comparable to gradient descent:
  - Initialised
  - Series of steps to improve model fit to data
  - Get stuck in local minima
- Split RVs into three groups:
  - **Model RVs**: Latent variables shared between all data
  - **Missing RVs**: Latent variables associated with data (confusingly, often called latent variables)
  - **Observed RVs**: The data
- Two steps:
  - **E-step**: Calculate **likelihood function** of missing values with model parameters fixed (Often referred to as calculating **expectation**. It isn't.)
  - **M-Step**: **Maximise** model parameters given expectation of missing values

  Alternate until convergence

UNIVERSITY OF BATH

Aside: K-means

- K-means – alternate between:
  1. Assign points to nearest cluster centre
  2. Update cluster centres to mean of assigned points

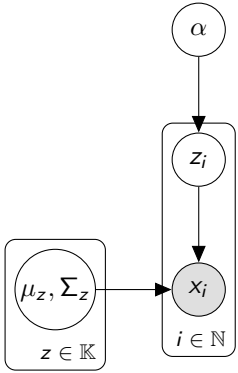
UNIVERSITY OF BATH

Aside: K-means

- K-means – alternate between:
  1. Assign points to nearest cluster centre – **E-step, without probability**
  2. Update cluster centres to mean of assigned points – **M-step**

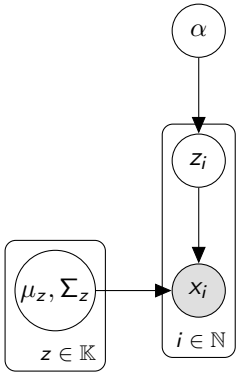
- EM is a generalisation of k-means optimisation
- EM handles probabilities (GMM is probabilistic k-means)

EM for GMM



- Model RVs: α, μ, Σ
- Missing RVs: z
- Observed RVs: x

EM for GMM



- Model RVs: α, μ, Σ
- Missing RVs: z
- Observed RVs: x

- Likelihood function of z_i:
  Categorical distribution, $z_i \sim \texttt{Cat}(w_i)$
  $\sum_{z=1}^{K} w_{iz} = 1$

Initialisation

- Any random parameters will do, but...

Initialisation

- Any random parameters will do, but...
- Much better:
  1. Run k-means with K centres
  2. Set $\alpha \propto$ number assigned to each centre
  3. Fit μ and Σ to points assigned to each centre
- Closer to goal $\therefore$ faster to converge
  (clever k-means initialisations also exist)

E-step

Update: (for all i, z)

$$w_{iz} \propto \alpha_z \mathcal{N}(x_i | \mu_z, \Sigma_z)$$

(normalise so $\sum_{z=1}^{K} w_{iz} = 1$)

- In English: Calculate probability of each data point being generated by each component
- Soft assignment, rather than hard assignment of k-means

M-step

Update: (for all z)

$$\alpha_z \propto T_z, T_z = \sum_{i=1}^{N} w_{iz}$$

(normalise so $\sum_{z=1}^{K} \alpha_z = 1$)

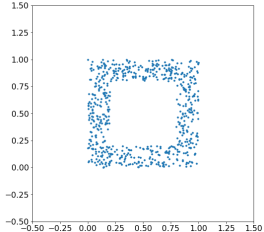
$$\mu_z = \frac{1}{T_z} \sum_{i=1}^{N} w_{iz} x_i, \qquad \Sigma_z = \frac{1}{T_z} \sum_{i=1}^{N} w_{iz} (x_i - \mu_z)(x_i - \mu_z)^T$$

- In English: Fit (maximise) weight + Gaussian of each component using weights / weighted mean / weighted covariance
- Soft update due to weights

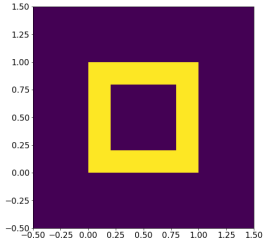
UNIVERSITY OF BATH

Demonstration

Input:



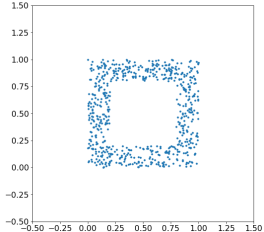
Ground Truth:



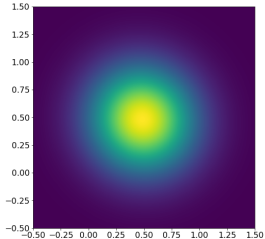
UNIVERSITY OF BATH

Demonstration

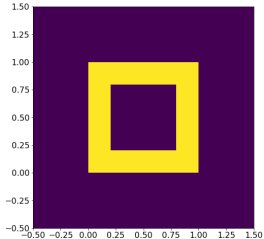
Input:



Gaussian:



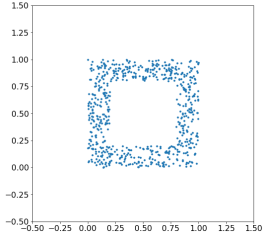
Ground Truth:



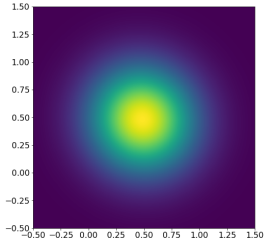
UNIVERSITY OF BATH

Demonstration

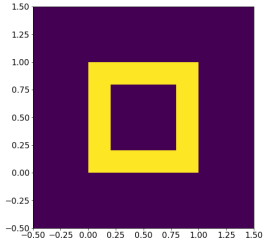
Input:



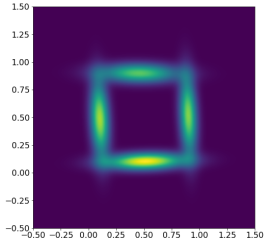
Gaussian:



Ground Truth:



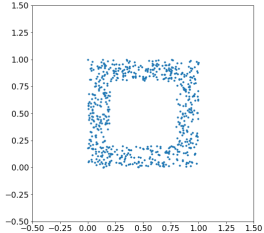
GMM with 4 components:



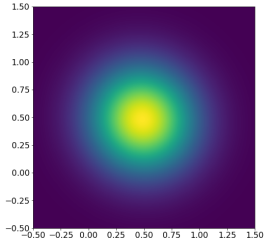
UNIVERSITY OF BATH

Demonstration

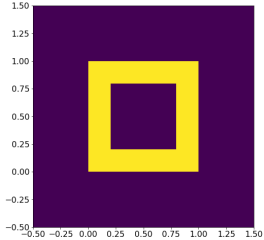
Input:



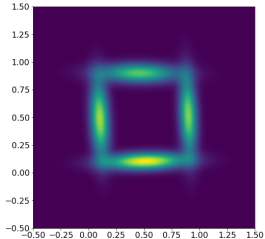
Gaussian:



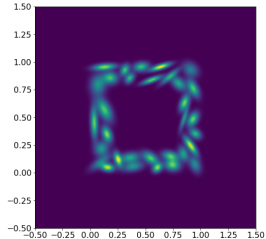
Ground Truth:



GMM with 4 components:



GMM with 32 components:



How many components?

- Need to choose K
- Hyperparameter optimisation:
  1. Sweep K, fitting model for each value
  2. Calculate probability of holdout set
  3. Choose model with maximum holdout probability (log space)

UNIVERSITY OF BATH

How many components?

- Need to choose K
- Hyperparameter optimisation:
  1. Sweep K, fitting model for each value
  2. Calculate probability of holdout set
  3. Choose model with maximum holdout probability (log space)
- Alternatively:
  1. Sweep K, fitting model for each value
  2. Calculate *Bayesian information criterion* (BIC)
  3. Choose model with lowest BIC

UNIVERSITY OF BATH

Bayesian information criterion

also called Schwarz information criterion

- Idea: Penalise models with lots of parameters

Bayesian information criterion

also called Schwarz information criterion

- Idea: Penalise models with lots of parameters
- Select model to minimise:

$$\ln(N)\mathrm{len}(\theta) - 2\ln(P(x|\theta))$$

- N – Data point count
- $\mathrm{len}(\theta)$ – Model parameter count
- $P(x|\theta)$ – Probability of data given optimal model parameters (θ)

UNIVERSITY OF BATH

Bayesian information criterion

also called Schwarz information criterion

- Idea: Penalise models with lots of parameters
- Select model to minimise:

$$\ln(N)\mathrm{len}(\theta) - 2\ln(P(x|\theta))$$

- N – Data point count
- $\mathrm{len}(\theta)$ – Model parameter count
- $P(x|\theta)$ – Probability of data given optimal model parameters (θ)

- BIC is crude – fails easily
- Many others, e.g. minimum description length (mostly just as crude)
- Non-parametric techniques better but hard, e.g. Dirichlet process prior

Summary

- Real problems have unknowns
- Plate notation

- Latent semantic analysis
- Gaussian mixture model

UNIVERSITY OF BATH

Further reading

- Paper describing the use of LSA on the Netflix challenge:
  "*Matrix Factorization Techniques for Recommender Systems*"
  by Koren, Bell & Volinsky (2009) (source of results images)

- Original EM paper, if you're so inclined:
  "*Maximum Likelihood from Incomplete Data via the EM Algorithm*",
  by Dempster, Laird, & Rubin (1977)
  (Special cases were discovered before, has mistake in convergence proof)

- If you sensibly decide to ignore the above:
  Chapter 9 of "*Pattern Recognition and Machine Learning*" by Bishop