ColumbiaX: Machine Learning
Lecture 23

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

Association analysis

SETUP

Many businesses have massive amounts of customer purchasing data.

- Amazon has your order history
- A grocery store knows objects purchased in each transaction
- Other retailers have data on purchases in their stores

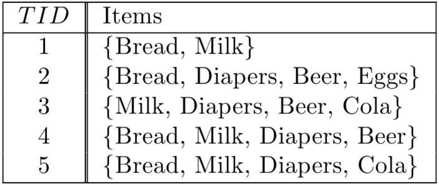
Using this data, we may want to find sub-groups of products that tend to co-occur in purchasing or viewing behavior.

- Retailers can use this to cross-promote products through “deals”
- Grocery stores can use this to strategically place items
- Online retailers can use this to recommend content
- This is more general than finding purchasing patterns

Market basket analysis

Association analysis is the task of understanding these patterns.

For example consider the following “market baskets” of five customers.



TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Using such data, we want to analyze patterns of co-occurance within it. We can use these patterns to define *association rules*. For example,

$$\{\text{diapers}\} \Rightarrow \{\text{beer}\}$$

ASSOCIATION ANALYSIS AND RULES

Imagine we have:

- p different objects indexed by $\{1, \dots, p\}$
- A collection of subsets of these objects $X_n \subset \{1, \dots, p\}$. Think of X_n as the index of things purchased by customer $n = 1, \dots, N$.

Association analysis: Find subsets of objects that often appear together. For example, if $\mathcal{K} \subset \{1, \dots, p\}$ indexes objects that frequently co-occur, then

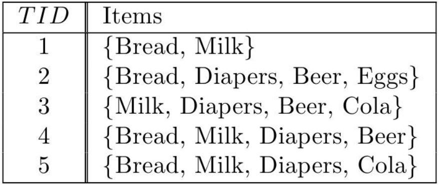
$$P(\mathcal{K}) = \frac{\#\{n \text{ such that } \mathcal{K} \subseteq X_n\}}{N} \quad \text{is large relatively speaking}$$

Example: $\mathcal{K} = \{$`peanut_butter, jelly, bread`$\}$

Association rules: Learn correlations. Let A and B be disjoint sets. Then $A \Rightarrow B$ means purchasing A increases likelihood of also purchasing B.

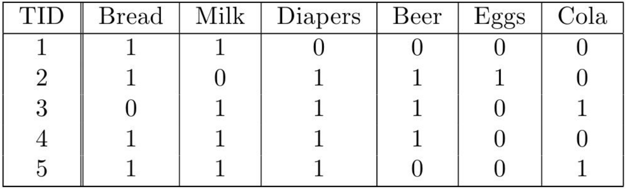
Example: $\{$`peanut_butter, jelly`$\} \Rightarrow \{$`bread`$\}$

PROCESSING THE BASKET



TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Figure: An example of 5 baskets.



TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Figure: A binary representation of these 5 baskets for analysis.

PROCESSING THE BASKET

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Want to find subsets that occur with probability above some threshold.

For example, does {`bread, milk`} occur relatively frequently?

- Go to each of the 5 baskets and count the number that contain both.
- Divide this number by 5 to get the frequency.
- Aside: Notice that the basket might have more items in it.

When $N = 5$ and $p = 6$ as in this case, we can easily check every possible combination. However, real problems might have $N \approx 10^8$ and $p \approx 10^4$.

SOME COMBINATORICS

Some combinatorial analysis will show that brute-force search isn't possible.

Q: How many different subsets $\mathcal{K} \subseteq \{1, \ldots, p\}$ are there?

A: Each subset can be represented by a binary indicator vector of length p. The total number of possible vectors is 2^p.

Q: Nobody will have a basket with every item in it, so we shouldn't check every combination. How about if we only check up to k items?

A: The number of sets of size k picked from p items is $\binom{p}{k} = \frac{p!}{k!(p-k)!}$. For example, if $p = 10^4$ and $k = 5$, then $\binom{p}{k} \approx 10^{18}$.

Takeaway: Though the problem only requires counting, we need an algorithm that can tell us which $\mathcal{K}$ we should count and which we can ignore.

QUANTITIES OF INTEREST

Before we find an efficient counting algorithm, what do we want to count?

- Again, let $\mathcal{K} \subset \{1, \ldots, p\}$ and $A, B \subset \mathcal{K}$, where $A \cup B = \mathcal{K}$, $A \cap B = \emptyset$.

We're interested in the following empirically-calculated probabilities:

1. $P(\mathcal{K}) = P(A, B)$: The *prevalence* (or support) of items in set $\mathcal{K}$. We want to find which combinations co-occur often.
2. $P(B|A) = \frac{P(\mathcal{K})}{P(A)}$: The *confidence* that B appears in the basket given A is in the basket. We use this to define a *rule* $A \Rightarrow B$.
3. $L(A, B) = \frac{P(A,B)}{P(A)P(B)} = \frac{P(B|A)}{P(B)}$: The *lift* of the rule $A \Rightarrow B$. This is a measure of how much *more* confident we are in B given that we see A.

EXAMPLE

For example, let

$$\mathcal{K} = \{\texttt{peanut\_butter, jelly, bread}\},$$

$$A = \{\texttt{peanut\_butter, jelly}\}, B = \{\texttt{bread}\}$$

- A *prevalence* of 0.03 means that `peanut_butter, jelly` and `bread` appeared together in 3% of baskets.

- A *confidence* of 0.82 means that when both `peanut_butter` and `jelly` were purchased, 82% of the time `bread` was also purchased.

- A *lift* of 1.95 means that it's 1.95 more probable that `bread` will be purchased given that `peanut_butter` and `jelly` were purchased.

Apriori algorithm

The goal of the **Apriori algorithm** is to quickly find all of the subsets $\mathcal{K} \subset \{1, \dots, p\}$ that have probability greater than a predefined threshold t.

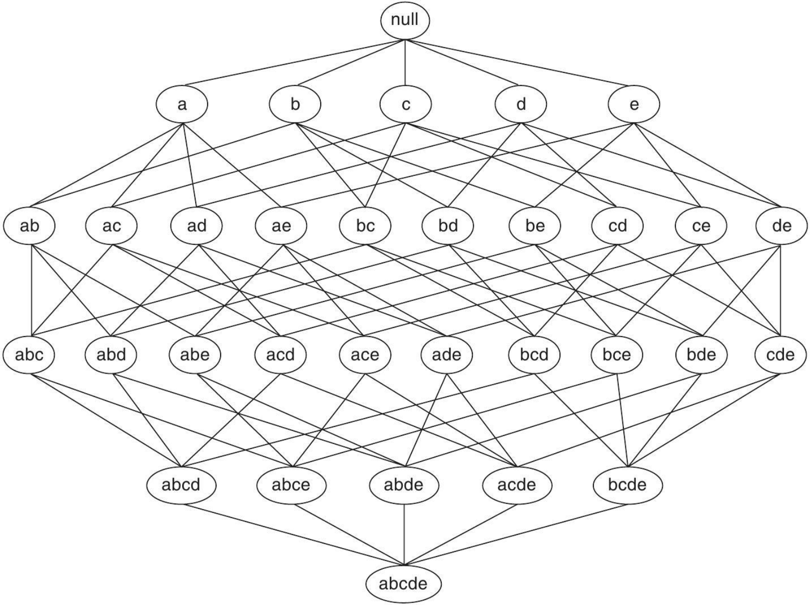
- Such a $\mathcal{K}$ will contain items that appear in at least $N \cdot t$ of the N baskets.
- A small fraction of such $\mathcal{K}$ should exist out of the 2^p possibilities.

Apriori uses properties about $P(\mathcal{K})$ to reduce the number of subsets that need to be checked to a small fraction of all 2^p sets.

- It starts with $\mathcal{K}$ containing 1 item. It then moves to 2 items, etc.
- Sets of size $k-1$ that "survive" help determine sets of size k to check.
- Important: Apriori finds *every* set $\mathcal{K}$ such that $P(\mathcal{K}) > t$.

Next slide: The structure of the problem can be organized in a lattice.

LATTICE REPRESENTATION



FREQUENCY DEPENDENCE

We can use two properties to develop an algorithm for efficiently counting.

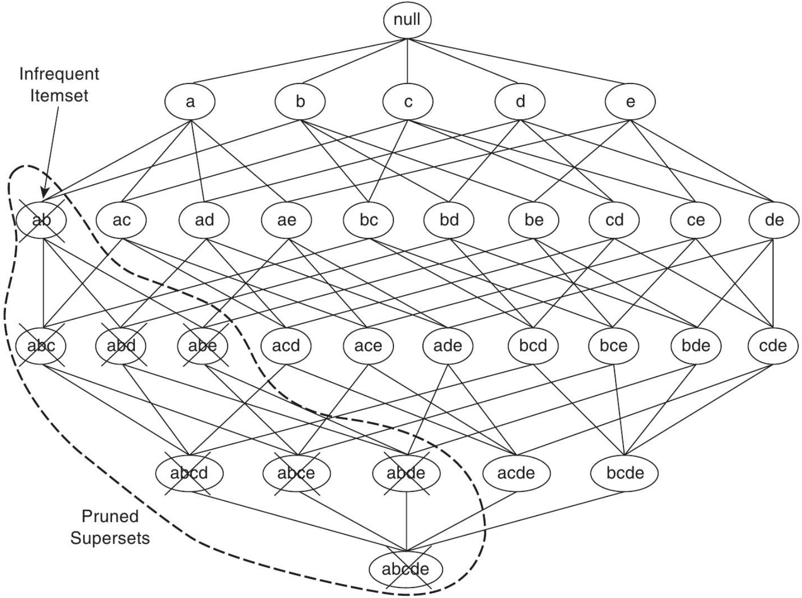
1. If the set $\mathcal{K}$ is not big enough, then $\mathcal{K}' = \mathcal{K} \cup A$ with $A \subset \{1, \ldots, p\}$ is not big enough. In other words: $P(\mathcal{K}) < t$ implies $P(\mathcal{K}') < t$

   e.g., Let $\mathcal{K} = \{a, b\}$. If these items appear together in x baskets, then the set of items $\mathcal{K}' = \{a, b, c\}$ appears in $\leq x$ baskets since $\mathcal{K} \subset \mathcal{K}'$.

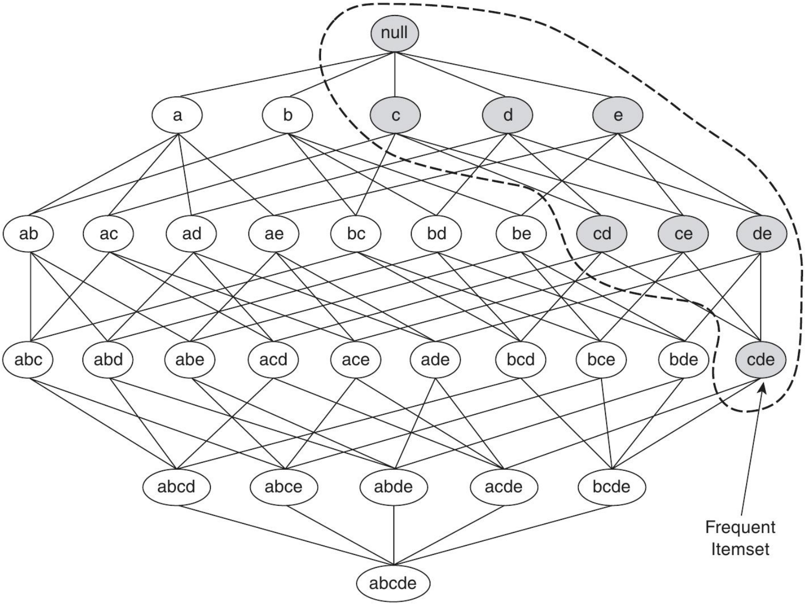
   Mathematically: $P(\mathcal{K}') = P(\mathcal{K}, A) = P(A|\mathcal{K})P(\mathcal{K}) \leq P(\mathcal{K}) < t$

2. By the converse, if $P(\mathcal{K}) > t$ and $A \subset \mathcal{K}$, then $P(A) > P(\mathcal{K}) > t$.

FREQUENCY DEPENDENCE: PROPERTY 1



FREQUENCY DEPENDENCE: PROPERTY 2



Apriori algorithm (one version)

Here is a basic version of the algorithm. It can be improved in clever ways.

Apriori algorithm

Set a threshold $N \cdot t$, where $0 < t < 1$ (but relatively small).

1. $|\mathcal{K}| = 1$: Check each object and keep those that appear in $\geq N \cdot t$ baskets.
2. $|\mathcal{K}| = 2$: Check all pairs of objects that survived Step 1 and keep the sets that appear in $\geq N \cdot t$ baskets.

⋮

k. $|\mathcal{K}| = k$: Using all sets of size $k - 1$ that appear in $\geq N \cdot t$ baskets,

- Increment each set with an object surviving Step 1 not already in the set.
- Keep all sets that appear in $\geq N \cdot t$ baskets

It should be clear that as k increases, we can hope that the number of sets that survive decrease. At a certain $k < p$, no sets will survive and we're done.

MORE CONSIDERATIONS

1. We can show that this algorithm returns *every* set $\mathcal{K}$ for which $P(\mathcal{K}) > t$.

- Imagine we know every set of size $k-1$ for which $P(\mathcal{K}) > t$. Then every potential set of size k that could have $P(\mathcal{K}) > t$ will be checked.

  e.g. Let $k = 3$: The set $\{a, b, c\}$ appears in $> N \cdot t$ baskets. Will we check it?
  Known: $\{a, b\}$ and $\{c\}$ must appear in $> N \cdot t$ baskets.
  Assumption: We've found $\mathcal{K} = \{a, b\}$ as a set satisfying $P(\mathcal{K}) > t$.
  Apriori algorithm: We know $P(\{c\}) > t$ and so will check $\{a, b\} \cup \{c\}$.
  Induction: We have all $|\mathcal{K}| = 1$ by brute-force search (start induction).

2. As written, this can lead to duplicate sets for checking, e.g., $\{a, b\} \cup \{c\}$ and $\{a, c\} \cup \{b\}$. Indexing methods can ensure we create $\{a, b, c\}$ once.

3. For each proposed $\mathcal{K}$, should we iterate through each basket for checking? There are tricks to make this faster that takes structure into account.

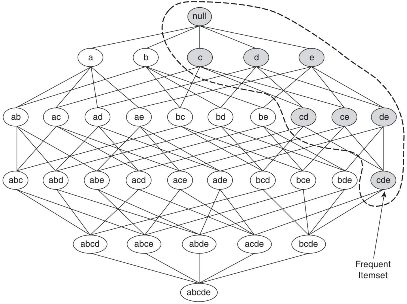
FINDING ASSOCIATION RULES

We've found all $\mathcal{K}$ such that

$$P(\mathcal{K}) > t.$$

Now we want to find association rules.

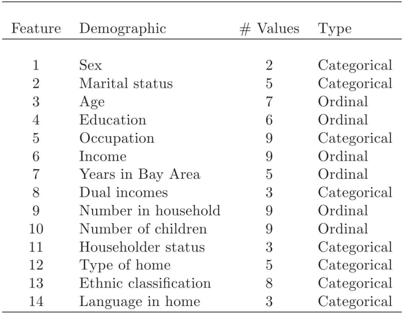
These are of the form $P(A|B) > t_2$ where we split $\mathcal{K}$ into subsets A and B.



Notice:

1. $P(A|B) = \frac{P(\mathcal{K})}{P(B)}$.
2. If $P(\mathcal{K}) > t$ and A and B partition $\mathcal{K}$, then $P(A) > t$ and $P(B) > t$.
3. Since Apriori found all $\mathcal{K}$ such that $P(\mathcal{K}) > t$, it found $P(A)$ and $P(B)$, so we can calculate $P(A|B)$ without counting again.

Example



Feature	Demographic	# Values	Type
1	Sex	2	Categorical
2	Marital status	5	Categorical
3	Age	7	Ordinal
4	Education	6	Ordinal
5	Occupation	9	Categorical
6	Income	9	Ordinal
7	Years in Bay Area	5	Ordinal
8	Dual incomes	3	Categorical
9	Number in household	9	Ordinal
10	Number of children	9	Ordinal
11	Householder status	3	Categorical
12	Type of home	5	Categorical
13	Ethnic classification	8	Categorical
14	Language in home	3	Categorical

Data

$N = 6876$ questionnaires

14 questions coded into $p = 50$ items

For example:

- ordinal (2 items): Pick the item based on value being $\lessgtr$ median
- categorical: item = category

  x categories → x items

- Based on the item encoding, it's clear that no "basket" can have every item.
- We see that association analysis extends to more than consumer analysis.

Example

Association rule 1: Support 13.4%, confidence 80.8%, and lift 2.13.

$$\left[\begin{array}{rcl} \text{language in home} & = & \textit{English} \\ \text{householder status} & = & \textit{own} \\ \text{occupation} & = & \{\textit{professional/managerial}\} \end{array}\right]$$

$$\Downarrow$$

$$\text{income} \geq \textit{\$40,000}$$

Association rule 2: Support 26.5%, confidence 82.8% and lift 2.15.

$$\left[\begin{array}{rcl} \text{language in home} & = & \textit{English} \\ \text{income} & < & \textit{\$40,000} \\ \text{marital status} & = & \textit{not married} \\ \text{number of children} & = & \textit{0} \end{array}\right]$$

$$\Downarrow$$

$$\text{education} \notin \{\textit{college graduate, graduate study}\}$$

ColumbiaX: Machine Learning
Lecture 24

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

MODEL SELECTION

Model Selection

The model selection problem

We've seen how often model parameters need to be set in advance and discussed how this can be done using using cross-validation.

Another type of model selection problem is learning model order.

Model order: The complexity of a class of models

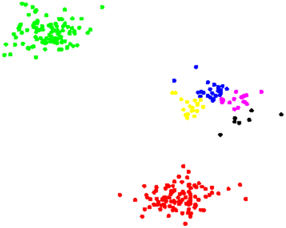
- Gaussian mixture model: How many Gaussians?
- Matrix factorization: What rank?
- Hidden Markov models: How many states?

In each of these problems, we can't simply look at the log-likelihood because a more complex model can always fit the data better.

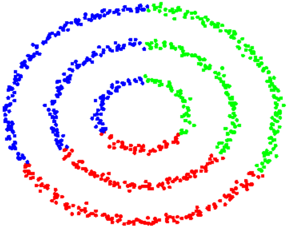
Model Selection

Model Order

We will discuss two methods for selecting an “appropriate” complexity of the model. This assumes a good model type was chosen to begin with.



(a) Inappropriate model order.



(b) Inappropriate model type.

Example: Maximum likelihood

Notation

We write $\mathcal{L}$ for the log-likelihood of a parameter under a model $p(x|\theta)$:

$$x_i \overset{iid}{\sim} p(x|\theta) \quad \Longleftrightarrow \quad \mathcal{L} = \sum_{i=1}^{N} \log p(x_i|\theta)$$

The maximum likelihood solution is: $\theta_{\text{ML}} = \arg\max_{\theta} \ \mathcal{L}$.

Example: How many clusters? (wrong way)

The parameters θ could be those of a GMM. We could find θ_{ML} for different numbers of clusters and pick the one with the largest $\mathcal{L}$.

Problem: We can perfectly fit the data by putting each observation in its own cluster. Then shrink the variance of each Gaussian to zero.

Number of Parameters

The general problem

- Models with more degrees of freedom are more prone to overfitting.
- The degrees of freedom is roughly the number of scalar parameters, K.
- By increasing K (done by increasing #clusters, rank, #states, etc.) the model can add more degrees of freedom.

Some common solutions

- **Stability**: Bootstrap sample the data, learn a model, calculate the likelihood on the original data set. Repeat and pick the best model.
- **Bayesian nonparametric methods**: Each possible value of K is assigned a prior probability. The posterior learns the best K.
- **Penalization approaches**: A penalty term makes adding parameters expensive. Must be overcome by a greater improvement in likelihood.

Penalizing model complexity

General form

Define a *penalty function* on the number of model parameters. Instead of maximizing $\mathcal{L}$, minimize $-\mathcal{L}$ and add the defined penalty.

Two popular penalties are:

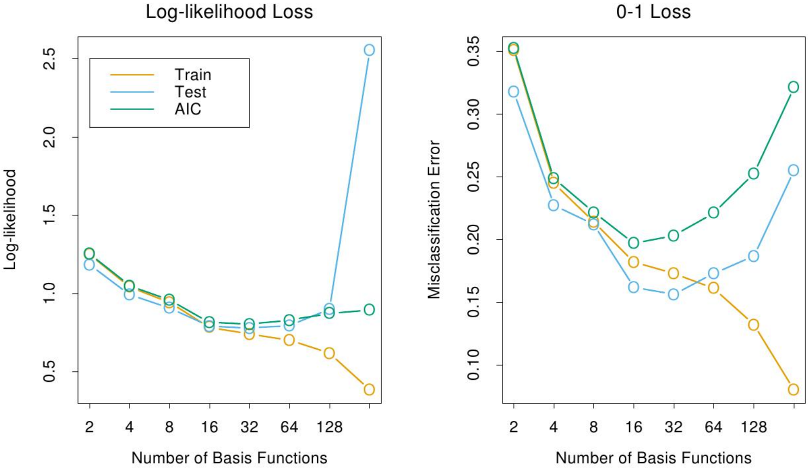
- **Akaike information criterion (AIC)**: $-\mathcal{L} + K$
- **Bayesian information criterion (BIC)**: $-\mathcal{L} + \frac{1}{2}K \ln N$

When $\frac{1}{2}\ln N > 1$, BIC encourages a simpler model (happens when $N \geq 8$).

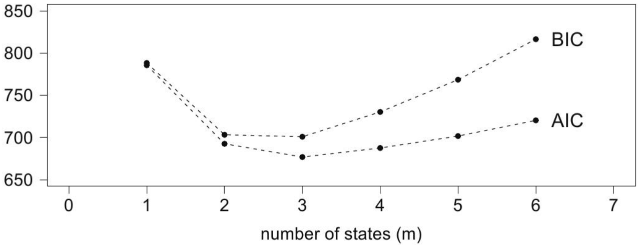
Example: For NMF with an $M_1 \times M_2$ matrix and rank R factorization,

$$\text{AIC} \;\rightarrow\; (M_1 + M_2)R, \qquad \text{BIC} \;\rightarrow\; \frac{1}{2}(M_1 + M_2)R\ln(M_1 M_2)$$

Example of AIC output



Example: AIC vs BIC on HMM



model	$-\log L$	AIC	BIC
'1-state HM'	391.9189	785.8	788.5
2-state HM	342.3183	692.6	703.3
3-state HM	329.4603	**676.9**	**701.0**
4-state HM	327.8316	687.7	730.4
5-state HM	325.9000	701.8	768.6
6-state HM	324.2270	720.5	816.7
indep. mixture (2)	360.3690	726.7	734.8
indep. mixture (3)	356.8489	723.7	737.1
indep. mixture (4)	356.7337	727.5	746.2

Notice:

- Likelihood is always improving
- Only compare location of AIC and BIC minima, not the values.

DERIVATION OF BIC

AIC AND BIC

Recall the two penalties:

- **Akaike information criterion (AIC)**: $-\mathcal{L} + K$
- **Bayesian information criterion (BIC)**: $-\mathcal{L} + \frac{1}{2} K \ln N$

Algorithmically, there is no extra work required:

1. Find the ML solution of the selected models and calculate $\mathcal{L}$.
2. Add the AIC or BIC penalty to get a score useful for picking a model.

Q: Where do these penalties come from? Currently they seem arbitrary.

A: We will derive BIC next. AIC also has a theoretical motivation, but we will not discuss that derivation.

DERIVING THE BIC

Imagine we have r candidate models, $\mathcal{M}_1, \dots, \mathcal{M}_r$. For example, r HMMs each having a different number of states.

We also have data $\mathcal{D} = \{x_1, \dots, x_N\}$. We want the posterior of each $\mathcal{M}_i$.

$$p(\mathcal{M}_i|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{M}_i)p(\mathcal{M}_i)}{\sum_j p(\mathcal{D}|\mathcal{M}_j)p(\mathcal{M}_j)}$$

If we assume a uniform prior distribution on models, then because the denominator is constant in $\mathcal{M}_i$, we can pick

$$\mathcal{M} = \arg\max_{\mathcal{M}_i} \ln p(\mathcal{D}|\mathcal{M}_i) = \int \ln p(\mathcal{D}|\theta, \mathcal{M}_i)p(\theta|\mathcal{M}_i)d\theta$$

We're choosing the model with the largest *marginal likelihood* of the data by integrating out all parameters of the model. This is usually not solvable.

DERIVING THE BIC

We will see how the BIC arises from the approximation,

$$\mathcal{M} = \arg\max_{\mathcal{M}_i}\ \ln p(\mathcal{D}|\mathcal{M}_i)\ \approx\ \arg\max_{\mathcal{M}_i}\ \ln p(\mathcal{D}|\theta_{\text{ML}}, \mathcal{M}_i) - \frac{1}{2}K\ln N$$

Step 1: Recognize that the difficulty is with the integral

$$\ln p(\mathcal{D}|\mathcal{M}_i)\ =\ \ln\int p(\mathcal{D}|\theta)p(\theta)d\theta.$$

$\mathcal{M}_i$ determines $p(\mathcal{D}|\theta)$, $p(\theta)$—we will suppress this conditioning.

Step 2: Approximate this integral using a second-order Taylor expansion.

Deriving the BIC

1. We want to calculate:

$$\ln p(\mathcal{D}|\mathcal{M}) \;=\; \ln \int p(\mathcal{D}|\theta)p(\theta)d\theta \;=\; \ln \int \exp\{\ln p(\mathcal{D}|\theta)\}p(\theta)d\theta$$

2. We use a second-order Taylor expansion of $\ln p(\mathcal{D}|\theta)$ at the point θ_{ML},

$$\begin{aligned}\ln p(\mathcal{D}|\theta) \quad &\approx \quad \ln p(\mathcal{D}|\theta_{\text{ML}}) + (\theta - \theta_{\text{ML}})^T \underbrace{\nabla \ln p(\mathcal{D}|\theta_{\text{ML}})}_{=\,0} \\ &\qquad + \frac{1}{2}(\theta - \theta_{\text{ML}})^T \underbrace{\nabla^2 \ln p(\mathcal{D}|\theta_{\text{ML}})}_{=\,-\mathcal{J}(\theta_{\text{ML}})}(\theta - \theta_{\text{ML}})\end{aligned}$$

3. Approximate $p(\theta)$ as uniform and plug this approximation back in,

$$\ln p(\mathcal{D}|\mathcal{M}) \approx \ln p(\mathcal{D}|\theta_{\text{ML}}) + \ln \int \exp\left\{-\frac{1}{2}(\theta-\theta_{\text{ML}})^T \mathcal{J}(\theta_{\text{ML}})(\theta-\theta_{\text{ML}})\right\} d\theta$$

DERIVING THE BIC

Observation: The integral is the normalizing constant of a Gaussian,

$$\int \exp\Big\{ -\frac{1}{2}(\theta - \theta_{\text{ML}})^T \mathcal{J}(\theta_{\text{ML}})(\theta - \theta_{\text{ML}})\Big\} d\theta \;=\; \left(\frac{2\pi}{|\mathcal{J}(\theta_{\text{ML}})|}\right)^{K/2}$$

Remember the definition that

$$-\mathcal{J}(\theta_{\text{ML}}) \;=\; \nabla^2 \ln p(\mathcal{D}|\theta_{\text{ML}}) \;\overset{(a)}{=}\; \underbrace{N \sum_{i=1}^{N} \frac{1}{N} \nabla^2 \ln p(x_i|\theta_{\text{ML}})}_{\text{converges as } N \text{ increases}}$$

(a) is by the i.i.d. model assumption made at the beginning of the lecture.

DERIVING THE BIC

4. Plugging this in,

$$\ln p(\mathcal{D}|\mathcal{M}) \;\approx\; \ln p(\mathcal{D}|\theta_{\text{ML}}) \;+\; \ln\left(\frac{2\pi}{|\mathcal{J}(\theta_{\text{ML}})|}\right)^{K/2}$$

and $|\mathcal{J}(\theta_{\text{ML}})| = N\left|\sum_{i=1}^{N}\frac{1}{N}\nabla^2 \ln p(x_i|\theta_{\text{ML}})\right|$.

Therefore we arrive at the BIC,

$$\ln p(\mathcal{D}|\mathcal{M}) \;\approx\; \ln p(\mathcal{D}|\theta_{\text{ML}}) \;-\; \frac{1}{2}K\ln N \;+\; \underbrace{\text{something not growing with } N}_{O(1) \text{ term, so we ignore it}}$$

SOME NEXT STEPS

ICML Sessions (subset)

The International Conference on Machine Learning (ICML) is a major ML conference. Many of the session titles should look familiar:

- Bayesian Optimization and Gaussian Processes
- PCA and Subspace Models
- Supervised Learning
- Matrix Completion and Graphs
- Clustering and Nonparametrics
- Active Learning
- Clustering
- Boosting and Ensemble Methods
- Matrix Factorization I & II
- Kernel Methods I & II
- Topic models
- Time Series and Sequences
- etc.

ICML Sessions (subset)

Other sessions might not look so familiar:

- Reinforcement Learning I & II
- Bandits I & II
- Optimization I, II & III
- Bayesian nonparametrics I & II
- Online learning I & II
- Graphical Models I & II
- Neural Networks and Deep Learning I & II
- Metric Learning and Feature Selection
- etc.

Many of these topics are taught in advanced machine learning courses at Columbia in the CS, Statistics, IEOR and EE departments.