ELEN 4903: Machine Learning
Lecture 24, 4/26/2016

Prof. John Paisley
Columbia University
The model selection problem
We’ve seen how some model parameters need to be set in advance and discussed how this can be done 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 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.
**Notation**
We write $\mathcal{L}$ for the log-likelihood of a parameter under a model $p(x|\theta)$:

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

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

**Example: How many clusters? (wrong way)**
The parameters $\theta$ could be those of a GMM. We could find $\theta_{\text{ML}}$ 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.
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 one.
- **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 $M \times N$ matrix and rank $R$ factorization,

\[
\text{AIC } \rightarrow (M + N)R, \quad \text{BIC } \rightarrow \frac{1}{2}(M + N)R \ln(MN)
\]
EXAMPLE OF AIC OUTPUT

Log-likelihood Loss

0-1 Loss

Number of Basis Functions

Number of Basis Functions

Log-likelihood

Misclassification Error

Train

Test

AIC
EXAMPLE: AIC vs BIC on HMM

Notice:

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

<table>
<thead>
<tr>
<th>model</th>
<th>$k$</th>
<th>$-\log L$</th>
<th>AIC</th>
<th>BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘1-state HM’</td>
<td>1</td>
<td>391.9189</td>
<td>785.8</td>
<td>788.5</td>
</tr>
<tr>
<td>2-state HM</td>
<td>4</td>
<td>342.3183</td>
<td>692.6</td>
<td>703.3</td>
</tr>
<tr>
<td>3-state HM</td>
<td>9</td>
<td>329.4603</td>
<td><strong>676.9</strong></td>
<td><strong>701.0</strong></td>
</tr>
<tr>
<td>4-state HM</td>
<td>16</td>
<td>327.8316</td>
<td>687.7</td>
<td>730.4</td>
</tr>
<tr>
<td>5-state HM</td>
<td>25</td>
<td>325.9000</td>
<td>701.8</td>
<td>768.6</td>
</tr>
<tr>
<td>6-state HM</td>
<td>36</td>
<td>324.2270</td>
<td>720.5</td>
<td>816.7</td>
</tr>
<tr>
<td>indep. mixture (2)</td>
<td>3</td>
<td>360.3690</td>
<td>726.7</td>
<td>734.8</td>
</tr>
<tr>
<td>indep. mixture (3)</td>
<td>5</td>
<td>356.8489</td>
<td>723.7</td>
<td>737.1</td>
</tr>
<tr>
<td>indep. mixture (4)</td>
<td>7</td>
<td>356.7337</td>
<td>727.5</td>
<td>746.2</td>
</tr>
</tbody>
</table>
DERIVATION OF 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 calculated \(\mathcal{L}\).
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 skip that derivation.
Imagine we have \( r \) candidate models, \( \mathcal{M}_1, \ldots, \mathcal{M}_r \). For example, \( r \) HMMs each having a different number of states.

We also have data \( \mathcal{D} = \{x_1, \ldots, x_N\} \). We want the posterior on each \( \mathcal{M} \).

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

If we assume a uniform prior distribution on models, then because the denominator doesn’t depend on a model, we pick

\[
\mathcal{M} = \arg \max_{\mathcal{M}_i} \ln p(\mathcal{D}|\mathcal{M}_i)
\]

We’re picking the model with the largest *marginal likelihood*. That is, we integrate out all parameters of the model. This is difficult in general.
**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 what \(p(\mathcal{D}|\theta)\) and \(p(\theta)\) are.)

**Step 2:** Approximate this integral using 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 \(\theta_{ML}\),

\[
\ln p(\mathcal{D}|\theta) \approx \ln p(\mathcal{D}|\theta_{ML}) + (\theta - \theta_{ML})^T \nabla \ln p(\mathcal{D}|\theta_{ML}) + \frac{1}{2} (\theta - \theta_{ML})^T \nabla^2 \ln p(\mathcal{D}|\theta_{ML})(\theta - \theta_{ML}) = 0
\]

\[
= -\mathcal{J}(\theta_{ML})
\]

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

\[
\ln p(\mathcal{D}|\mathcal{M}) \approx \ln p(\mathcal{D}|\theta_{ML}) + \ln \int \exp \left\{ -\frac{1}{2} (\theta - \theta_{ML})^T \mathcal{J}(\theta_{ML})(\theta - \theta_{ML}) \right\}d\theta
\]
**Observation:** The integral is the normalizing constant of a Gaussian,

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

Remember the definition that

\[-\mathcal{J}(\theta_{ML}) = \nabla^2 \ln p(\mathcal{D}|\theta_{ML}) = N \sum_{i=1}^{N} \frac{1}{N} \nabla^2 \ln p(x_i|\theta_{ML})\]

converges as \(N\) increases

4. Therefore we arrive at the BIC,

\[\ln p(\mathcal{D}|\mathcal{M}) \approx \ln p(\mathcal{D}|\theta_{ML}) - \frac{1}{2} K \ln N + \text{something not growing with } N\]

\(O(1)\) term, so we ignore it
SOME NEXT STEPS
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.