UNIVERSITY OF MARYLAND

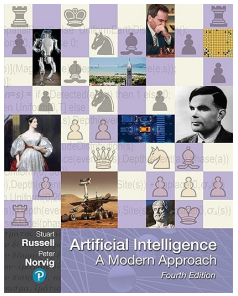
MSML610: Advanced Machine Learning

7.3: Hierarchical Models

Instructor: Dr. GP Saggese - gsaggese@umd.edu

References:

- AIMA (Artificial Intelligence: a Modern Approach)
  - Chap 15: Probabilistic programming
- Martin, Bayesian Analysis with Python, 2018 (2e)



- ***Hierarchical Models***

Hierarchical Models

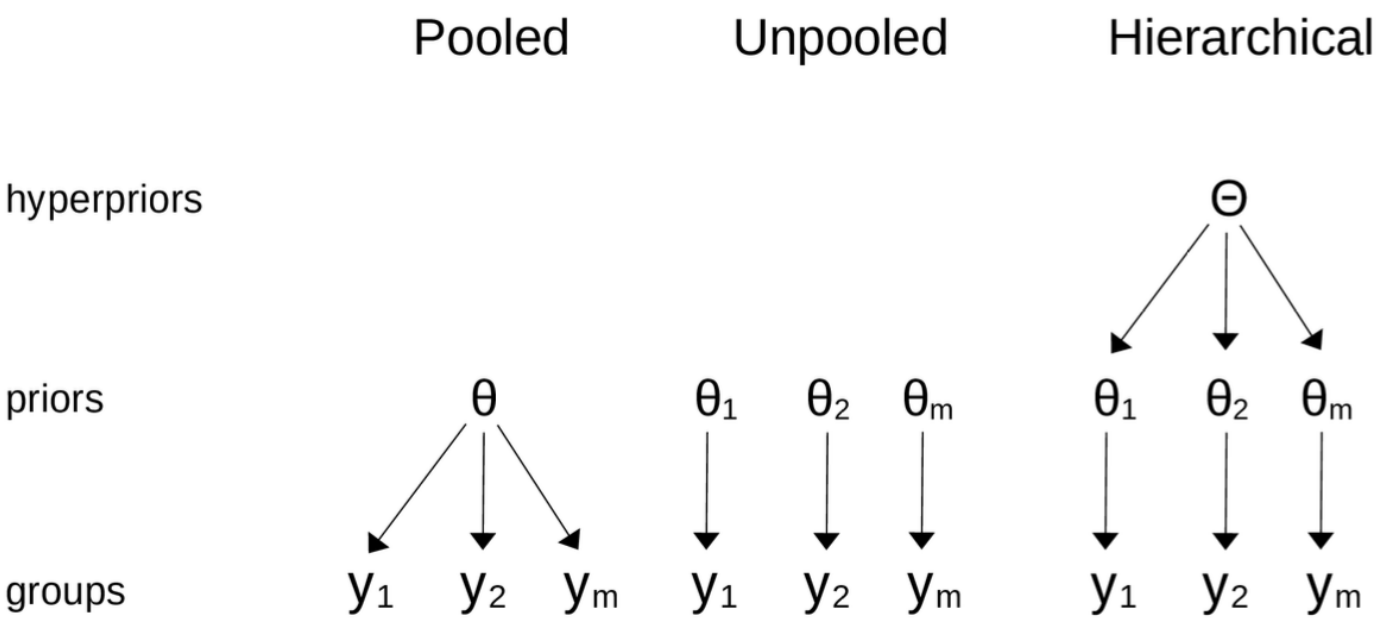
- Aka "multilevel", "nested", "mixed-effects" models
- **Key observation**: data points share structure, but also have variations
  - **Group data**
    - E.g., sales in cities: each city is a market, with common trends
  - **Hierarchical structure**
    - E.g., students in a school: each student is different, with common factors
  - **Repeated measurements on same objects**
- **Idea**: 💡
  - Model shares information between groups, but allows differences
  - Parameters of prior distributions have a prior distribution
    - Aka "hyper-priors" (!)
- You can't do this with frequentist approach, only Bayesian approach

Hierarchical Models: Examples

- Many data problems lend themselves to **hierarchical descriptions**
- E.g.,
  - Medical research:
    - Estimate drug effectiveness
    - Categorize patients by demographics, disease severity
    - Estimate cure probability for subgroups
  - Market research
    - Understand consumer purchasing behavior
    - Categorize consumers by age, gender, income, education

Unpooled, Pooled, Hierarchical Models

- **Pooled**
  - Groups have the same priors
- **Unpooled**
  - Groups have different priors
- **Hierarchical**
  - Groups have different priors which come from a common prior



Hierarchical Models: Chemical Shift

- **Proteins** are made of 20 amino acids
  - Study proteins with nuclear magnetic resonance
  - Measure "chemical shift"
- **Data** looks like:

	ID	aa	theo	exp	diff
0	1BM8	ILE	61.18	58.27	2.91
1	1BM8	TYR	56.95	56.18	0.77
2	1BM8	SER	56.35	56.84	-0.49
3	1BM8	ALA	51.96	51.01	0.95
4	1BM8	ARG	56.54	54.64	1.90
...	...	...	...	...	...
1771	1KS9	LYS	55.79	57.51	-1.72
1772	1KS9	ARG	58.91	59.02	-0.11
1773	1KS9	LYS	59.49	58.92	0.57
1774	1KS9	GLU	59.48	58.36	1.12
1775	1KS9	SER	58.07	60.55	-2.48

1776 rows × 5 columns

- `ID`: Code of the protein
- `aa`: Name of the amino acid
- `theo`: Theoretical values of chemical shift
- `exp`: Experimental value
- `diff`: Difference between theoretical and experimental value

Hierarchical Models: Chemical Shift

- Given experimental measures of chemical shifts vs theoretical values, evaluate model using different styles
  1. **Pooled**
     - Compute difference between estimates and measures, fit Gaussian
     - More accurate estimates / lose amino acid info
  2. **Unpooled**
     - Fit 20 Gaussians for 20 amino acids
     - Detailed analysis / less accuracy
  3. **Hierarchical**
     - Model groups assuming common population

Chemical Shift: Unpooled Model

- **Model each group independently**
  - Use same model structure to compare groups

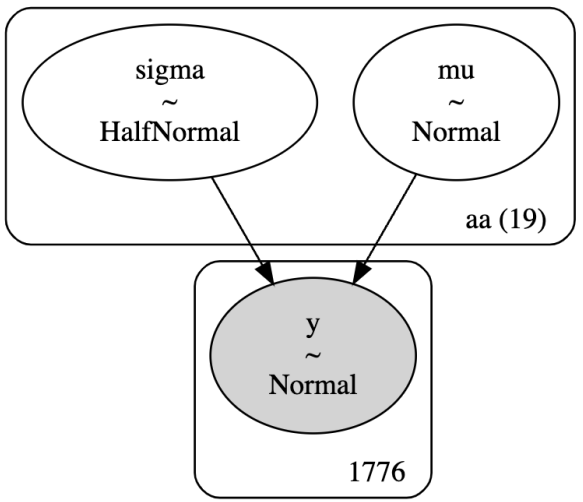
In [9]:
```
# Non-hierarchical model.
with pm.Model(coords=coords) as cs_nh:
    # One separate prior for each group.
    mu = pm.Normal('mu', mu=0, sigma=10, dims="aa")
    sigma = pm.HalfNormal("sigma", sigma=10, dims="aa")
    # Likelihood.
    y = pm.Normal("y", mu=mu[idx], sigma=sigma[idx], observed=diff)
    idata_cs_nh = pm.sample()
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 4 jobs)
NUTS: [mu, sigma]
Output()
```

```
Sampling 4 chains for 1_000 tune and 1_000 draw iterations (4_000 + 4_
```

In [10]:
```
pm.model_to_graphviz(cs_nh)
```

Out[10]:



Chemical Shift: Hierarchical Model

- **Add two hyperpriors on μ**
  - Mean of μ
  - Standard deviation of μ
- **Assume same variance σ for all groups**
  - Modeling choice
  - Option to add hyperpriors for σ
- Intermediate situation between single group and 20 separate groups

In [12]:

```
with pm.Model(coords=coords) as cs_h:
    # Hyper-priors.
    mu_mu = pm.Normal("mu_mu", mu=0, sigma=10)
    mu_sigma = pm.HalfNormal("mu_sigma", sigma=10)

    # Priors.
    mu = pm.Normal("mu", mu=mu_mu, sigma=mu_sigma, dims="aa")
    sigma = pm.HalfNormal("sigma", sigma=10, dims="aa")

    # Likelihood (same as before).
    y = pm.Normal("y", mu=mu[idx], sigma=sigma[idx], observed=diff)
    idata_cs_h = pm.sample()
```

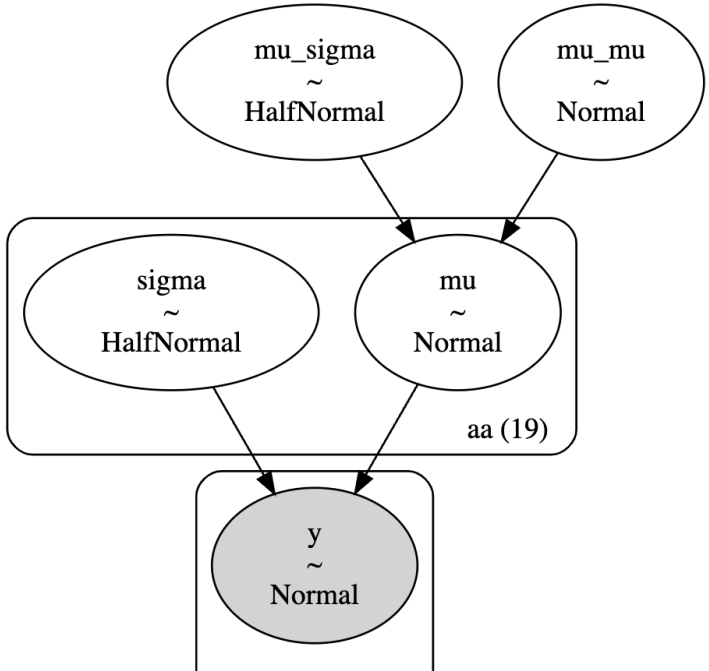
```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 4 jobs)
NUTS: [mu_mu, mu_sigma, mu, sigma]
Output()
```

```
Sampling 4 chains for 1_000 tune and 1_000 draw iterations (4_000 + 4_
```

In [13]:

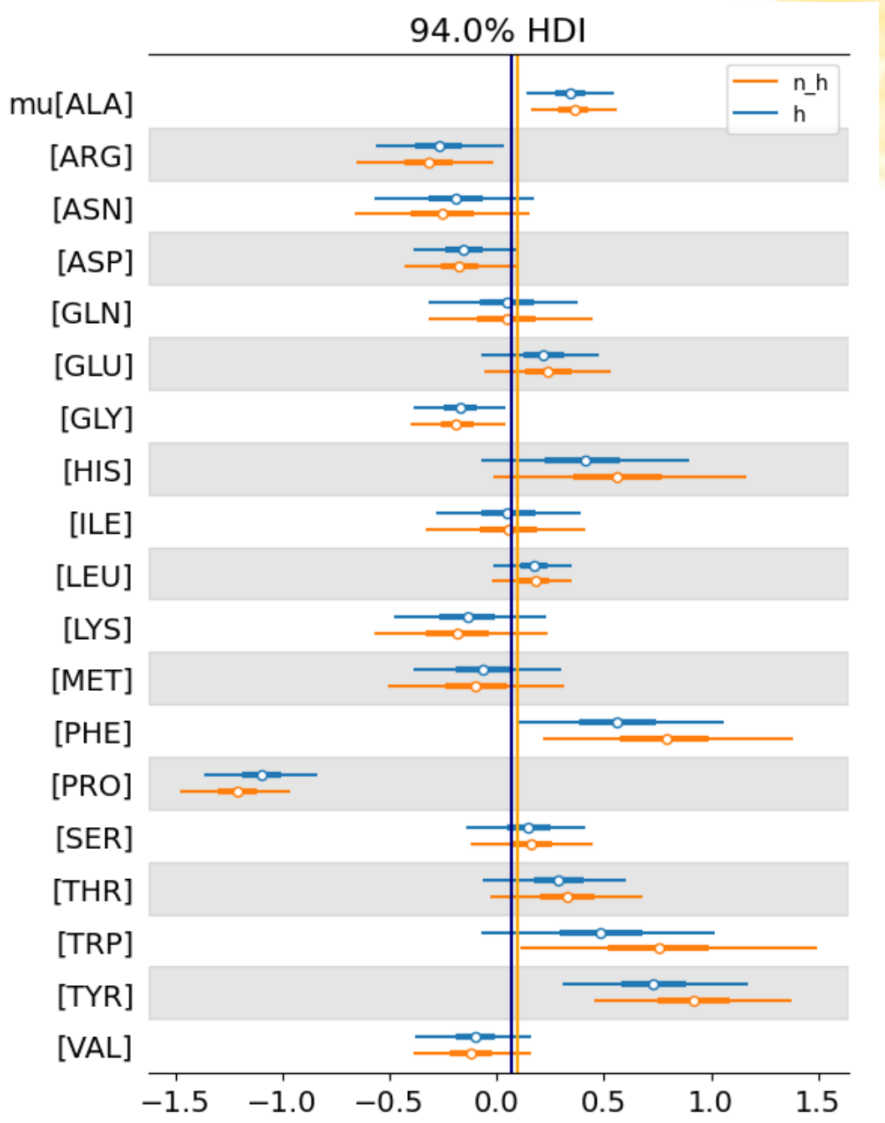
```
pm.model_to_graphviz(cs_h)
```

Out[13]:



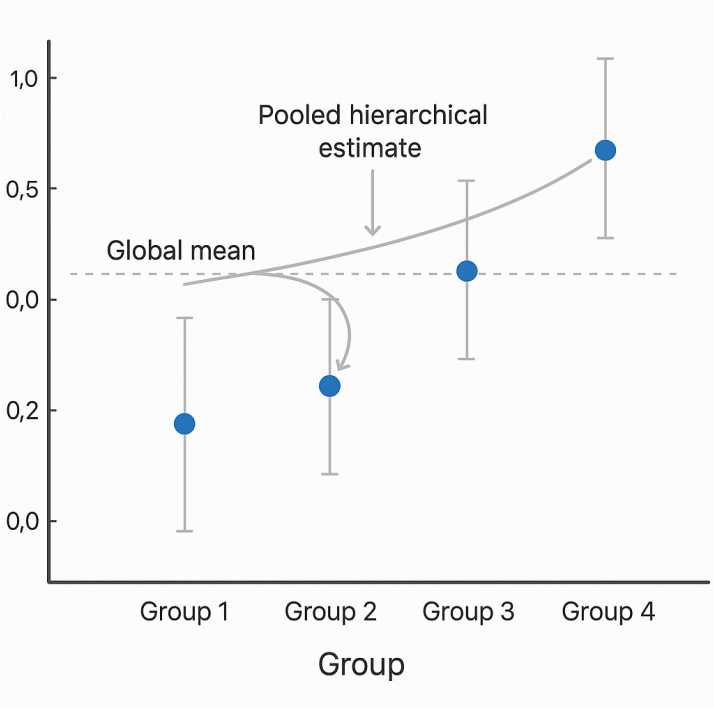
Chemical Shift: Results

- **Compare estimates of two models**
  - 20 groups
  - Each model has 4 estimated variables
- **\textcolor{gray}{Plot 94% credible intervals}**
  - **Blue**: hierarchical
  - **Orange**: non-hierarchical
  - **Black** vertical line: global mean (hierarchical model)
  - **Blue** means pulled towards mean compared to **orange**
- **Shrinkage occurs**



Shrinkage

- **Hierarchical models shrink parameters towards a common mean**
  - Groups share information through the hyper-prior
  - Model groups as neither independent nor a single group
  - Less responsive to extreme values in individual groups
  - Improve estimation for small groups using data from others



- **Amount of shrinkage** depends on data:
  - Groups with more data influence estimates more
  - Similar groups reinforce common estimation
  - Global optimization
- **Result**: inference is more stable

You Need to Know When to Stop

- You can create **hierarchical models with as many levels** as you want
  - **Pros**:
    - Leverage data structure
  - **Cons**
    - Don't improve inference quality
    - Complicate interpretation
- *"Add as many degrees as freedom as needed, but not more than what is warranted"* (Occam's razor)

Tutorial

- Hierarchical Models