Aster Models Short Course

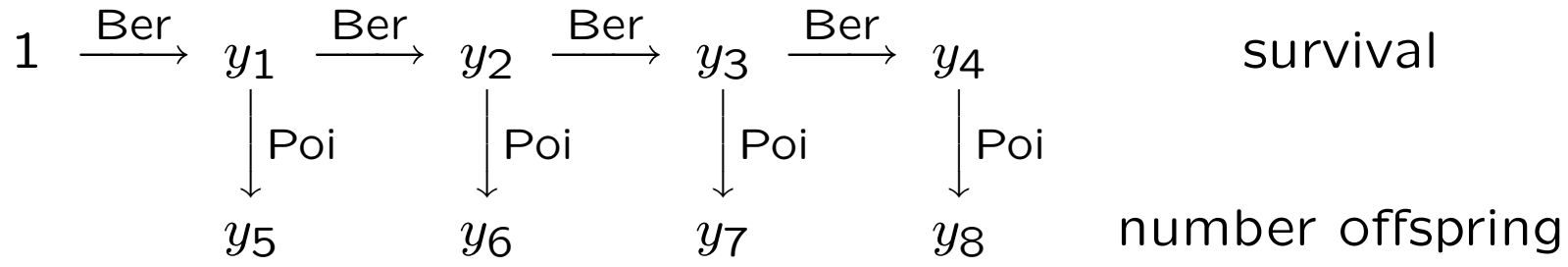
Charles J. Geyer
School of Statistics
University of Minnesota

Ruth G. Shaw
Department of Ecology, Evolution, and Behavior
University of Minnesota

http://www.stat.umn.edu/geyer/aster/

Why Aster Models?

Complex data require it.



y_j are components of data for one individual

Arrows indicate conditional distributions (Ber = Bernoulli and Poi = Poisson)

Each Bernoulli (zero-or-one-valued variable) is survival in one year. Each Poisson is number of offspring in that year.

What Else Could You Do?

Could analyze each component separately, but

- software doesn't support combining the results of separate analyses and

- difficult or impossible to address questions that involve all the data.

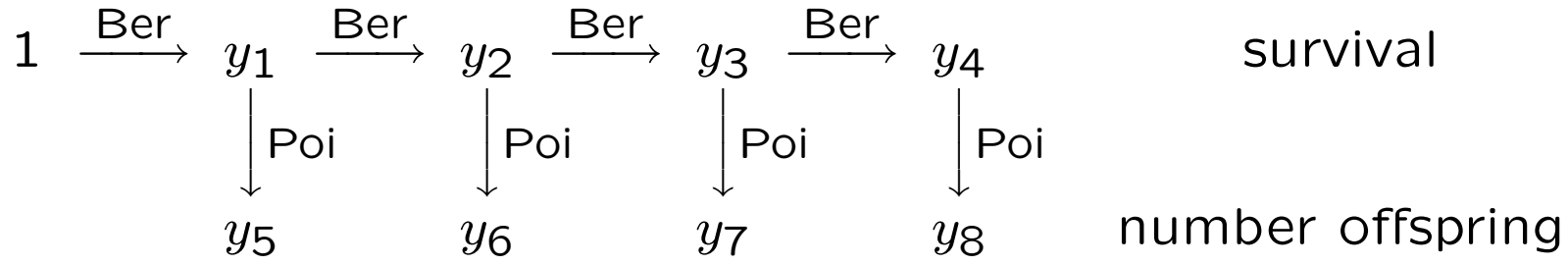
What Else Could You Do? (Cont.)

Could analyze a numerical function of the data (e. g., fitness), but

- such functions don't have simple distribution analyzable by known methodology so
- must proceed without a statistical model, and
- statistical inference weak or invalid.

What do Aster Models Do?

Aster models fit a statistical model for the joint distribution of dependent data like that shown in the graph we saw before repeated here



Valid statistical inference possible and easy.

Why are Aster Models Hard to Understand?

Like simpler forms of statistical inference in some ways but different in others.

We don't teach the general ideas of statistical inference well or at all in lower level courses.

Statistical Model Hierarchy

- linear models (multiple regression and ANOVA)
  - responses are independent from normal distribution
  - means are linear function of regression coefficients
- generalized linear models (logistic and Poisson regression)
  - responses are independent from same distribution
  - means are monotone function of regression coefficients
- aster models (life history analysis)
  - responses are dependent from different distributions
  - means are monotone function of regression coefficients

Regression

In regression models we divide the data into two parts the response vector $\mathbf{y}$ and the predictor data $\mathbf{x}$, which usually comprises one or more vectors.

The response $\mathbf{y}$ is considered random, the predictor $\mathbf{x}$ is considered non-random. If it is actually random, we say we are conditioning on it.

The objective: model the conditional distribution of $\mathbf{y}$ given $\mathbf{x}$.

Note: no assumptions whatsoever yet. Regression includes generalized linear models and aster models. Why do they call it logistic regression? Because it is a form of regression modeling!

Linear Models

In linear models the response $\mathbf{y}$ is a vector. Its components are stochastically independent. They are assumed normally distributed with the same variance σ^2. They have different means, which are components of a vector $\boldsymbol{\mu}$.

The linear part is that we assume $\boldsymbol{\mu}$ is a linear function of another parameter $\boldsymbol{\beta}$ of smaller dimension

$$\boldsymbol{\mu} = \mathbf{M}\boldsymbol{\beta}$$

where $\mathbf{M}$ is a non-random matrix, called the *model matrix*. Usually, $\mathbf{M}$ depends on the predictor data $\mathbf{x}$, the form of this dependence is arbitrary.

It's called linear regression because $\boldsymbol{\mu}$ is a linear function of $\boldsymbol{\beta}$, not because $\boldsymbol{\mu}$ is a linear function of $\mathbf{x}$ (it may be an arbitrary function of $\mathbf{x}$).

Linear Models (Cont.)

Some people like to write

$$\mathbf{y} = \mathbf{M}\boldsymbol{\beta} + \mathbf{e}$$

where $\mathbf{e}$ "error" is a vector of independent and identically distributed mean zero normal random variables.

But this does not generalize even to generalized linear models, much less aster models. Forget it. The important equation is $\boldsymbol{\mu} = \mathbf{M}\boldsymbol{\beta}$.

Data Structure

The structure of $\mathbf{y}$ we have already discussed. The structure of $\mathbf{x}$ is in principle arbitrary but in practice is one or more numerical or categorical vectors of the same dimension as $\mathbf{y}$. The `lm` function in R which fits linear models requires this.

The R terminology for a categorical variable is `factor`.

Simple Linear Regression

Here we have only one column of predictor data. Say y_i are the components of $\mathbf{y}$ and x_i the components of $\mathbf{x}$ and

$$\mu_i = \beta_1 + x_i \beta_2$$

How does that fit into linear models framework?

$$\mathbf{M} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$$

A linear model has an "intercept" (here β_1) when the model matrix has a column that is all ones.

Simple Linear Regression in R

Supposing we already have R objects `x` and `y` which are numerical vectors of the same length

```
out <- lm(y ~ x)
summary(out)
```

does the regression and reports regression coefficients, standard errors, and P-values;

```
plot(x, y)
abline(out)
```

draws the scatterplot with regression line;

```
plot(out)
```

does four diagnostic plots.

Linear Regression in R: Intercept

R automatically includes an intercept. If you don't want an intercept you have to indicate that in the formula provided to the `lm` function.

Instead of `y ~ x` which implies an intercept, either of

```
out <- lm(y ~ x + 0)
out <- lm(y ~ x - 1)
```

indicates the intercept should be omitted so the model is

$$\mu_i = x_i \beta_1$$

Multiple Linear Regression in R

If we have R objects `y`, `x1`, `x2`, and `x3`, all of which are numerical vectors of the same dimension, then

```
out <- lm(y ~ x1 + x2 + x3)
summary(out)
```

fits the model

$$\mu_i = \beta_1 + x_{i1}\beta_2 + x_{i2}\beta_3 + x_{i3}\beta_4$$

where x_{ik} are the components of x_k. Model matrix is

$$\mathbf{M} = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n3} \end{pmatrix}$$

coefficients of betas are columns of $\mathbf{M}$.

Multiple Linear Regression in R: Polynomials

```
out <- lm(y ~ x + I(x^2) + I(x^3))
```

fits the model

$$\mu_i = \beta_1 + x_i\beta_2 + x_i^2\beta_3 + x_i^3\beta_4$$

with model matrix

$$\mathbf{M} = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{pmatrix}$$

Multiple Linear Regression in R: Polynomials (Cont.)

```
out <- lm(y ~ x1 + x2 + I(x1^2) + I(x1 * x2) + I(x2^2))
```

fits the model

$$\mu_i = \beta_1 + x_{i1}\beta_2 + x_{i2}^2\beta_3 + x_{i1}^2\beta_4 + x_{i1}x_{i2}\beta_5 + x_{i2}^2\beta_6$$

with model matrix

$$\mathbf{M} = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{11}^2 & x_{11}x_{12} & x_{12}^2 \\ 1 & x_{21} & x_{22} & x_{21}^2 & x_{21}x_{22} & x_{22}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n1}^2 & x_{n1}x_{n2} & x_{n2}^2 \end{pmatrix}$$

Multiple Linear Regression in R: Polynomials (Cont.)

```
out <- lm(y ~ x1 + x2 + I(x1^2) + I(x1 * x2) + I(x2^2))
out <- lm(y ~ poly(x1, x2, degree = 2, raw = TRUE))
```

fit the same model in the same parameterization.

```
out <- lm(y ~ poly(x1, x2, degree = 2))
```

fits the same model in a different parameterization.

Dummy Variables

A "dummy variable" is a predictor variable that is zero-or-one-valued. Each categorical predictor is replaced by a set of dummy variables, one for each category. The dummy variable is one when the case is in that category and zero otherwise.

Because

$$\begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

must drop one dummy variable from set if there is intercept.

Dummy Variables (Cont.)

Must drop one dummy variable from each set (for each categorical predictor) if there is intercept in model.

Must drop one dummy variable from each set (for each categorical predictor) except for one of them if there is no intercept in model.

Dummy Variables (Cont.)

R is smart enough to do the right thing. If `fred` is a categorical predictor variable, then

```
out <- lm(y ~ fred)
out <- lm(y ~ fred + 0)
```

fit the same model in different parameterizations

$$\mu_i = \beta_1 + d_{i1}\beta_2 + d_{i2}\beta_3$$
$$\mu_i = d_{i1}\beta_1 + d_{i2}\beta_2 + d_{i3}\beta_3$$

where d_{ik} is the dummy variable for the k-th category (assuming 3 categories).

Tests of Statistical Hypotheses

Hypothesis tests involve two nested models, called the big model and the little model or the submodel and the supermodel. It is important that you know the models are nested — the little model is obtained by fixing the values of some parameter(s) in the big model — because *R does not check nestedness*.

If you have two result objects, say

```
out.little <- lm(y ~ x1 + x2)
out.big <- lm(y ~ x1 + x2 + I(x1^2) + I(x1 * x2) + I(x2^2))
```

then

```
anova(out.little, out.big)
```

will do the test reporting the value of the F statistic and the P-value.

Tests of Statistical Hypotheses (Cont.)

Can do tests about single regression coefficients using P-values reported by the `summary` function. Not recommended unless you interpret only one P-value per model fit. It can happen that

- none of the P-values for single regression coefficients appear significant although the correct model comparison test (done by the `anova` function) says they are jointly significant, or

- some of the P-values for single regression coefficients appear significant although the correct model comparison test (done by the `anova` function) says they are not jointly significant.

In either case the model comparison test is right and doing multiple single tests without correction for multiple testing is wrong and foolish.

Prediction

Often the estimated mean value vector $\hat{\mu} = \mathrm{M}\hat{\beta}$ is called the vector of "predicted values" and many textbooks denote it $\hat{\mathrm{y}}$ rather than $\hat{\mu}$ even though it is an estimate of a parameter vector. Moreover, the R function that produces it is called `predict`.

These things really shouldn't be called "predictions" because they are just parameter estimates, but we won't introduce a new name. We will however understand that they are just parameter estimates. Like any parameter estimates they have standard errors and the `predict` function produces them too, if asked

```
predict(out, interval = "confidence")
```

A Digression on Data Frames

In R a matrix must have all values of the same type (all `numeric`, all `logical`, or all `character`)

But R also has an object class called `data.frame` which looks like a matrix but different columns are allowed to have different types, in particular, some can be `numeric` and some `factor`.

Data Frames (Cont.)

You can read in a data frame from a file which must be a plain text file (not produced by a "word processor") having variable names as column headings

```
fred <- read.table("fred.txt", header = TRUE)
```

The file name must be quoted so R doesn't think it is an R variable name.

Data Frames (Cont.)

If the file had three variables `y`, `x1`, and `x2`, then we could do a regression two ways

```
out <- lm(y ~ x1 + x2, data = fred)
```

and the variables will be found in the data frame specified by the optional argument `data = fred`. We can also do

```
attach(fred)
out <- lm(y ~ x1 + x2)
```

but this is not recommended and doesn't generalize to aster models.

Data Frames (Cont.)

If the file `fred.txt` also contained a variable `sex` coded `M` and `F`, then the `read.table` function would automatically make `sex` a `factor` because it cannot be `numeric` and only `numeric` and `factor` variables make sense in a regression context. So

```
fred <- read.table("fred.txt", header = TRUE)
out <- lm(y ~ x1 + x2 + sex)
```

will do the right thing.

Data Frames (Cont.)

If, however, `sex` was coded `1` and `2`, then the `read.table` function could not know that `sex` was supposed to be a factor (the computer can't read your mind) and an additional step is necessary

```
fred <- read.table("fred.txt", header = TRUE)
fred$sex <- as.factor(fred$sex)
out <- lm(y ~ x1 + x2 + sex)
```

To check what the types of the variables in a data frame are do

```
sapply(fred, class)
```

Data Frames (Cont.)

If one has used Microsoft Excel as a data entry tool, then one should

- write out the file in CSV (comma separated values) format,
- remove all but the header line and the data lines,
- and read into R using the `read.csv` function

```
fred <- read.csv("fred.csv")
```

Prediction (Cont.)

Remember prediction? We often want to "predict" at data values different from the ones in the observed data.

This is equivalent to using a different model matrix. We are estimating $\hat{\boldsymbol{\mu}}_{\text{new}} = \mathbf{M}_{\text{new}}\hat{\boldsymbol{\beta}}$ rather than $\hat{\boldsymbol{\mu}} = \mathbf{M}\hat{\boldsymbol{\beta}}$.

Of course, when we are using the R formula mini-language, we don't specify model matrices explicitly, just formulas and data. So we use the same formula with new data, which we specify as a `data.frame` having all the predictor variables involved in the formula (the response is not necessary).

Prediction (Cont.)

```
out <- lm(y ~ x1 + x2 + I(x1^2) + I(x1 * x2) + I(x2^2)
predict(out, newdata = data.frame(x1 = 4.3, x2 = 6.2),
    interval = "confidence")
```

does the prediction with standard error for an individual with predictor values `x1 = 4.3` and `x2 = 6.2`.

Can also do

```
predict(out, newdata = data.frame(x1 = x1new, x2 = x2new),
    interval = "confidence")
```

where `x1new` and `x2new` are vectors of the same length, say k, in which case the prediction is done for k individuals with these values.

Generalized Linear Models

Generalized linear models (GLM) are a very large class of statistical models and non-models. Here we are only concerned with the subset that are also aster models. So we won't tell you everything about GLM.

In a GLM like in a LM (linear model), the response y is a numeric vector having independent components. The distributions of the components are all in the same "family" — all binomial, all Poisson, all negative binomial with the same shape parameter, and so forth.

Generalized Linear Models (Cont.)

Given the "family" the distribution of the components of the response is specified by a single parameter, the mean. For example, we may have

$$y_i \sim \text{Ber}(\mu_i)$$
$$y_i \sim \text{Bin}(n_i, \mu_i)$$
$$y_i \sim \text{Poi}(\mu_i)$$

where the wiggle $\sim$ means "distributed as" and "Ber" is short for Bernoulli (binomial with sample size one), "Bin" is short for binomial, and "Poi" is short for Poisson.

A Digression on Bernoulli Random Variables

A random variable is Bernoulli if it is zero-or-one-valued (only possible values are 0 and 1).

Any dichotomous (two-valued) random variable can be recoded to be Bernoulli.

For a Bernoulli random variable y

$$E(y) = \mathsf{Pr}(y = 1)$$

Probability is a special case of expectation. Probability is expectation of Bernoulli random variables.

Generalized Linear Models (Cont.)

In GLM writing

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{e}$$

is part of the problem not part of the solution because $\boldsymbol{\mu}$ is not a location parameter, hence the distribution of $\mathbf{e}$ depends on $\boldsymbol{\mu}$.

Writing things this way brings only confusion. Forget this "mean plus error" stuff!

Generalized Linear Models (Cont.)

In GLM it makes no sense to model the mean μ as a linear function of other parameters

$$\boldsymbol{\mu} = \mathrm{M}\boldsymbol{\beta}$$

because $\boldsymbol{\mu}$ is constrained.

For binomial $0 < \mu_i < 1$. For Poisson $\mu_i < 0$. This would lead to very difficult constraints on the $\boldsymbol{\beta}$ vector.

Generalized Linear Models (Cont.)

Instead model

$$\boldsymbol{\eta} = \mathrm{M}\boldsymbol{\beta}$$

where $\boldsymbol{\eta}$ is componentwise monotone function of $\boldsymbol{\mu}$ and vice versa

$$\eta_i = g(\mu_i)$$

where g is strictly increasing function called *link function*. A strictly increasing function is always invertible, so both $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$ are parameters (either specifies the probability distribution).

Choose link function so that $\boldsymbol{\eta}$ is unconstrained.

Generalized Linear Models (Cont.)

In aster models we always use a special link function (more on this below) called *canonical*. For Bernoulli or binomial we use

$$\eta_i = \text{logit}(\mu_i) = \log\left(\frac{\mu_i}{1 - \mu_i}\right)$$

For Poisson we use

$$\eta_i = \log(\mu_i)$$

These are the default link functions in R so don't need to be specified explicitly.

Differences between LM and GLM

In GLM model linear predictor $\boldsymbol{\eta}$ linearly; In LM model mean $\boldsymbol{\mu}$ linearly.

GLM have two important parameters *linear predictor* $\boldsymbol{\eta}$ and *mean value parameter* $\boldsymbol{\mu}$, which are componentwise monotone functions of each other.

Tests and confidence intervals are no longer exact. Reference distributions for GLM are normal and chi-square whereas those for LM are t and F.

Similarities between LM and GLM

Components of the response are independent.

R functions work similarly. Function `glm` fits GLM using formulas like `lm` fits LM. Function `anova` does tests of model comparison. Function `predict` does "predictions" of both kinds of parameters (linear predictor and mean value).

Generalized Linear Models (Cont.)

```
out <- glm(y ~ x, family = binomial)
```

does Bernoulli (more usually called "logistic") regression.

```
out <- glm(y ~ x, family = poisson)
```

does Poisson regression.

Generalized Linear Models (Cont.): Hypothesis Tests

If two nested models have been fit and stored in `out.little` and `out.big`

```
anova(out.little, out.big, test = "Chisq")
```

does the "analysis of deviance" test also called likelihood ratio test (the test statistic is twice the log likelihood ratio, which is called *deviance* for short).

Generalized Linear Models (Cont.): Predictions

If one model has been fit and stored in `out`

```
predict(out, type = "response")
```

produces "predictions" of the mean values for each case.

To predict a new data, supply a `newdata` argument just like for LM.

If the argument `type = "response"` is omitted the "prediction" is of the linear predictor.

Aster Models (Finally!)

Aster models are a large class of statistical models that include some GLM and LM as special cases.

Aster models are especially designed for life history analysis, but are just statistical models, applicable to whatever data they fit.

Differences between Aster Models and GLM

Not all GLM are aster models, even for those that are.

In GLM components of the response are independent, in aster models can be dependent, hence both conditional and unconditional distributions are relevant.

In GLM components of the response all have same family, in aster models can have different families.

In GLM components relation between linear predictor and mean value parameter is componentwise monotone, in aster models multivariate monotone.

Similarities between Aster Models and GLM

R functions work similarly. Function `aster` fits aster models using formulas like `glm` fits GLM. Function `anova` does tests of model comparison. Function `predict` does "predictions" of four kinds of parameters (linear predictor and mean value, conditional and unconditional).

Graphical Models

In aster models we represent dependence structure using graphs (this is popular in statistics, there are many textbooks with "graphical models" in the title).

Nodes of graph are associated with random variables. Arrows between nodes indicate stochastic dependence.

For arrow $x \longrightarrow y$ say x is *predecessor* and y is *successor*. Variable can be both predecessor and successor. In

$$x \longrightarrow y \longrightarrow z$$

y is predecessor of z and successor of x.

Graphical Models (Cont.)

Aster graphical models are very special case of statistical graphical models.

Every node has at most one predecessor. Graphs are "forests". Nodes can have arbitrarily many successors.

Graphical Models (Cont.)

In "simple" aster models, which are the only ones currently implemented, successors of one node are conditionally independent given that node.

This means joint probability distribution has simple factorization. Let J denote the set of nodes that are successors and F those that are not. Let $\mathbf{y}_J$ and $\mathbf{y}_F$ denote the vectors of variables at these nodes. We consider $\mathbf{y}_J$ the response vector and treat it as random. We consider $\mathbf{y}_F$ non-random (fixed constants). Then

$$f(\mathbf{y}_J \mid \mathbf{y}_F) = \prod_{j \in J} f(y_j \mid y_{p(j)})$$

where $p(j)$ denotes the predecessor of j (pedantically, the index of the node that is the predecessor of the node with index j).

Predecessor is Sample Size

$$\underset{\text{predecessor}}{y_{p(j)}} \longrightarrow \underset{\text{successor}}{y_j}$$

$y_{p(j)}$ is sample size for y_j. Only form of dependence allowed.

This means y_j is the sum of $y_{p(j)}$ independent and identically distributed random variables having some distribution which does not depend on any components of the response.

If this distribution is Ber(μ_j), then $y_j \sim \text{Bin}(y_{p(j)}, \mu_j)$.
If this distribution is Poi(μ_j), then $y_j \sim \text{Poi}(y_{p(j)}\mu_j)$.
In other cases distribution of y_j may not be "brand name".

Random variables of which y_j is the sum are not recorded.

Means, Conditional and Unconditional

"Predecessor is sample size" is the only form of dependence allowed in aster models. It has following important consequence. Define

$$\begin{aligned} \mu_j &= E(y_j) \\ \xi_j &= E(y_j \mid y_{p(j)} = 1) \end{aligned}$$

(ξ_j is the mean of one of the $y_{p(j)}$ independent and identically distributed random variables of which y_j is the sum). Then

$$\begin{aligned} E(y_j \mid y_{p(j)}) &= \xi_j y_{p(j)} \\ E(y_j) &= \xi_j \mu_{p(j)} \\ &= \xi_j \xi_{p(j)} \mu_{p(p(j))} \\ &= \xi_j \xi_{p(j)} \xi_{p(p(j))} \mu_{p(p(p(j)))} \end{aligned}$$

and so forth.

Means, Conditional and Unconditional (Cont.)

If we go back to a root node, say $y_{p(p(p(p(p(j)))))} = 1$, then

$$\mu_j = \xi_j \xi_{p(j)} \xi_{p(p(j))} \xi_{p(p(p(j)))} \xi_{p(p(p(p(j))))}$$

So unconditional means are products of (a certain set of) conditional means.

Very special property of the aster model structure, consequence of predecessor is sample size. Not true of general dependent data.

Means, Conditional and Unconditional (Cont.)

In aster models have two kinds of mean value parameter vectors: conditional ξ and unconditional μ.

Simple but nonlinear relationship between them. Each determines the other, so either is parameter vector.

Key issue in aster modeling is relationship between β, ξ, and μ.

A Digression on Exponential Families

A family of probability density functions (PDF) or probability mass functions (PMF) is *exponential* if it has the following form

$$f_\psi(w) = g(\psi)h(w)\exp\left(\sum_{i=1}^{k}\varphi_i(\psi)y_i(w)\right)$$

where $g(\psi)$ is an arbitrary nonnegative function of the parameter, where $h(w)$ is an arbitrary nonnegative function of the data, and the only term that contains both data and parameters has the exponential form shown here.

A Digression on Exponential Families (Cont.)

Regardless of the original parameter ψ, we can always take the vector φ with components φ_i as the parameter because densities must integrate to one (or sum to one if the data are discrete) so

$$f_\varphi(w) = \frac{h(w) \exp\left(\sum_{i=1}^k \varphi_i y_i(w) \right)}{\int h(w) \exp\left(\sum_{i=1}^k \varphi_i y_i(w) \right) \, dw}$$

Since φ determines the densities, it is a parameter.

This parameterization is very special having many desirable properties; φ is called the *natural* or *canonical* parameter. All of our aster papers and tech reports use *canonical*.

A Digression on Exponential Families (Cont.)

The statistic $\mathbf{y}$ having components $y_i(w)$ that goes with the canonical parameter in the exponential term is called the *natural* or *canonical* statistic. All of our aster papers and tech reports use *canonical*.

We introduce the notation

$$\langle \mathbf{y}, \boldsymbol{\varphi} \rangle = \sum_{i=1}^{k} \varphi_i y_i$$

Those who like to write all vector operations as matrix multiplication would write this as $\mathbf{y}^T \boldsymbol{\varphi}$ or $\mathbf{y}' \boldsymbol{\varphi}$, but we prefer this more mathematical and more elegant notation.

A Digression on Exponential Families (Cont.)

With this notation we can simplify the PDF or PMF

$$f_{\boldsymbol{\varphi}}(w) = e^{\langle \mathbf{y}, \boldsymbol{\varphi} \rangle - c(\boldsymbol{\varphi})} h(w)$$

Terms that do not contain the parameter may be dropped from log likelihoods so we have

$$l(\boldsymbol{\varphi}) = \langle \mathbf{y}, \boldsymbol{\varphi} \rangle - c(\boldsymbol{\varphi})$$

A Digression on Exponential Families (Cont.)

The function $c(\boldsymbol{\varphi})$ determines all the properties of the distribution of the canonical statistic $\mathbf{y}$ and is called the *cumulant* function.

In particular,

$$E_{\boldsymbol{\varphi}}(\mathbf{y}) = \nabla c(\boldsymbol{\varphi}) \tag{1}$$

$$\text{var}_{\boldsymbol{\varphi}}(\mathbf{y}) = \nabla^2 c(\boldsymbol{\varphi}) \tag{2}$$

(1) is a vector equation and (2) is a matrix equation.

A Digression on Exponential Families (Cont.)

Applying (1) to the log likelihood gives

$$\begin{aligned} \nabla l(\boldsymbol{\varphi}) &= \mathbf{y} - \nabla c(\boldsymbol{\varphi}) \\ &= \mathbf{y} - E_{\boldsymbol{\varphi}}(\mathbf{y}) \end{aligned}$$

and the maximum likelihood estimate (MLE) of $\boldsymbol{\varphi}$, if it exists (more on this later) is the value of $\boldsymbol{\varphi}$ that makes this zero.

Thus maximum likelihood estimation sets the expected value of the canonical statistic equal to its observed value. For short, "observed equals expected".

A Digression on Exponential Families (Cont.)

Applying (2) to the log likelihood gives

$$\begin{aligned} \nabla^2 l(\varphi) &= -\nabla^2 c(\varphi) \\ &= -\operatorname{var}_\varphi(\mathbf{y}) \end{aligned}$$

The latter, being the negative of a variance is a negative definite matrix unless the distribution of $\mathbf{y}$ is degenerate (more on this below) which implies that the log likelihood is a strictly concave function and the MLE is unique if it exists.

The left hand side of this equation is minus Fisher information (no difference between observed and expected Fisher information because $\nabla^2 l(\varphi)$ is not a function of the data $\mathbf{y}$). Thus Fisher information is

$$I(\varphi) = \operatorname{var}_\varphi(\mathbf{y})$$

A Digression on Exponential Families (Cont.)

Thus likelihood inference in exponential families is very simple.

- The MLE is unique if it exists.
- Fisher information is calculated by two derivatives of the cumulant function, no integrals or sums necessary.
- Any algorithm that always goes uphill on the log likelihood cannot fail to find the MLE.

A Digression on Exponential Families (Cont.)

We can not only do maximum likelihood with possible data values.

For any vector $\boldsymbol{\mu}$ the function

$$\langle \boldsymbol{\mu}, \boldsymbol{\varphi} \rangle - c(\boldsymbol{\varphi}),$$

is strictly concave so the solution for $\boldsymbol{\varphi}$ of

$$\boldsymbol{\mu} = E_{\boldsymbol{\varphi}}(\mathbf{y})$$

is unique if it exists. Thus $\boldsymbol{\mu}$ determines $\boldsymbol{\varphi}$ as well as vice versa.

Hence $\boldsymbol{\mu}$ is a parameter called the *mean value parameter*. If multiple parameterizations bother you, just call it the mean, but remember that this is not any mean but the mean of the canonical statistic.

A Digression on Exponential Families (Cont.)

The relationship between the canonical parameter $\boldsymbol{\varphi}$ and the mean value parameter $\boldsymbol{\mu}$ is *multivariate monotone*. This means that if $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ are distinct canonical parameter values and $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the corresponding mean value parameter values, that is,

$$\boldsymbol{\mu}_i = E_{\boldsymbol{\theta}_i}(\mathbf{y}),$$

then

$$\langle \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2, \boldsymbol{\varphi}_1 - \boldsymbol{\varphi}_2 \rangle > 0 \tag{3}$$

In particular, increasing one component of $\boldsymbol{\varphi}$ leaving the rest fixed increases the corresponding component of $\boldsymbol{\mu}$ (the other components of $\boldsymbol{\mu}$ also change one way or the other). But (3) is a much stronger statement than this. And (3) is the key to reasoning about the connection between canonical and mean value parameters.

A Digression on Exponential Families (Cont.)

Independent and identically distributed (IID) sampling does not take us out of exponential families. Suppose w_1, ..., x_n are IID data having an exponential family and $\mathbf{y}_1$, ..., $\mathbf{y}_n$ are the corresponding canonical statistics. Then

$$\begin{aligned} f_{\boldsymbol{\theta}}(w_1, \ldots, w_n) &= \prod_{i=1}^{n} e^{\langle \mathbf{y}_i, \boldsymbol{\varphi} \rangle - c(\boldsymbol{\varphi})} h(w_i) \\ &= e^{\langle \mathbf{y}_1 + \cdots + \mathbf{y}_n, \boldsymbol{\varphi} \rangle - nc(\boldsymbol{\varphi})} \prod_{i=1}^{n} h(w_i) \end{aligned}$$

so we again have an exponential family with canonical statistic $\mathbf{y}_1 + \cdots + \mathbf{y}_n$, canonical parameter $\boldsymbol{\varphi}$, and cumulant function $nc(\boldsymbol{\varphi})$.

Affine Functions

In real math what most people call linear functions $x \mapsto a + bx$ are called *affine functions*. The function $x \mapsto bx$ (with no "intercept") is a *linear function*. This is the sense of "linear" used in the subject linear algebra. More generally, a vector-to-vector *affine function* has the form

$$g(\beta) = \mathbf{a} + \mathbf{M}\beta$$

where $\mathbf{a}$ is a vector and $\mathbf{M}$ is a matrix of the appropriate dimensions for this definition to make sense. And this is a *linear function* if and only if $\mathbf{a} = 0$.

Affine Functions (Cont.)

So why the pedantry about affine and linear? The "linear" in *linear models* really means linear in the pedantic sense.

How can that be when models have "intercepts"? Because the "linear" in linear models refers to $\boldsymbol{\mu} = \mathbf{M}\boldsymbol{\beta}$ being a linear function of $\boldsymbol{\beta}$ — which it is in the pedantic sense —- and not being a linear function of the predictor $\mathbf{x}$ — which it need not be, it can be arbitrary. The "intercept" is an intercept considered as a function of $\mathbf{x}$ not $\boldsymbol{\beta}$.

An *affine model* would be

$$\boldsymbol{\eta} = \mathbf{a} + \mathbf{M}\boldsymbol{\beta}$$

where $\mathbf{a}$ is a known vector and $\mathbf{M}$ a known matrix (possibly functions of the predictor but not functions of the response).

Affine Functions (Cont.)

GLM (generalized *linear* models) are actually misnamed, because affine models $\boldsymbol{\eta} = \mathbf{a} + \mathbf{M}\boldsymbol{\beta}$ are actually allowed and occasionally useful. $\mathbf{M}$ is still called the model matrix, and $\mathbf{a}$ is called the *offset* in the documentation for the R function GLM.

A Digression on Exponential Families (Cont.)

Cannonical affine submodels do not take us out of exponential families. Suppose $\boldsymbol{\varphi} = \mathbf{a} + \mathbf{M}\boldsymbol{\beta}$. Then

$$\begin{aligned} f_{\boldsymbol{\theta}}(w) &= e^{\langle \mathbf{y}, \mathbf{a} + \mathbf{M}\boldsymbol{\beta} \rangle - c(\mathbf{a} + \mathbf{M}\boldsymbol{\beta})} h(w) \\ &= e^{\langle \mathbf{M}^T \mathbf{y}, \boldsymbol{\beta} \rangle - c(\mathbf{a} + \mathbf{M}\boldsymbol{\beta})} h(w) e^{\langle \mathbf{y}, \mathbf{a} \rangle} \end{aligned}$$

so we again have an exponential family with canonical statistic $\mathbf{M}^T \mathbf{y}$, canonical parameter $\boldsymbol{\beta}$, and cumulant function $c(\mathbf{a} + \mathbf{M}\boldsymbol{\beta})$.