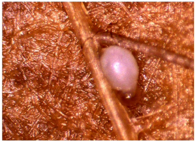
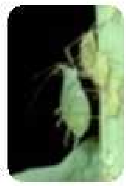
Inverse Methods For Time Series

Thomas Shores
Department of Mathematics
University of Nebraska

April 27, 2006

The suspects: Freshwater copepod, freshwater turtle, female cyst nematode, pea aphid.

Outline

Outline

References

1. R. Aster, B. Borchers, C. Thurber, *Estimation and Inverse Problems,* Elsivier, New York, 2005.
2. H. Caswell, *Matrix Population Models, 2nd Ed.,* Sinaur Associates, Sunderland, MA, 2001.
3. K. Gross, B. Craig and W. Hutchison, *Bayesian estimation of a demographic matrix model from stage-frequency data,* Ecology 83(12), 3285–3298, 2002.
4. J. Mitchell, *Population ecology and life histories of the freshwater turtles Chrysemys Picta and Sternotherus Odoratus in an urban lake,* Herpetological Monographs (2), 40–61, 1989.
5. S Twombly, *Comparative demography and population dynamics of two coexisting copepods in a Venezuelan floodplain lake*, Limnol. Oceanogr. 39(2), 234–247, 1994.
6. S Twombly and C. Burns, *Exuvium analysis: A nondestructive method of analyzing copepod growth and development*, Limnol. Oceanogr. 41(6), 1324–1329, 1996.
7. S. Wood, *Inverse problems and structured-population dynamics,* Structured-population Models in Marine, Terrestrial and Freshwater Systems (Ed. S. Tuljapurkar and J. Caswell), Chapman and Hall, New York, 555–586, 1997.

References

1. R. Aster, B. Borchers, C. Thurber, *Estimation and Inverse Problems*, Elsivier, New York, 2005.
2. H. Caswell, *Matrix Population Models, 2nd Ed.*, Sinaur Associates, Sunderland, MA, 2001.
3. K. Gross, B. Craig and W. Hutchison, *Bayesian estimation of a demographic matrix model from stage-frequency data*, Ecology 83(12), 3285–3298, 2002.
4. J. Mitchell, *Population ecology and life histories of the freshwater turtles Chrysemys Picta and Sternotherus Odoratus in an urban lake*, Herpetological Monographs (2), 40–61, 1989.
5. S Twombly, *Comparative demography and population dynamics of two coexisting copepods in a Venezuelan floodplain lake*, Limnol. Oceanogr. 39(2), 234–247, 1994.
6. S Twombly and C. Burns, *Exuvium analysis: A nondestructive method of analyzing copepod growth and development*, Limnol. Oceanogr. 41(6), 1324–1329, 1996.
7. S. Wood, *Inverse problems and structured-population dynamics*, Structured-population Models in Marine, Terrestrial and Freshwater Systems (Ed. S. Tuljapurkar and J. Caswell), Chapman and Hall, New York, 555–586, 1997.

References

1. R. Aster, B. Borchers, C. Thurber, *Estimation and Inverse Problems,* Elsivier, New York, 2005.
2. H. Caswell, *Matrix Population Models, 2nd Ed.*, Sinaur Associates, Sunderland, MA, 2001.
3. K. Gross, B. Craig and W. Hutchison, *Bayesian estimation of a demographic matrix model from stage-frequency data,* Ecology 83(12), 3285–3298, 2002.
4. J. Mitchell, *Population ecology and life histories of the freshwater turtles Chrysemys Picta and Sternotherus Odoratus in an urban lake,* Herpetological Monographs (2), 40–61, 1989.
5. S Twombly, *Comparative demography and population dynamics of two coexisting copepods in a Venezuelan floodplain lake*, Limnol. Oceanogr. 39(2), 234–247, 1994.
6. S Twombly and C. Burns, *Exuvium analysis: A nondestructive method of analyzing copepod growth and development*, Limnol. Oceanogr. 41(6), 1324–1329, 1996.
7. S. Wood, *Inverse problems and structured-population dynamics,* Structured-population Models in Marine, Terrestrial and Freshwater Systems (Ed. S. Tuljapurkar and J. Caswell), Chapman and Hall, New York, 555–586, 1997.

References

1. R. Aster, B. Borchers, C. Thurber, *Estimation and Inverse Problems*, Elsivier, New York, 2005.
2. H. Caswell, *Matrix Population Models, 2nd Ed.*, Sinaur Associates, Sunderland, MA, 2001.
3. K. Gross, B. Craig and W. Hutchison, *Bayesian estimation of a demographic matrix model from stage-frequency data*, Ecology 83(12), 3285–3298, 2002.
4. J. Mitchell, *Population ecology and life histories of the freshwater turtles Chrysemys Picta and Sternotherus Odoratus in an urban lake*, Herpetological Monographs (2), 40–61, 1989.
5. S Twombly, *Comparative demography and population dynamics of two coexisting copepods in a Venezuelan floodplain lake*, Limnol. Oceanogr. 39(2), 234–247, 1994.
6. S Twombly and C. Burns, *Exuvium analysis: A nondestructive method of analyzing copepod growth and development*, Limnol. Oceanogr. 41(6), 1324–1329, 1996.
7. S. Wood, *Inverse problems and structured-population dynamics*, Structured-population Models in Marine, Terrestrial and Freshwater Systems (Ed. S. Tuljapurkar and J. Caswell), Chapman and Hall, New York, 555–586, 1997.

References

1. R. Aster, B. Borchers, C. Thurber, *Estimation and Inverse Problems*, Elsivier, New York, 2005.
2. H. Caswell, *Matrix Population Models, 2nd Ed.*, Sinaur Associates, Sunderland, MA, 2001.
3. K. Gross, B. Craig and W. Hutchison, *Bayesian estimation of a demographic matrix model from stage-frequency data*, Ecology 83(12), 3285–3298, 2002.
4. J. Mitchell, *Population ecology and life histories of the freshwater turtles Chrysemys Picta and Sternotherus Odoratus in an urban lake*, Herpetological Monographs (2), 40–61, 1989.
5. S Twombly, *Comparative demography and population dynamics of two coexisting copepods in a Venezuelan floodplain lake*, Limnol. Oceanogr. 39(2), 234–247, 1994.
6. S Twombly and C. Burns, *Exuvium analysis: A nondestructive method of analyzing copepod growth and development*, Limnol. Oceanogr. 41(6), 1324–1329, 1996.
7. S. Wood, *Inverse problems and structured-population dynamics*, Structured-population Models in Marine, Terrestrial and Freshwater Systems (Ed. S. Tuljapurkar and J. Caswell), Chapman and Hall, New York, 555–586, 1997.

References

1. R. Aster, B. Borchers, C. Thurber, *Estimation and Inverse Problems*, Elsivier, New York, 2005.
2. H. Caswell, *Matrix Population Models, 2nd Ed.*, Sinaur Associates, Sunderland, MA, 2001.
3. K. Gross, B. Craig and W. Hutchison, *Bayesian estimation of a demographic matrix model from stage-frequency data*, Ecology 83(12), 3285–3298, 2002.
4. J. Mitchell, *Population ecology and life histories of the freshwater turtles Chrysemys Picta and Sternotherus Odoratus in an urban lake*, Herpetological Monographs (2), 40–61, 1989.
5. S Twombly, *Comparative demography and population dynamics of two coexisting copepods in a Venezuelan floodplain lake*, Limnol. Oceanogr. 39(2), 234–247, 1994.
6. S Twombly and C. Burns, *Exuvium analysis: A nondestructive method of analyzing copepod growth and development*, Limnol. Oceanogr. 41(6), 1324–1329, 1996.
7. S. Wood, *Inverse problems and structured-population dynamics*, Structured-population Models in Marine, Terrestrial and Freshwater Systems (Ed. S. Tuljapurkar and J. Caswell), Chapman and Hall, New York, 555–586, 1997.

References

1. R. Aster, B. Borchers, C. Thurber, *Estimation and Inverse Problems*, Elsivier, New York, 2005.
2. H. Caswell, *Matrix Population Models, 2nd Ed.*, Sinaur Associates, Sunderland, MA, 2001.
3. K. Gross, B. Craig and W. Hutchison, *Bayesian estimation of a demographic matrix model from stage-frequency data*, Ecology 83(12), 3285–3298, 2002.
4. J. Mitchell, *Population ecology and life histories of the freshwater turtles Chrysemys Picta and Sternotherus Odoratus in an urban lake*, Herpetological Monographs (2), 40–61, 1989.
5. S Twombly, *Comparative demography and population dynamics of two coexisting copepods in a Venezuelan floodplain lake*, Limnol. Oceanogr. 39(2), 234–247, 1994.
6. S Twombly and C. Burns, *Exuvium analysis: A nondestructive method of analyzing copepod growth and development*, Limnol. Oceanogr. 41(6), 1324–1329, 1996.
7. S. Wood, *Inverse problems and structured-population dynamics*, Structured-population Models in Marine, Terrestrial and Freshwater Systems (Ed. S. Tuljapurkar and J. Caswell), Chapman and Hall, New York, 555–586, 1997.

States, Age and Stage

About States:

- Assume at any one time members of a time varying population are classified into one of s mutually exclusive states, indicated by the index i, whence population vector is

$$\mathbf{n}(t) = (n_1(t), \ldots, n_s(t)) \equiv [n_1(t), \ldots, n_s(t)]^T$$

- All individuals experience the same environment.
- The effects of the population on the environment can be written as the sum of the contributions of the individuals.
- Candidates: age, size, instars, genders, geographical locales (patches) and and meaningful combination of these states.

States, Age and Stage

About States:

- Assume at any one time members of a time varying population are classified into one of s mutually exclusive states, indicated by the index i, whence population vector is

$$\mathbf{n}(t) = (n_1(t), \ldots, n_s(t)) \equiv [n_1(t), \ldots, n_s(t)]^T$$

- All individuals experience the same environment.
- The effects of the population on the environment can be written as the sum of the contributions of the individuals.
- Candidates: age, size, instars, genders, geographical locales (patches) and and meaningful combination of these states.

States, Age and Stage

About States:

- Assume at any one time members of a time varying population are classified into one of s mutually exclusive states, indicated by the index i, whence population vector is

$$\mathbf{n}(t) = (n_1(t), \ldots, n_s(t)) \equiv [n_1(t), \ldots, n_s(t)]^T$$

- All individuals experience the same environment.
- The effects of the population on the environment can be written as the sum of the contributions of the individuals.
- Candidates: age, size, instars, genders, geographical locales (patches) and and meaningful combination of these states.

States, Age and Stage

About States:

- Assume at any one time members of a time varying population are classified into one of s mutually exclusive states, indicated by the index i, whence population vector is

$$\mathbf{n}(t) = (n_1(t), \ldots, n_s(t)) \equiv [n_1(t), \ldots, n_s(t)]^T$$

- All individuals experience the same environment.
- The effects of the population on the environment can be written as the sum of the contributions of the individuals.
- Candidates: age, size, instars, genders, geographical locales (patches) and and meaningful combination of these states.

States, Age and Stage

About States:

- Assume at any one time members of a time varying population are classified into one of s mutually exclusive states, indicated by the index i, whence population vector is

$$\mathbf{n}(t) = (n_1(t), \ldots, n_s(t)) \equiv [n_1(t), \ldots, n_s(t)]^T$$

- All individuals experience the same environment.
- The effects of the population on the environment can be written as the sum of the contributions of the individuals.
- Candidates: age, size, instars, genders, geographical locales (patches) and and meaningful combination of these states.

Linear Form of Classic Examples

Projection Model:

$$\mathbf{n}(t+1) = A_t \mathbf{n}(t)$$

where $A_t = [a_{i,j}]$ is $s \times s$ projection matrix.

- Note: coefficients could vary with time. If not, this is a stationary (or time-invariant) model and we simply write $A_t = A$.
- Coefficients could even be non-local: e.g., birth rates could be dependent on a carrying capacity of environment. Ditto other forms of recruitment.

Linear Form of Classic Examples

Projection Model:

$$\mathbf{n}(t+1) = A_t \mathbf{n}(t)$$

where $A_t = [a_{i,j}]$ is $s \times s$ projection matrix.

- Note: coefficients could vary with time. If not, this is a stationary (or time-invariant) model and we simply write $A_t = A$.
- Coefficients could even be non-local: e.g., birth rates could be dependent on a carrying capacity of environment. Ditto other forms of recruitment.

Linear Form of Classic Examples

Projection Model:

$$\mathbf{n}(t+1) = A_t \mathbf{n}(t)$$

where $A_t = [a_{i,j}]$ is $s \times s$ projection matrix.

- Note: coefficients could vary with time. If not, this is a stationary (or time-invariant) model and we simply write $A_t = A$.
- Coefficients could even be non-local: e.g., birth rates could be dependent on a carrying capacity of environment. Ditto other forms of recruitment.

Leslie Model (~1945)

Population is divided into discrete age groups, resulting in projection matrix

$$A = \begin{bmatrix} F_1 & F_2 & \cdots & F_{s-1} & F_s \\ P_1 & 0 & \cdots & & 0 \\ 0 & P_2 & \ddots & & \vdots \\ & & \ddots & 0 & 0 \\ & & & P_{s-1} & 0 \end{bmatrix}.$$

- Here F_i is the per-capita fertility of age class i and P_j is the survival rate of age class j. Clearly $0 \le P_j \le 1$.
- Linearity or stationarity not required.
- Example: $A = \begin{bmatrix} 0 & e^{-bN} & 5e^{-bN} \\ .3 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$, where $N = n_1 + n_2$.

Leslie Model (~1945)

Population is divided into discrete age groups, resulting in projection matrix

$$A = \begin{bmatrix} F_1 & F_2 & \cdots & F_{s-1} & F_s \\ P_1 & 0 & \cdots & & 0 \\ 0 & P_2 & \ddots & & \vdots \\ & & \ddots & 0 & 0 \\ & & & P_{s-1} & 0 \end{bmatrix}.$$

- Here F_i is the per-capita fertility of age class i and P_j is the survival rate of age class j. Clearly $0 \leq P_j \leq 1$.
- Linearity or stationarity not required.
- Example: $A = \begin{bmatrix} 0 & e^{-bN} & 5e^{-bN} \\ .3 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$, where $N = n_1 + n_2$.

Leslie Model (~1945)

Population is divided into discrete age groups, resulting in projection matrix

$$A = \begin{bmatrix} F_1 & F_2 & \cdots & F_{s-1} & F_s \\ P_1 & 0 & \cdots & & 0 \\ 0 & P_2 & \ddots & & \vdots \\ & & \ddots & 0 & 0 \\ & & & P_{s-1} & 0 \end{bmatrix}.$$

- Here F_i is the per-capita fertility of age class i and P_j is the survival rate of age class j. Clearly $0 \leq P_j \leq 1$.
- Linearity or stationarity not required.
- Example: $A = \begin{bmatrix} 0 & e^{-bN} & 5e^{-bN} \\ .3 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$, where $N = n_1 + n_2$.

Leslie Model (~1945)

Population is divided into discrete age groups, resulting in projection matrix

$$A = \begin{bmatrix} F_1 & F_2 & \cdots & F_{s-1} & F_s \\ P_1 & 0 & \cdots & & 0 \\ 0 & P_2 & \ddots & & \vdots \\ & & \ddots & 0 & 0 \\ & & & P_{s-1} & 0 \end{bmatrix}.$$

- Here F_i is the per-capita fertility of age class i and P_j is the survival rate of age class j. Clearly $0 \leq P_j \leq 1$.
- Linearity or stationarity not required.
- Example: $A = \begin{bmatrix} 0 & e^{-bN} & 5e^{-bN} \\ .3 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$, where $N = n_1 + n_2$.

Lefkovitch Model (~1962)

Population is divided into discrete stages, resulting in a very general projection matrix $A = [a_{i,j}]_{s,s}$.

- Here one chooses a projection interval $(t, t+1)$ representing the time states of the model.
- Linearity or stationarity not required.
- The entry $a_{i,j}$ represents a rate (or probability) of passage from stage i to stage j. (Hence $a_{i,j} \geq 0$.)
- The matrix A is equivalent to a (directed) life cycle graph for the population.

Lefkovitch Model (~1962)

Population is divided into discrete stages, resulting in a very general projection matrix $A = [a_{i,j}]_{s,s}$.

- Here one chooses a projection interval $(t, t+1)$ representing the time states of the model.
- Linearity or stationarity not required.
- The entry $a_{i,j}$ represents a rate (or probability) of passage from stage i to stage j. (Hence $a_{i,j} \geq 0$.)
- The matrix A is equivalent to a (directed) life cycle graph for the population.

Lefkovitch Model (~1962)

Population is divided into discrete stages, resulting in a very general projection matrix $A = [a_{i,j}]_{s,s}$.

- Here one chooses a projection interval $(t, t+1)$ representing the time states of the model.
- Linearity or stationarity not required.
- The entry $a_{i,j}$ represents a rate (or probability) of passage from stage i to stage j. (Hence $a_{i,j} \geq 0$.)
- The matrix A is equivalent to a (directed) life cycle graph for the population.

Lefkovitch Model (~1962)

Population is divided into discrete stages, resulting in a very general projection matrix $A = [a_{i,j}]_{s,s}$.

- Here one chooses a projection interval $(t, t+1)$ representing the time states of the model.
- Linearity or stationarity not required.
- The entry $a_{i,j}$ represents a rate (or probability) of passage from stage i to stage j. (Hence $a_{i,j} \geq 0$.)
- The matrix A is equivalent to a (directed) life cycle graph for the population.

Lefkovitch Model (~1962)

Population is divided into discrete stages, resulting in a very general projection matrix $A = [a_{i,j}]_{s,s}$.

- Here one chooses a projection interval $(t, t+1)$ representing the time states of the model.
- Linearity or stationarity not required.
- The entry $a_{i,j}$ represents a rate (or probability) of passage from stage i to stage j. (Hence $a_{i,j} \geq 0$.)
- The matrix A is equivalent to a (directed) life cycle graph for the population.

Tensor Notation

Tensor (Kronecker, direct) Product:

Given $m \times n$ matrix $A = [a_{i,j}]$ and $p \times q$ matrix $B = [b_{i,j}]$, the tensor product of A and B is the $mp \times nq$ block matrix

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,n}B \\ \vdots & & \ddots & \vdots \\ a_{m,1}B & a_{m,2}B & \cdots & a_{m,1}B \end{bmatrix}.$$

- There are lots of algebra laws, e.g., $(A \otimes (\beta B + \gamma C) = \beta A \otimes B + \gamma A \otimes C$ and interesting properties, e.g., eigenvalues of $A \otimes B$ are just the products of eigenvalues of A and B, etc., etc., that we won't need.
- Matlab knows tensors: `C = kron(A,B)` does the trick.

Tensor Notation

Tensor (Kronecker, direct) Product:

Given $m \times n$ matrix $A = [a_{i,j}]$ and $p \times q$ matrix $B = [b_{i,j}]$, the tensor product of A and B is the $mp \times nq$ block matrix

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,n}B \\ \vdots & & \ddots & \vdots \\ a_{m,1}B & a_{m,2}B & \cdots & a_{m,1}B \end{bmatrix}.$$

- There are lots of algebra laws, e.g., $(A \otimes (\beta B + \gamma C) = \beta A \otimes B + \gamma A \otimes C$ and interesting properties, e.g., eigenvalues of $A \otimes B$ are just the products of eigenvalues of A and B, etc., etc., that we won't need.
- Matlab knows tensors: `C = kron(A,B)` does the trick.

Tensor Notation

Tensor (Kronecker, direct) Product:

Given $m \times n$ matrix $A = [a_{i,j}]$ and $p \times q$ matrix $B = [b_{i,j}]$, the tensor product of A and B is the $mp \times nq$ block matrix

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,n}B \\ \vdots & & \ddots & \vdots \\ a_{m,1}B & a_{m,2}B & \cdots & a_{m,1}B \end{bmatrix}.$$

- There are lots of algebra laws, e.g., $(A \otimes (\beta B + \gamma C) = \beta A \otimes B + \gamma A \otimes C$ and interesting properties, e.g., eigenvalues of $A \otimes B$ are just the products of eigenvalues of A and B, etc., etc., that we won't need.
- Matlab knows tensors: `C = kron(A,B)` does the trick.

Tensors as Bookkeeper

Vec or Co(concatenate) Notion:

Given $m \times n$ matrix $A = [a_{i,j}] = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ as a row of columns,

$$\text{vec}(A) = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}.$$

- There are more algebra laws, e.g., $\text{vec}(\alpha A + \beta B) = \alpha\,\text{vec}(A) + \beta\,\text{vec}(B)$.
- **Key Bookkeeping Property:**

$$\text{vec}(AXB) = \left(B^T \otimes A\right)\text{vec}(X).$$

- Matlab knows vec. The command `x = vec(X)` does the trick.

Vec or Co(concatenate) Notion:

Given $m \times n$ matrix $A = [a_{i,j}] = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ as a row of columns,

$$\mathrm{vec}\,(A) = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}.$$

- There are more algebra laws, e.g., $\mathrm{vec}\,(\alpha A + \beta B) = \alpha\,\mathrm{vec}\,(A) + \beta\,\mathrm{vec}\,(B)$.
- Key Bookkeeping Property:

$$\mathrm{vec}\,(AXB) = \left(B^T \otimes A\right) \mathrm{vec}\,(X).$$

- Matlab knows vec. The command `x = vec(X)` does the trick.

Tensors as Bookkeeper

Vec or Co(concatenate) Notion:

Given $m \times n$ matrix $A = [a_{i,j}] = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ as a row of columns,

$$\mathrm{vec}\,(A) = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}.$$

- There are more algebra laws, e.g., $\mathrm{vec}\,(\alpha A + \beta B) = \alpha\,\mathrm{vec}\,(A) + \beta\,\mathrm{vec}\,(B)$.
- **Key Bookkeeping Property:**

$$\mathrm{vec}\,(AXB) = \left(B^T \otimes A\right) \mathrm{vec}\,(X)\,.$$

- Matlab knows vec. The command `x = vec(X)` does the trick.

Formulation

The Problems:

- Forward (Direct) Problem: Given a question, find the answer. E.g., given a projection matrix A and present system state $\mathbf{n}(t)$, find the next state of the system $\mathbf{n}(t+1)$. Solution: $\mathbf{n}(t+1) = A\mathbf{n}(t)$.
- Inverse Problem: Given an answer, find the question. E.g., given a projection matrix A and present system state $\mathbf{n}(t)$, find the *previous* state of the system $\mathbf{n}(t-1)$. Solution: $\mathbf{n}(t-1) = A^{-1}\mathbf{n}(t)$ (maybe!)
- Parameter Identification: a special class of inverse problems that finds parameters of a model, e.g., given many system states $\mathbf{n}(t_i)$, find the projection matrix. This is tougher.

Formulation

The Problems:

- Forward (Direct) Problem: Given a question, find the answer. E.g., given a projection matrix A and present system state $\mathbf{n}(t)$, find the next state of the system $\mathbf{n}(t+1)$. Solution: $\mathbf{n}(t+1) = A\mathbf{n}(t)$.
- Inverse Problem: Given an answer, find the question. E.g., given a projection matrix A and present system state $\mathbf{n}(t)$, find the *previous* state of the system $\mathbf{n}(t-1)$. Solution: $\mathbf{n}(t-1) = A^{-1}\mathbf{n}(t)$ (maybe!)
- Parameter Identification: a special class of inverse problems that finds parameters of a model, e.g., given many system states $\mathbf{n}(t_i)$, find the projection matrix. This is tougher.

Formulation

The Problems:

- Forward (Direct) Problem: Given a question, find the answer. E.g., given a projection matrix A and present system state $\mathbf{n}(t)$, find the next state of the system $\mathbf{n}(t+1)$. Solution: $\mathbf{n}(t+1) = A\mathbf{n}(t)$.
- Inverse Problem: Given an answer, find the question. E.g., given a projection matrix A and present system state $\mathbf{n}(t)$, find the *previous* state of the system $\mathbf{n}(t-1)$. Solution: $\mathbf{n}(t-1) = A^{-1}\mathbf{n}(t)$ (maybe!)
- Parameter Identification: a special class of inverse problems that finds parameters of a model, e.g., given many system states $\mathbf{n}(t_i)$, find the projection matrix. This is tougher.

Formulation

The Problems:

- Forward (Direct) Problem: Given a question, find the answer. E.g., given a projection matrix A and present system state $\mathbf{n}(t)$, find the next state of the system $\mathbf{n}(t+1)$. Solution: $\mathbf{n}(t+1) = A\mathbf{n}(t)$.
- Inverse Problem: Given an answer, find the question. E.g., given a projection matrix A and present system state $\mathbf{n}(t)$, find the *previous* state of the system $\mathbf{n}(t-1)$. Solution: $\mathbf{n}(t-1) = A^{-1}\mathbf{n}(t)$ (maybe!)
- Parameter Identification: a special class of inverse problems that finds parameters of a model, e.g., given many system states $\mathbf{n}(t_i)$, find the projection matrix. This is tougher.

Nature of Inverse Problems

Question:

Given a two-way process with "answers" at both ends, what makes one direction "direct" and the other "inverse"?

- Forward problems are generally **well-posed**, that is, have a solution, it is unique and it varies continuously with the parameters of the problem
- Inverse problems are generally **ill-posed**, i.e., not well-posed, and all three possible failings occur in the simple inverse problem of solving $A\mathbf{x} = \mathbf{b}$ for the unknown x, given $A, \mathbf{b}$.

Nature of Inverse Problems

Question:

Given a two-way process with "answers" at both ends, what makes one direction "direct" and the other "inverse"?

- Forward problems are generally **well-posed**, that is, have a solution, it is unique and it varies continuously with the parameters of the problem
- Inverse problems are generally **ill-posed**, i.e., not well-posed, and all three possible failings occur in the simple inverse problem of solving $A\mathbf{x} = \mathbf{b}$ for the unknown x, given $A, \mathbf{b}$.

Nature of Inverse Problems

Question:

Given a two-way process with "answers" at both ends, what makes one direction "direct" and the other "inverse"?

- Forward problems are generally **well-posed**, that is, have a solution, it is unique and it varies continuously with the parameters of the problem
- Inverse problems are generally **ill-posed**, i.e., not well-posed, and all three possible failings occur in the simple inverse problem of solving $A\mathbf{x} = \mathbf{b}$ for the unknown x, given $A, \mathbf{b}$.

A Generic Example

Postulate $s \times s$ projection matrix A for a stage structured population, together with data (possibly replicated and averaged) for the states $\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)$. We have prior knowledge of A: all entries are nonnegative and certain entries are zero.

Frame the problem as follows:

- $A\mathbf{n}(k) = \mathbf{n}(k+1)$, $k = 1, \ldots, s$.
- Set $M = [\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s)]$ and $P = [\mathbf{n}(2), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)]$.
- Recast problem as $AM = P = I_s AN$.
- Tensor bookkeeping: $\operatorname{vec}(I_s AM) = \left(M^T \otimes I_s\right) \operatorname{vec}(A) = \operatorname{vec}(P) \equiv \mathbf{d}$.
- Delete zero variables from $\operatorname{vec}(A)$ and columns of $M^T \otimes I_s$.
- Rename resulting vectors, matrix as $\mathbf{m}, G$ and rhs $\mathbf{d}$. System takes form $G\mathbf{m} = \mathbf{d}$, with G $p \times q$, typically $p > q$.

A Generic Example

Postulate $s \times s$ projection matrix A for a stage structured population, together with data (possibly replicated and averaged) for the states $\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)$. We have prior knowledge of A: all entries are nonnegative and certain entries are zero.

Frame the problem as follows:

- $A\mathbf{n}(k) = \mathbf{n}(k+1)$, $k = 1, \ldots, s$.
- Set $M = [\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s)]$ and $P = [\mathbf{n}(2), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)]$.
- Recast problem as $AM = P = I_s AN$.
- Tensor bookkeeping: $\mathrm{vec}(I_s AM) = \left(M^T \otimes I_s\right) \mathrm{vec}(A) = \mathrm{vec}(P) \equiv \mathbf{d}$.
- Delete zero variables from $\mathrm{vec}(A)$ and columns of $M^T \otimes I_s$.
- Rename resulting vectors, matrix as $\mathbf{m}, G$ and rhs $\mathbf{d}$. System takes form $G\mathbf{m} = \mathbf{d}$, with G $p \times q$, typically $p > q$.

A Generic Example

Postulate $s \times s$ projection matrix A for a stage structured population, together with data (possibly replicated and averaged) for the states $\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)$. We have prior knowledge of A: all entries are nonnegative and certain entries are zero.

Frame the problem as follows:

- $A\mathbf{n}(k) = \mathbf{n}(k+1),\ k = 1, \ldots, s.$
- Set $M = [\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s)]$ and $P = [\mathbf{n}(2), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)]$.
- Recast problem as $AM = P = I_s AN$.
- Tensor bookkeeping: $\text{vec}(I_s AM) = \left(M^T \otimes I_s\right) \text{vec}(A) = \text{vec}(P) \equiv \mathbf{d}.$
- Delete zero variables from $\text{vec}(A)$ and columns of $M^T \otimes I_s$.
- Rename resulting vectors, matrix as $\mathbf{m}, G$ and rhs $\mathbf{d}$. System takes form $G\mathbf{m} = \mathbf{d}$, with G $p \times q$, typically $p > q$.

A Generic Example

Postulate $s \times s$ projection matrix A for a stage structured population, together with data (possibly replicated and averaged) for the states $\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)$. We have prior knowledge of A: all entries are nonnegative and certain entries are zero.

Frame the problem as follows:

- $A\mathbf{n}(k) = \mathbf{n}(k+1),\ k = 1, \ldots, s.$
- Set $M = [\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s)]$ and $P = [\mathbf{n}(2), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)]$.
- Recast problem as $AM = P = I_s AN$.
- Tensor bookkeeping: $\operatorname{vec}(I_s AM) = \left(M^T \otimes I_s\right) \operatorname{vec}(A) = \operatorname{vec}(P) \equiv \mathbf{d}$.
- Delete zero variables from $\operatorname{vec}(A)$ and columns of $M^T \otimes I_s$.
- Rename resulting vectors, matrix as $\mathbf{m}, G$ and rhs $\mathbf{d}$. System takes form $G\mathbf{m} = \mathbf{d}$, with G $p \times q$, typically $p > q$.

A Generic Example

Postulate $s \times s$ projection matrix A for a stage structured population, together with data (possibly replicated and averaged) for the states $\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)$. We have prior knowledge of A: all entries are nonnegative and certain entries are zero.

Frame the problem as follows:

- $A\mathbf{n}(k) = \mathbf{n}(k+1),\ k = 1, \ldots, s.$
- Set $M = [\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s)]$ and $P = [\mathbf{n}(2), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)]$.
- Recast problem as $AM = P = I_s AN$.
- Tensor bookkeeping: $\text{vec}(I_s AM) = \left(M^T \otimes I_s\right) \text{vec}(A) = \text{vec}(P) \equiv \mathbf{d}.$
- Delete zero variables from $\text{vec}(A)$ and columns of $M^T \otimes I_s$.
- Rename resulting vectors, matrix as $\mathbf{m}, G$ and rhs $\mathbf{d}$. System takes form $G\mathbf{m} = \mathbf{d}$, with G $p \times q$, typically $p > q$.

A Generic Example

Postulate $s \times s$ projection matrix A for a stage structured population, together with data (possibly replicated and averaged) for the states $\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)$. We have prior knowledge of A: all entries are nonnegative and certain entries are zero.

Frame the problem as follows:

- $A\mathbf{n}(k) = \mathbf{n}(k+1),\ k = 1, \ldots, s.$
- Set $M = [\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s)]$ and $P = [\mathbf{n}(2), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)]$.
- Recast problem as $AM = P = I_s AN$.
- Tensor bookkeeping:
  $\mathrm{vec}(I_s AM) = \left(M^T \otimes I_s\right) \mathrm{vec}(A) = \mathrm{vec}(P) \equiv \mathbf{d}$.
- Delete zero variables from $\mathrm{vec}(A)$ and columns of $M^T \otimes I_s$.
- Rename resulting vectors, matrix as $\mathbf{m}, G$ and rhs $\mathbf{d}$. System takes form $G\mathbf{m} = \mathbf{d}$, with G $p \times q$, typically $p > q$.

A Generic Example

Postulate $s \times s$ projection matrix A for a stage structured population, together with data (possibly replicated and averaged) for the states $\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)$. We have prior knowledge of A: all entries are nonnegative and certain entries are zero.

Frame the problem as follows:

- $A\mathbf{n}(k) = \mathbf{n}(k+1)$, $k = 1, \ldots, s$.
- Set $M = [\mathbf{n}(1), \mathbf{n}(2), \ldots, \mathbf{n}(s)]$ and $P = [\mathbf{n}(2), \mathbf{n}(2), \ldots, \mathbf{n}(s+1)]$.
- Recast problem as $AM = P = I_s AN$.
- Tensor bookkeeping:
  $\mathrm{vec}(I_s AM) = \left(M^T \otimes I_s\right) \mathrm{vec}(A) = \mathrm{vec}(P) \equiv \mathbf{d}$.
- Delete zero variables from $\mathrm{vec}(A)$ and columns of $M^T \otimes I_s$.
- Rename resulting vectors, matrix as $\mathbf{m}, G$ and rhs $\mathbf{d}$. System takes form $G\mathbf{m} = \mathbf{d}$, with G $p \times q$, typically $p > q$.

Taken from Caswell's text, in turn from a referenced paper that I can't find by Kaplan and Caswell-Chen: the sugarbeet nematode Heterodera schachtii has five stages (eggs, juvenile J2, J3, J4 and adult.) Following data is density of nematodes (per 60cc of soil) for stages J2, J3+J4, adult, averaged over four replicates, measured every two days:

$t=0$	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
5.32	0.33	2.41	2.06	1.70	3.16
24.84	18.16	17.14	3.25	2.08	11.23
115.50	167.16	159.25	112.87	132.62	149.62

This population leads to a population projection matrix

$$A = \begin{bmatrix} P_1 & 0 & F_3 \\ G_1 & P_2 & \\ & G_2 & P_3 \end{bmatrix}$$

Least Squares (?)

What we do:

Of course, with much data we will almost certainly have an inconsistent system $G\mathbf{m} = \mathbf{d}$. The problem is therefore ill-posed.

- Recouch the (probably) ill-posed problem $G\mathbf{m} = \mathbf{d}$ as the optimization problem

$$\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2 .$$

- This is equivalent to solving the normal equations

$$G^T G\mathbf{m} = G^T \mathbf{d}$$

which, ASSUMING G has full column rank, has a unique solution $\mathbf{m}^*$.

What we do:

Of course, with much data we will almost certainly have an inconsistent system $G\mathbf{m} = \mathbf{d}$. The problem is therefore ill-posed.

- Recouch the (probably) ill-posed problem $G\mathbf{m} = \mathbf{d}$ as the optimization problem

$$\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2 .$$

- This is equivalent to solving the normal equations

$$G^T G\mathbf{m} = G^T\mathbf{d}$$

which, ASSUMING G has full column rank, has a unique solution $\mathbf{m}^*$.

What we do:

Of course, with much data we will almost certainly have an inconsistent system $G\mathbf{m} = \mathbf{d}$. The problem is therefore ill-posed.

- Recouch the (probably) ill-posed problem $G\mathbf{m} = \mathbf{d}$ as the optimization problem

$$\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2 .$$

- This is equivalent to solving the normal equations

$$G^T G\mathbf{m} = G^T\mathbf{d}$$

which, ASSUMING G has full column rank, has a unique solution $\mathbf{m}^*$.

Least Squares and Working Example

More Least Squares:

- Least squares has many pleasant statistical properties, e.g., if the data errors are i.i.d. normal r.v.'s, then entries of $\mathbf{m}^*$ are normally distributed and $E[\mathbf{m}^*] = \mathbf{m}_{true}$, where $G\mathbf{m}_{true} = \mathbf{d}_{true}$.
- Given that the variance of data error is σ^2, one can form the chi-square statistic

$$\chi^2_{obs} = \|G\mathbf{m} - \mathbf{d}\|_2^2 / \sigma^2$$

  and this turns out to be a r.v. with a χ^2 distribution with $m - n$ (row number of G minus column number) degrees of freedom.
- The probability of obtaining a χ^2 value as large or larger than the observed number is the p-value $p = \int_{\chi^2_{obs}}^{\infty} f_{\chi^2}(x)\,dx$ which is a uniformly distributed r.v.

More Least Squares:

- Least squares has many pleasant statistical properties, e.g., if the data errors are i.i.d. normal r.v.'s, then entries of $\mathbf{m}^*$ are normally distributed and $E[\mathbf{m}^*] = \mathbf{m}_{true}$, where $G\mathbf{m}_{true} = \mathbf{d}_{true}$.
- Given that the variance of data error is σ^2, one can form the chi-square statistic

$$\chi^2_{obs} = \|G\mathbf{m} - \mathbf{d}\|_2^2 / \sigma^2$$

and this turns out to be a r.v. with a χ^2 distribution with $m - n$ (row number of G minus column number) degrees of freedom.
- The probability of obtaining a χ^2 value as large or larger than the observed number is the p-value $p = \int_{\chi^2_{obs}}^{\infty} f_{\chi^2}(x)\, dx$ which is a uniformly distributed r.v.

More Least Squares:

- Least squares has many pleasant statistical properties, e.g., if the data errors are i.i.d. normal r.v.'s, then entries of $\mathbf{m}^*$ are normally distributed and $E[\mathbf{m}^*] = \mathbf{m}_{true}$, where $G\mathbf{m}_{true} = \mathbf{d}_{true}$.
- Given that the variance of data error is σ^2, one can form the chi-square statistic

$$\chi^2_{obs} = \|G\mathbf{m} - \mathbf{d}\|_2^2 / \sigma^2$$

  and this turns out to be a r.v. with a χ^2 distribution with $m - n$ (row number of G minus column number) degrees of freedom.
- The probability of obtaining a χ^2 value as large or larger than the observed number is the p-value $p = \int_{\chi^2_{obs}}^{\infty} f_{\chi^2}(x)\, dx$ which is a uniformly distributed r.v.

More Least Squares:

- Least squares has many pleasant statistical properties, e.g., if the data errors are i.i.d. normal r.v.'s, then entries of $\mathbf{m}^*$ are normally distributed and $E[\mathbf{m}^*] = \mathbf{m}_{true}$, where $G\mathbf{m}_{true} = \mathbf{d}_{true}$.
- Given that the variance of data error is σ^2, one can form the chi-square statistic

$$\chi^2_{obs} = \|G\mathbf{m} - \mathbf{d}\|_2^2 / \sigma^2$$

  and this turns out to be a r.v. with a χ^2 distribution with $m - n$ (row number of G minus column number) degrees of freedom.
- The probability of obtaining a χ^2 value as large or larger than the observed number is the p-value $p = \int_{\chi^2_{obs}}^{\infty} f_{\chi^2}(x)\, dx$ which is a uniformly distributed r.v.

Wood's Quadratic Programming Method

Why cast aspersions on least squares? Inter alia, the good features don't apply: take a look at G. Moreover, it may result in negative entries in the projection matrix!

Fix the Negativity:

- Recast the problem as constrained optimization problem:

$$\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2$$

subject to constraints $C\mathbf{m} \geq \mathbf{b}$ where the constaints ensure conditions like $\mathbf{m} \geq \mathbf{0}$ and $P_i + G_i \leq 1$.

- Solving a least squares problem only by adding constraints is one kind of regularization strategy. Sometimes it works well, but there are examples where it's awful.

Now let's run the script WorkingExample.m.

Wood's Quadratic Programming Method

Why cast aspersions on least squares? Inter alia, the good features don't apply: take a look at G. Moreover, it may result in negative entries in the projection matrix!

Fix the Negativity:

- Recast the problem as constrained optimization problem:

$$\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2$$

  subject to constraints $C\mathbf{m} \geq \mathbf{b}$ where the constaints ensure conditions like $\mathbf{m} \geq \mathbf{0}$ and $P_i + G_i \leq 1$.
- Solving a least squares problem only by adding constraints is one kind of regularization strategy. Sometimes it works well, but there are examples where it's awful.

Now let's run the script WorkingExample.m.

Wood's Quadratic Programming Method

Why cast aspersions on least squares? Inter alia, the good features don't apply: take a look at G. Moreover, it may result in negative entries in the projection matrix!

Fix the Negativity:

- Recast the problem as constrained optimization problem:

$$\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2$$

  subject to constraints $C\mathbf{m} \geq \mathbf{b}$ where the constaints ensure conditions like $\mathbf{m} \geq \mathbf{0}$ and $P_i + G_i \leq 1$.
- Solving a least squares problem only by adding constraints is one kind of regularization strategy. Sometimes it works well, but there are examples where it's awful.

Now let's run the script WorkingExample.m.

Regularized Least Squares

Suppose that our problem had been severely poorly conditioned or even rank deficient.

What to do?

Now even the problem $\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2$ gets us into trouble, with or without constraints.

- Tikhonov regularization: add a regularizing term that makes the problem well posed:

$$\min \|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \|L(\mathbf{m} - \mathbf{m}_0)\|_2^2 .$$

- Here α has to be chosen and L is a "smoothing" matrix like $L = I$ (zeroth order Tikhonov regularization) or matrices which mimic discretized first or second derivatives (higher order regularization. There's a Bayesian flavor here, esp. if $\mathbf{m}_0 \neq \mathbf{0}$.)
- Note: statisticians are somewhat wary of this regularization as that it introduces bias into model estimates.

Suppose that our problem had been severely poorly conditioned or even rank deficient.

What to do?

Now even the problem $\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2$ gets us into trouble, with or without constraints.

- Tikhonov regularization: add a regularizing term that makes the problem well posed:

$$\min \|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \|L(\mathbf{m} - \mathbf{m}_0)\|_2^2 .$$

- Here α has to be chosen and L is a "smoothing" matrix like $L = I$ (zeroth order Tikhonov regularization) or matrices which mimic discretized first or second derivatives (higher order regularization. There's a Bayesian flavor here, esp. if $\mathbf{m}_0 \neq \mathbf{0}$.)
- Note: statisticians are somewhat wary of this regularization as that it introduces bias into model estimates.

Suppose that our problem had been severely poorly conditioned or even rank deficient.

What to do?

Now even the problem $\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2$ gets us into trouble, with or without constraints.

- Tikhonov regularization: add a regularizing term that makes the problem well posed:

$$\min \|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \|L(\mathbf{m} - \mathbf{m}_0)\|_2^2 .$$

- Here α has to be chosen and L is a "smoothing" matrix like $L = I$ (zeroth order Tikhonov regularization) or matrices which mimic discretized first or second derivatives (higher order regularization. There's a Bayesian flavor here, esp. if $\mathbf{m}_0 \neq \mathbf{0}$.)
- Note: statisticians are somewhat wary of this regularization as that it introduces bias into model estimates.

Suppose that our problem had been severely poorly conditioned or even rank deficient.

What to do?

Now even the problem $\min_{\mathbf{m}} \|G\mathbf{m} - \mathbf{d}\|_2^2$ gets us into trouble, with or without constraints.

- Tikhonov regularization: add a regularizing term that makes the problem well posed:

$$\min \|G\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \|L(\mathbf{m} - \mathbf{m}_0)\|_2^2 .$$

- Here α has to be chosen and L is a "smoothing" matrix like $L = I$ (zeroth order Tikhonov regularization) or matrices which mimic discretized first or second derivatives (higher order regularization. There's a Bayesian flavor here, esp. if $\mathbf{m}_0 \neq \mathbf{0}$.)
- Note: statisticians are somewhat wary of this regularization as that it introduces bias into model estimates.

Determination of α

To choose an α:

Some of the principal options:

- The L-curve: do a loglog plot of $\|G\mathbf{m}_\alpha - \mathbf{d}\|_2^2$ vs $\|L\mathbf{m}_\alpha\|_2^2$ and look for the α that gives a "corner" value that balances these two terms.
- (Morozov's discrepancy principle) Choose α so that the misfit $\|G\mathbf{m}_\alpha - \mathbf{d}\|_2$ is the same size as the data noise $\|\delta\mathbf{d}\|_2$
- GCV (comes from statistical "leave-one-out" cross validation): Leave out one data point and use model to predict it. Sum these up and choose regularization parameter α that minimizes the sum of the squares of the predictive errors

$$V_0(\alpha) = \frac{1}{m}\sum_{k=1}^{m}\left(\left(Gm_{\alpha,L}^{[k]}\right)_k - d_k\right)^2.$$

To choose an α:

Some of the principal options:

- The L-curve: do a loglog plot of $\|G\mathbf{m}_\alpha - \mathbf{d}\|_2^2$ vs $\|L\mathbf{m}_\alpha\|_2^2$ and look for the α that gives a "corner" value that balances these two terms.
- (Morozov's discrepancy principle) Choose α so that the misfit $\|G\mathbf{m}_\alpha - \mathbf{d}\|_2$ is the same size as the data noise $\|\delta\mathbf{d}\|_2$
- GCV (comes from statistical "leave-one-out" cross validation): Leave out one data point and use model to predict it. Sum these up and choose regularization parameter α that minimizes the sum of the squares of the predictive errors

$$V_0(\alpha) = \frac{1}{m}\sum_{k=1}^{m}\left(\left(Gm_{\alpha,L}^{[k]}\right)_k - d_k\right)^2.$$

To choose an α:

Some of the principal options:

- The L-curve: do a loglog plot of $\|G\mathbf{m}_\alpha - \mathbf{d}\|_2^2$ vs $\|L\mathbf{m}_\alpha\|_2^2$ and look for the α that gives a "corner" value that balances these two terms.
- (Morozov's discrepancy principle) Choose α so that the misfit $\|G\mathbf{m}_\alpha - \mathbf{d}\|_2$ is the same size as the data noise $\|\delta\mathbf{d}\|_2$
- GCV (comes from statistical "leave-one-out" cross validation): Leave out one data point and use model to predict it. Sum these up and choose regularization parameter α that minimizes the sum of the squares of the predictive errors

$$V_0(\alpha) = \frac{1}{m}\sum_{k=1}^{m}\left(\left(Gm_{\alpha,L}^{[k]}\right)_k - d_k\right)^2.$$

To choose an α:

Some of the principal options:

- The L-curve: do a loglog plot of $\|G\mathbf{m}_\alpha - \mathbf{d}\|_2^2$ vs $\|L\mathbf{m}_\alpha\|_2^2$ and look for the α that gives a "corner" value that balances these two terms.
- (Morozov's discrepancy principle) Choose α so that the misfit $\|G\mathbf{m}_\alpha - \mathbf{d}\|_2$ is the same size as the data noise $\|\delta\mathbf{d}\|_2$
- GCV (comes from statistical "leave-one-out" cross validation): Leave out one data point and use model to predict it. Sum these up and choose regularization parameter α that minimizes the sum of the squares of the predictive errors

$$V_0(\alpha) = \frac{1}{m}\sum_{k=1}^{m}\left(\left(Gm_{\alpha,L}^{[k]}\right)_k - d_k\right)^2.$$

Other Regularization Methods

- **Total least squares**: this method attempts to account for the error in the coefficient matrix as well as right hand side. If constraints are not an issue, this method is preferable to least squares and has some good statistical properties more favorable than ordinary least squares.
- Maximum likelihood approach: introduce a stochastic component into the model

$$\mathbf{n}(t+1) = \exp(D(t)) A\mathbf{n}(t)$$

  where $D(t)$ is a diagonal matrix with a multivariate normal distribution of mean zero and covariance matrix Σ. Let $\mathbf{p}$ be the vector of parameters to be estimated and use the observed data to obtain maximum likelihood estimates of $\mathbf{p}$ and Σ.
- And, of course, there are infinitely many other statistical methods for point estimates of individual parameters....

Other Regularization Methods

- **Total least squares**: this method attempts to account for the error in the coefficient matrix as well as right hand side. If constraints are not an issue, this method is preferable to least squares and has some good statistical properties more favorable than ordinary least squares.
- Maximum likelihood approach: introduce a stochastic component into the model

$$\mathbf{n}\left(t+1\right) = \exp\left(D\left(t\right)\right) A\mathbf{n}\left(t\right)$$

  where $D\left(t\right)$ is a diagonal matrix with a multivariate normal distribution of mean zero and covariance matrix Σ. Let $\mathbf{p}$ be the vector of parameters to be estimated and use the observed data to obtain maximum likelihood estimates of $\mathbf{p}$ and Σ.
- And, of course, there are infinitely many other statistical methods for point estimates of individual parameters....

Other Regularization Methods

- **Total least squares**: this method attempts to account for the error in the coefficient matrix as well as right hand side. If constraints are not an issue, this method is preferable to least squares and has some good statistical properties more favorable than ordinary least squares.
- Maximum likelihood approach: introduce a stochastic component into the model

$$\mathbf{n}\left(t+1\right)=\exp\left(D\left(t\right)\right)A\mathbf{n}\left(t\right)$$

  where $D\left(t\right)$ is a diagonal matrix with a multivariate normal distribution of mean zero and covariance matrix Σ. Let $\mathbf{p}$ be the vector of parameters to be estimated and use the observed data to obtain maximum likelihood estimates of $\mathbf{p}$ and Σ.
- And, of course, there are infinitely many other statistical methods for point estimates of individual parameters....

Summary

- Inverse problems arising from parameter recovery in Lefkowitz models are ill posed, but can be managed by tools of inverse theory such as least squares, Tikhonov regularization and constrained optimization.
- There are some interesting data in the literature relating to freshwater turtles that seem to exploit purely statistical methods. I plan to explore Lefkovitch modeling in this context. Regularization tools may offer new insights, particularly in modeling that leads to rank deficient problems.
- Specifically, one might try to push the envelope with a non-stationary projection matrix. Or nonlinear one. Or tackle unknown reproductive rates. These will likely give problems with worse conditioned that our working example.
- The role of total least squares seems to be largely unexplored for these problems. This warrants further investigation.

Summary

- Inverse problems arising from parameter recovery in Lefkowitz models are ill posed, but can be managed by tools of inverse theory such as least squares, Tikhonov regularization and constrained optimization.
- There are some interesting data in the literature relating to freshwater turtles that seem to exploit purely statistical methods. I plan to explore Lefkovitch modeling in this context. Regularization tools may offer new insights, particularly in modeling that leads to rank deficient problems.
- Specifically, one might try to push the envelope with a non-stationary projection matrix. Or nonlinear one. Or tackle unknown reproductive rates. These will likely give problems with worse conditioned that our working example.
- The role of total least squares seems to be largely unexplored for these problems. This warrants further investigation.

Summary

- Inverse problems arising from parameter recovery in Lefkowitz models are ill posed, but can be managed by tools of inverse theory such as least squares, Tikhonov regularization and constrained optimization.
- There are some interesting data in the literature relating to freshwater turtles that seem to exploit purely statistical methods. I plan to explore Lefkovitch modeling in this context. Regularization tools may offer new insights, particularly in modeling that leads to rank deficient problems.
- Specifically, one might try to push the envelope with a non-stationary projection matrix. Or nonlinear one. Or tackle unknown reproductive rates. These will likely give problems with worse conditioned that our working example.
- The role of total least squares seems to be largely unexplored for these problems. This warrants further investigation.

Summary

- Inverse problems arising from parameter recovery in Lefkowitz models are ill posed, but can be managed by tools of inverse theory such as least squares, Tikhonov regularization and constrained optimization.
- There are some interesting data in the literature relating to freshwater turtles that seem to exploit purely statistical methods. I plan to explore Lefkovitch modeling in this context. Regularization tools may offer new insights, particularly in modeling that leads to rank deficient problems.
- Specifically, one might try to push the envelope with a non-stationary projection matrix. Or nonlinear one. Or tackle unknown reproductive rates. These will likely give problems with worse conditioned that our working example.
- The role of total least squares seems to be largely unexplored for these problems. This warrants further investigation.