Experimental Design for Efficient Identification of Gene Regulatory Networks using Sparse Bayesian Models

Florian Steinke, Matthias Seeger, Koji Tsuda

Max Planck Institute for Biological Cybernetics
Tübingen, Germany

26 July 2007

MAX-PLANCK-GESELLSCHAFT

Outline

1. The Need for Experimental Design
2. System Identification of Genetic Regulation
3. Sparse Bayesian Linear Model
4. Experiments
5. Related Work. Conclusions

Why Experimental Design?

- Large-scale genome-wide experiments:
  Affordable today in fully automatized labs
- Solve problems by complete enumeration or random shooting?
  - Guaranteed to run out of steam on hard problems
  - Cutting-edge experiments always hard/expensive
  - Even for large labs: $(\#\mathrm{Results})/\$$ counts!
- Sequential Optimal Design
  Plan next experiment based on all previous outcomes
  ⇒ Every smart biologist does that anyway!
- Can optimal design be semi-automatized on a dumb machine?
  What general framework allows us to do that?

Why Experimental Design?

- Large-scale genome-wide experiments:
  Affordable today in fully automatized labs
- Solve problems by complete enumeration or random shooting?
  - Guaranteed to run out of steam on hard problems
  - Cutting-edge experiments always hard/expensive
  - Even for large labs: $(\#\text{Results})/\$$ counts!
- Sequential Optimal Design
  Plan next experiment based on all previous outcomes
  $\Rightarrow$ Every smart biologist does that anyway!
- Can optimal design be semi-automatized on a dumb machine?
  What general framework allows us to do that?

Why Experimental Design?

- Large-scale genome-wide experiments:
  Affordable today in fully automatized labs
- Solve problems by complete enumeration or random shooting?
  - Guaranteed to run out of steam on hard problems
  - Cutting-edge experiments always hard/expensive
  - Even for large labs: $(\#\mathrm{Results})/\$$ counts!
- Sequential Optimal Design
  Plan next experiment based on all previous outcomes
  $\Rightarrow$ Every smart biologist does that anyway!
- Can optimal design be semi-automatized on a dumb machine?
  What general framework allows us to do that?

Why Experimental Design?

- Large-scale genome-wide experiments:
  Affordable today in fully automatized labs
- Solve problems by complete enumeration or random shooting?
  - Guaranteed to run out of steam on hard problems
  - Cutting-edge experiments always hard/expensive
  - Even for large labs: $(\#\text{Results})/\$$ counts!
- Sequential Optimal Design
  Plan next experiment based on all previous outcomes
  $\Rightarrow$ Every smart biologist does that anyway!
- Can optimal design be semi-automatized on a dumb machine?
  What general framework allows us to do that?

Bayesian Optimal Design

Smart Biologist

- Which variables could explain my data? How could dependencies look like?
- X look well-determined. Did not learn much about Y
- I think: Exp. A (B) would tell me more about X (Y) now $\Rightarrow$ Of course I do B!
- 1000s of X, Y. Combinatorial number of possible interactions $\Rightarrow$ Human intuition

Bayesian Framework

- Model design
  Observed, hidden variables. Dependency model
- Posterior uncertainty
  Reduced on X, but not on Y
- Information Gain Scores
  $S(A; \mathrm{Data}) < S(B; \mathrm{Data})$
  $\Rightarrow$ `OK> Should do B`
- Run overnight, sift through raw data, (hopefully) help intuition along

Bayesian Optimal Design

Smart Biologist

- Which variables could explain my data? How could dependencies look like?
- X look well-determined.
  Did not learn much about Y
- I think: Exp. A (B) would tell me more about X (Y) now
  ⇒ Of course I do B!
- 1000s of X, Y. Combinatorial number of possible interactions ⇒ Human intuition

Bayesian Framework

- Model design
  Observed, hidden variables.
  Dependency model
- Posterior uncertainty
  Reduced on X, but not on Y
- Information Gain Scores
  $S(A; \mathrm{Data}) < S(B; \mathrm{Data})$
  ⇒ `OK> Should do B`
- Run overnight, sift through raw data, (hopefully) help intuition along

Bayesian Optimal Design

Smart Biologist

- Which variables could explain my data? How could dependencies look like?
- X look well-determined. Did not learn much about Y
- I think: Exp. A (B) would tell me more about X (Y) now ⇒ Of course I do B!
- 1000s of X, Y. Combinatorial number of possible interactions ⇒ Human intuition

Bayesian Framework

- Model design
  Observed, hidden variables. Dependency model
- Posterior uncertainty
  Reduced on X, but not on Y
- Information Gain Scores
  $S(A; \mathrm{Data}) < S(B; \mathrm{Data})$
  ⇒ `OK> Should do B`
- Run overnight, sift through raw data, (hopefully) help intuition along

Bayesian Optimal Design

Smart Biologist

- Which variables could explain my data? How could dependencies look like?
- X look well-determined.
  Did not learn much about Y
- I think: Exp. A (B) would tell me more about X (Y) now
  ⇒ Of course I do B!
- 1000s of X, Y. Combinatorial number of possible interactions ⇒ Human intuition

Bayesian Framework

- Model design
  Observed, hidden variables. Dependency model
- Posterior uncertainty
  Reduced on X, but not on Y
- Information Gain Scores
  $S(A; \mathrm{Data}) < S(B; \mathrm{Data})$
  ⇒ `OK> Should do B`
- Run overnight, sift through raw data, (hopefully) help intuition along

Gene Regulatory Networks

- Genes can regulate other genes
  Protein from gene A can be transcription factor:
  up-/down-regulates transcription of gene B.
  Causal link $A \rightarrow B$ in gene regulatory network
- Affordable Measurements
  m-RNA concentrations (micro-arrays), protein concentrations
  $\leftrightarrow$ Expression levels $x_A(t)$, $x_B(t)$
- System Identification
  Interventionist. Disturb system (without breaking it).
  Learn structure from changes in measurements
- Optimal Experimental Design
  For given model: Short(est) sequences of experiments leading to identification?

Gene Regulatory Networks

- Genes can regulate other genes
  Protein from gene A can be transcription factor:
  up-/down-regulates transcription of gene B.
  Causal link $A \longrightarrow B$ in gene regulatory network
- Affordable Measurements
  m-RNA concentrations (micro-arrays), protein concentrations
  $\longleftrightarrow$ Expression levels $x_A(t)$, $x_B(t)$
- System Identification
  Interventionist. Disturb system (without breaking it).
  Learn structure from changes in measurements
- Optimal Experimental Design
  For given model: Short(est) sequences of experiments leading to identification?

Gene Regulatory Networks

- Genes can regulate other genes
  Protein from gene A can be transcription factor:
  up-/down-regulates transcription of gene B.
  Causal link $A \rightarrow B$ in gene regulatory network
- Affordable Measurements
  m-RNA concentrations (micro-arrays), protein concentrations
  $\leftrightarrow$ Expression levels $x_A(t)$, $x_B(t)$
- System Identification
  Interventionist. Disturb system (without breaking it).
  Learn structure from changes in measurements
- Optimal Experimental Design
  For given model: Short(est) sequences of experiments leading to identification?

Gene Regulatory Networks

- Genes can regulate other genes
  Protein from gene A can be transcription factor:
  up-/down-regulates transcription of gene B.
  Causal link $A \rightarrow B$ in gene regulatory network
- Affordable Measurements
  m-RNA concentrations (micro-arrays), protein concentrations
  $\leftrightarrow$ Expression levels $x_A(t)$, $x_B(t)$
- System Identification
  Interventionist. Disturb system (without breaking it).
  Learn structure from changes in measurements
- Optimal Experimental Design
  For given model: Short(est) sequences of experiments leading to identification?

Linearized ODE Model

ODE Model

$$d\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t))dt + d\boldsymbol{W}(t)$$
$$\mathrm{E}[\boldsymbol{x}(t)] \to \boldsymbol{x}_0 \ (t \to \infty)$$

$\boldsymbol{x}(t)$	Expression levels n genes
$\boldsymbol{f}(\cdot)$	Non-linear model
$\boldsymbol{x}_0$	Unperturbed steady state

1. Linearize around steady state: $\boldsymbol{x}(t) \to \boldsymbol{x}(t) - \boldsymbol{x}_0$.
   System matrix $\boldsymbol{A} = (df_i/dx_{0,j})_{ij}$
2. Disturb system by $\boldsymbol{u}(t) \equiv \boldsymbol{u}_*$, measure new steady state:

$$d\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{x}(t) - \boldsymbol{u}(t) + d\boldsymbol{W}(t), \quad \boldsymbol{x}_* = \lim_{t\to\infty} \mathrm{E}[\boldsymbol{x}(t)]$$

3. Motivates linear model for measurements:

$$\boldsymbol{u}_* = \boldsymbol{A}\boldsymbol{x}_* + \varepsilon, \quad \varepsilon \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$

Linearized ODE Model

ODE Model

$$dx(t) = f(x(t))dt + dW(t)$$
$$\mathrm{E}[x(t)] \to x_0 \ (t \to \infty)$$

$x(t)$	Expression levels n genes
$f(\cdot)$	Non-linear model
x_0	Unperturbed steady state

1. Linearize around steady state: $x(t) \to x(t) - x_0$.
   System matrix $A = (df_i/dx_{0,j})_{ij}$
2. Disturb system by $u(t) \equiv u_*$, measure new steady state:

$$dx(t) = Ax(t) - u(t) + dW(t), \quad x_* = \lim_{t\to\infty} \mathrm{E}[x(t)]$$

3. Motivates linear model for measurements:

$$u_* = Ax_* + \varepsilon, \quad \varepsilon \sim N(\mathbf{0}, \sigma^2 I)$$

Linearized ODE Model

ODE Model

$$d\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t))dt + d\boldsymbol{W}(t)$$
$$\mathrm{E}[\boldsymbol{x}(t)] \to \boldsymbol{x}_0 \ (t \to \infty)$$

$\boldsymbol{x}(t)$	Expression levels n genes
$\boldsymbol{f}(\cdot)$	Non-linear model
$\boldsymbol{x}_0$	Unperturbed steady state

1. Linearize around steady state: $\boldsymbol{x}(t) \to \boldsymbol{x}(t) - \boldsymbol{x}_0$.
   System matrix $\boldsymbol{A} = (df_i/dx_{0,j})_{ij}$
2. Disturb system by $\boldsymbol{u}(t) \equiv \boldsymbol{u}_*$, measure new steady state:

$$d\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{x}(t) - \boldsymbol{u}(t) + d\boldsymbol{W}(t), \quad \boldsymbol{x}_* = \lim_{t\to\infty} \mathrm{E}[\boldsymbol{x}(t)]$$

3. Motivates linear model for measurements:

$$\boldsymbol{u}_* = \boldsymbol{A}\boldsymbol{x}_* + \varepsilon, \quad \varepsilon \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$

Linearized ODE Model

ODE Model

$$dx(t) = f(x(t))dt + dW(t)$$
$$\mathrm{E}[x(t)] \to x_0 \ (t \to \infty)$$

$x(t)$ Expression levels n genes

$f(\cdot)$ Non-linear model

x_0 Unperturbed steady state

1. Linearize around steady state: $x(t) \to x(t) - x_0$.
   System matrix $A = (df_i/dx_{0,j})_{ij}$
2. Disturb system by $u(t) \equiv u_*$, measure new steady state:

$$dx(t) = Ax(t) - u(t) + dW(t), \quad x_* = \lim_{t\to\infty} \mathrm{E}[x(t)]$$

3. Motivates linear model for measurements:

$$u_* = Ax_* + \varepsilon, \quad \varepsilon \sim N(\mathbf{0}, \sigma^2 I)$$

Bayesian Linear Model

- Likelihood $P(D|\boldsymbol{A}) = \prod_k N(\boldsymbol{u}_k|\boldsymbol{A}\boldsymbol{x}_k, \sigma^2\boldsymbol{I})$. Prior $P(\boldsymbol{A})$

$$\text{Bayesian Posterior :} \quad P(\boldsymbol{A}|D) \propto P(D|\boldsymbol{A})P(\boldsymbol{A})$$

Why not just (penalized) maximum likelihood estimation:

$$\hat{\boldsymbol{A}} = \operatorname{argmax} P(D|\boldsymbol{A})P(\boldsymbol{A}) \ ?$$

- Estimation is not sufficient here
  Optimal design fundamentally needs uncertainty quantification
  $\Rightarrow$ Posterior $P(\boldsymbol{A}|D)$ is just that
- Decisions are needed after many fewer than n experiments.
  $\Rightarrow$ "Objective" classical estimation theory breaks down
- Besides: Is $\boldsymbol{A}$ really completely unknown . . . ?

Bayesian Linear Model

- Likelihood $P(D|\boldsymbol{A}) = \prod_k N(\boldsymbol{u}_k|\boldsymbol{A}\boldsymbol{x}_k, \sigma^2\boldsymbol{I})$. Prior $P(\boldsymbol{A})$

$$\text{Bayesian Posterior :} \quad P(\boldsymbol{A}|D) \propto P(D|\boldsymbol{A})P(\boldsymbol{A})$$

Why not just (penalized) maximum likelihood estimation:

$$\hat{\boldsymbol{A}} = \operatorname{argmax} P(D|\boldsymbol{A})P(\boldsymbol{A}) \text{ ?}$$

- Estimation is not sufficient here
  Optimal design fundamentally needs uncertainty quantification
  $\Rightarrow$ Posterior $P(\boldsymbol{A}|D)$ is just that
- Decisions are needed after many fewer than n experiments.
  $\Rightarrow$ "Objective" classical estimation theory breaks down
- Besides: Is $\boldsymbol{A}$ really completely unknown . . . ?

Bayesian Linear Model

- Likelihood $P(D|\boldsymbol{A}) = \prod_k N(\boldsymbol{u}_k|\boldsymbol{A}\boldsymbol{x}_k, \sigma^2\boldsymbol{I})$. Prior $P(\boldsymbol{A})$

  $$\text{Bayesian Posterior}: \quad P(\boldsymbol{A}|D) \propto P(D|\boldsymbol{A})P(\boldsymbol{A})$$

  Why not just (penalized) maximum likelihood estimation:

  $$\hat{\boldsymbol{A}} = \text{argmax}\, P(D|\boldsymbol{A})P(\boldsymbol{A}) \; ?$$

- Estimation is not sufficient here
  Optimal design fundamentally needs uncertainty quantification
  ⇒ Posterior $P(\boldsymbol{A}|D)$ is just that
- Decisions are needed after many fewer than n experiments.
  ⇒ “Objective” classical estimation theory breaks down
- Besides: Is $\boldsymbol{A}$ really completely unknown . . . ?

Bayesian Linear Model

- Likelihood $P(D|\boldsymbol{A}) = \prod_k N(\boldsymbol{u}_k|\boldsymbol{A}\boldsymbol{x}_k, \sigma^2 \boldsymbol{I})$. Prior $P(\boldsymbol{A})$

$$\text{Bayesian Posterior :} \quad P(\boldsymbol{A}|D) \propto P(D|\boldsymbol{A})P(\boldsymbol{A})$$

  Why not just (penalized) maximum likelihood estimation:

$$\hat{\boldsymbol{A}} = \text{argmax}\, P(D|\boldsymbol{A})P(\boldsymbol{A}) \ ?$$

- Estimation is not sufficient here
  Optimal design fundamentally needs uncertainty quantification
  $\Rightarrow$ Posterior $P(\boldsymbol{A}|D)$ is just that
- Decisions are needed after many fewer than n experiments.
  $\Rightarrow$ “Objective” classical estimation theory breaks down
- Besides: Is $\boldsymbol{A}$ really completely unknown . . . ?

A Sparsity Prior Distribution

- All biological regulatory networks are sparsely connected
  $\Rightarrow$ $\boldsymbol{A}$ should have many very small entries
- Encoding sparsity of $\boldsymbol{A}$ is a must!
  $\Rightarrow$ Sparsity-enforcing prior distribution $P(\boldsymbol{A})$

Laplace Prior

$$P(\boldsymbol{A}) = \prod_{ij} P(a_{ij}), \quad P(a_{ij}) = \frac{\tau}{2} e^{-\tau |a_{ij}|}$$

A Sparsity Prior Distribution

- All biological regulatory networks are sparsely connected
  $\Rightarrow$ $\boldsymbol{A}$ should have many very small entries
- Encoding sparsity of $\boldsymbol{A}$ is a must!
  $\Rightarrow$ Sparsity-enforcing prior distribution $P(\boldsymbol{A})$

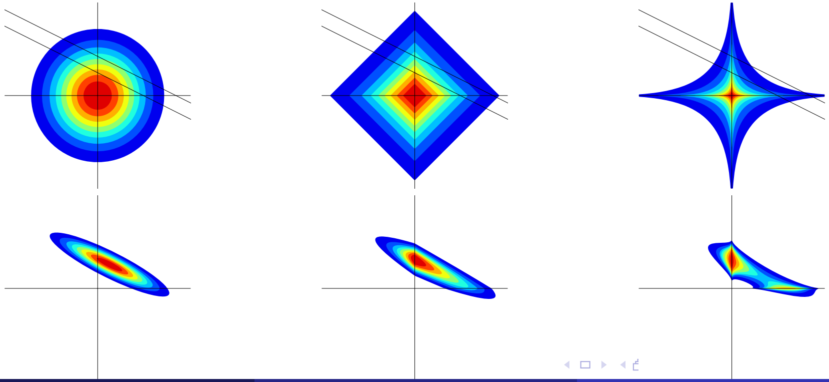
Laplace Prior

$$P(\boldsymbol{A}) = \prod_{ij} P(a_{ij}), \quad P(a_{ij}) = \frac{\tau}{2} \mathrm{e}^{-\tau |a_{ij}|}$$

A Sparsity Prior Distribution

Laplace Prior

$$P(\boldsymbol{A}) = \prod_{ij} P(a_{ij}), \quad P(a_{ij}) = \frac{\tau}{2} \mathrm{e}^{-\tau |a_{ij}|}$$



Approximate Inference: Rough Idea

- Bayesian posterior for one row $\boldsymbol{a}$ of $\boldsymbol{A}$

$$P(\boldsymbol{a}|D) \propto P(D|\boldsymbol{a}) \prod_i P(a_i)$$

Hard "just" because $P(a_i)$ are not Gaussian

- Moment matching idea: $P(D|\boldsymbol{a})P(a_i)$ not Gaussian either. Gaussian with same moments have form $P(D|\boldsymbol{a})\tilde{P}(a_i|b_i, \pi_i)$.

$$P(\boldsymbol{a}|D) \approx Q(\boldsymbol{a}) \propto P(D|\boldsymbol{a}) \prod_i \tilde{P}(a_i|b_i, \pi_i)$$

- Expectation Propagation: iterates moment matching over i: Update variational parameters b_i, π_i s.t.:

$$Q_{old}(\boldsymbol{a})P(a_i)/\tilde{P}(a_i) \longleftrightarrow Q_{new}(\boldsymbol{a}) \text{ [same moments]}$$

Approximate Inference: Rough Idea

- Bayesian posterior for one row ***a*** of ***A***

$$P(\boldsymbol{a}|D) \propto P(D|\boldsymbol{a}) \prod_i P(a_i)$$

Hard "just" because $P(a_i)$ are not Gaussian

- Moment matching idea: $P(D|\boldsymbol{a})P(a_i)$ not Gaussian either. Gaussian with same moments have form $P(D|\boldsymbol{a})\tilde{P}(a_i|b_i,\pi_i)$.

$$P(\boldsymbol{a}|D) \approx Q(\boldsymbol{a}) \propto P(D|\boldsymbol{a}) \prod_i \tilde{P}(a_i|b_i,\pi_i)$$

- Expectation Propagation: iterates moment matching over i: Update variational parameters b_i, π_i s.t.:

$$Q_{old}(\boldsymbol{a})P(a_i)/\tilde{P}(a_i) \longleftrightarrow Q_{new}(\boldsymbol{a}) \text{ [same moments]}$$

Approximate Inference: Rough Idea

- Bayesian posterior for one row $\boldsymbol{a}$ of $\boldsymbol{A}$

$$P(\boldsymbol{a}|D) \propto P(D|\boldsymbol{a}) \prod_i P(a_i)$$

Hard "just" because $P(a_i)$ are not Gaussian

- Moment matching idea: $P(D|\boldsymbol{a})P(a_i)$ not Gaussian either. Gaussian with same moments have form $P(D|\boldsymbol{a})\tilde{P}(a_i|b_i, \pi_i)$.

$$P(\boldsymbol{a}|D) \approx Q(\boldsymbol{a}) \propto P(D|\boldsymbol{a}) \prod_i \tilde{P}(a_i|b_i, \pi_i)$$

- Expectation Propagation: iterates moment matching over i: Update variational parameters b_i, π_i s.t.:

$$Q_{old}(\boldsymbol{a})P(a_i)/\tilde{P}(a_i) \longleftrightarrow Q_{new}(\boldsymbol{a}) \text{ [same moments]}$$

Bayesian Experimental Design

Information Gain Score

$$S(\boldsymbol{u}_*, \boldsymbol{x}_* | D) = \mathrm{D}[Q'(\boldsymbol{A} | D \cup \{(\boldsymbol{u}_*, \boldsymbol{x}_*)\}) \,\|\, Q(\boldsymbol{A} | D)]$$

$\mathrm{D}[Q' \,\|\, Q]$: Information gained in $Q \to Q'$.
Efficient exact computation for Gaussians Q, Q'

- But outcome $\boldsymbol{x}_*$ unknown before experiment $\boldsymbol{u}_*$ done!?
  $\Rightarrow$ Use expected score under current knowledge $Q(\boldsymbol{x}_* | D, \boldsymbol{u}_*)$.
  Exact sampling: $\boldsymbol{A} \sim Q(\cdot | D)$, $\boldsymbol{x}_* = \boldsymbol{A}^{-1}\boldsymbol{u}_*$
- Score many candidates $\boldsymbol{u}_*$ very efficiently:
  Pick maximizer of $\mathrm{E}_{Q(\boldsymbol{x}_* | D, \boldsymbol{u}_*)}[S(\boldsymbol{u}_*, \boldsymbol{x}_* | D)]$

Bayesian Experimental Design

Information Gain Score

$$S(\boldsymbol{u}_*, \boldsymbol{x}_* | D) = \mathrm{D}[Q'(\boldsymbol{A} | D \cup \{(\boldsymbol{u}_*, \boldsymbol{x}_*)\}) \,\|\, Q(\boldsymbol{A}|D)]$$

$\mathrm{D}[Q' \,\|\, Q]$: Information gained in $Q \to Q'$.
Efficient exact computation for Gaussians Q, Q'

- But outcome $\boldsymbol{x}_*$ unknown before experiment $\boldsymbol{u}_*$ done!?
  $\Rightarrow$ Use expected score under current knowledge $Q(\boldsymbol{x}_* | D, \boldsymbol{u}_*)$.
  Exact sampling: $\boldsymbol{A} \sim Q(\cdot|D)$, $\boldsymbol{x}_* = \boldsymbol{A}^{-1}\boldsymbol{u}_*$
- Score many candidates $\boldsymbol{u}_*$ very efficiently:
  Pick maximizer of $\mathrm{E}_{Q(\boldsymbol{x}_*|D,\boldsymbol{u}_*)}[S(\boldsymbol{u}_*, \boldsymbol{x}_*|D)]$

Bayesian Experimental Design

Information Gain Score

$$S(\boldsymbol{u}_*, \boldsymbol{x}_* | D) = \mathrm{D}[Q'(\boldsymbol{A} | D \cup \{(\boldsymbol{u}_*, \boldsymbol{x}_*)\}) \parallel Q(\boldsymbol{A} | D)]$$

$\mathrm{D}[Q' \parallel Q]$: Information gained in $Q \rightarrow Q'$.
Efficient exact computation for Gaussians Q, Q'

- But outcome $\boldsymbol{x}_*$ unknown before experiment $\boldsymbol{u}_*$ done!?
  $\Rightarrow$ Use expected score under current knowledge $Q(\boldsymbol{x}_* | D, \boldsymbol{u}_*)$.
  Exact sampling: $\boldsymbol{A} \sim Q(\cdot | D)$, $\boldsymbol{x}_* = \boldsymbol{A}^{-1} \boldsymbol{u}_*$
- Score many candidates $\boldsymbol{u}_*$ very efficiently:
  Pick maximizer of $\mathrm{E}_{Q(\boldsymbol{x}_* | D, \boldsymbol{u}_*)}[S(\boldsymbol{u}_*, \boldsymbol{x}_* | D)]$

Our Approach As Black Box

- Robust, efficient code will be released:
  Predictable running time. Easy to use for non-experts
- Free parameters σ^2, τ:
  Bayesian automatic selection, given related task data
- Applies to time series data just as well (if linear model does)
- Encompasses generalized linear models:
  - Non-Gaussian noise (outliers)
  - Discrete or point process observations

Our Approach As Black Box

- Robust, efficient code will be released:
  Predictable running time. Easy to use for non-experts
- Free parameters σ^2, τ:
  Bayesian automatic selection, given related task data
- Applies to time series data just as well (if linear model does)
- Encompasses generalized linear models:
  - Non-Gaussian noise (outliers)
  - Discrete or point process observations

Our Approach As Black Box

- Robust, efficient code will be released:
  Predictable running time. Easy to use for non-experts
- Free parameters σ^2, τ:
  Bayesian automatic selection, given related task data
- Applies to time series data just as well (if linear model does)
- Encompasses generalized linear models:
  - Non-Gaussian noise (outliers)
  - Discrete or point process observations

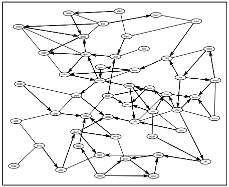
Our Approach As Black Box

- Robust, efficient code will be released:
  Predictable running time. Easy to use for non-experts
- Free parameters σ^2, τ:
  Bayesian automatic selection, given related task data
- Applies to time series data just as well (if linear model does)
- Encompasses generalized linear models:
  - Non-Gaussian noise (outliers)
  - Discrete or point process observations

Experimental Setup

Common practice: validate on data from realistic simulation.

- Sample small-world network, $n = 50$ genes



- Model with Hill-type kinetics, parameters randomly drawn (similar to Kholodenko *et.al.*, 02)

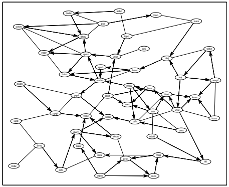
$$f_i(\boldsymbol{x}) = -V_{di}\frac{x_i}{d_i + x_i} + V_{si}\prod_{j\in\mathcal{A}_i}\frac{1 + A_{ij}\left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}{1 + \left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}\prod_{j\in\mathcal{I}_i}\frac{1}{1 + \left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}$$

- Pool of 200 $\boldsymbol{u}_*$ (unit norm; 3 non-zeros, sparsity for biological relevance) randomly drawn
- Noise variance σ^2 estimated from simpler random networks. Prior precision τ set by heuristic

Experimental Setup

Common practice: validate on data from realistic simulation.

- Sample small-world network, $n = 50$ genes



- Model with Hill-type kinetics, parameters randomly drawn (similar to Kholodenko *et.al.*, 02)

$$f_i(\boldsymbol{x}) = -V_{di}\frac{x_i}{d_i + x_i}$$

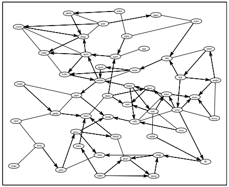
$$+ V_{si}\prod_{j\in\mathcal{A}_i}\frac{1 + A_{ij}\left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}{1 + \left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}\prod_{j\in\mathcal{I}_i}\frac{1}{1 + \left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}$$

- Pool of 200 $\boldsymbol{u}_*$ (unit norm; 3 non-zeros, sparsity for biological relevance) randomly drawn
- Noise variance σ^2 estimated from simpler random networks. Prior precision τ set by heuristic

Experimental Setup

Common practice: validate on data from realistic simulation.

- Sample small-world network, $n = 50$ genes



- Model with Hill-type kinetics, parameters randomly drawn (similar to Kholodenko *et.al.*, 02)

$$f_i(\boldsymbol{x}) = -V_{di}\frac{x_i}{d_i + x_i} + V_{si}\prod_{j\in\mathcal{A}_i}\frac{1 + A_{ij}\left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}{1 + \left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}\prod_{j\in\mathcal{I}_i}\frac{1}{1 + \left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}$$

- Pool of 200 $\boldsymbol{u}_*$ (unit norm; 3 non-zeros, sparsity for biological relevance) randomly drawn
- Noise variance σ^2 estimated from simpler random networks. Prior precision τ set by heuristic

Decision and Evaluation

- Network from joint posterior $Q(\boldsymbol{A})$?
  Rank edges $i \leftarrow j$ by $Q(\{|a_{ij}| > 0.1\})$
- ROC curve: false positive rate → true positive rate.
  iAUC: area under ROC curve, up to # FPs $=$ # edges.
  Random ranking has iAUC $= 0.02$
- About 25% edges have value ≈ 0 in true $\boldsymbol{A}$ (at steady state), not detectable by linearized model. Excluded from iAUC computation

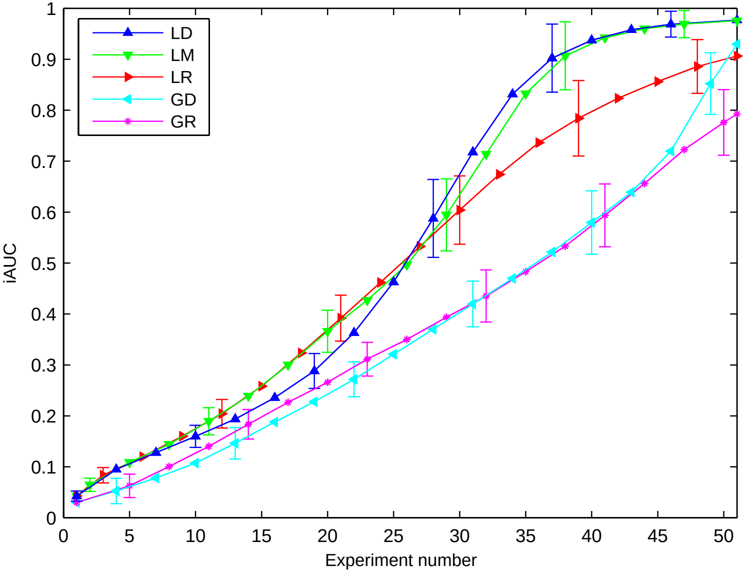
Decision and Evaluation

- Network from joint posterior $Q(\boldsymbol{A})$?
  Rank edges $i \leftarrow j$ by $Q(\{|a_{ij}| > 0.1\})$
- ROC curve: false positive rate → true positive rate.
  iAUC: area under ROC curve, up to # FPs $=$ # edges.
  Random ranking has iAUC $= 0.02$
- About 25% edges have value ≈ 0 in true $\boldsymbol{A}$ (at steady state), not detectable by linearized model. Excluded from iAUC computation

Decision and Evaluation

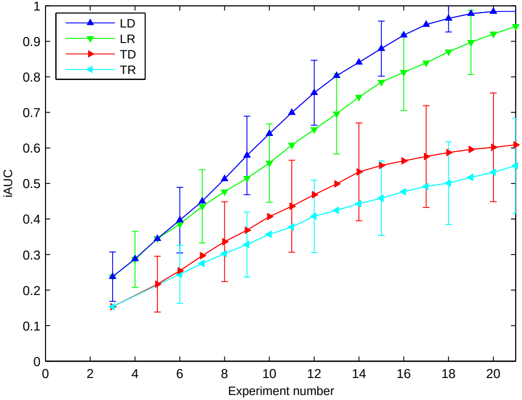
- Network from joint posterior $Q(\boldsymbol{A})$?
  Rank edges $i \leftarrow j$ by $Q(\{|a_{ij}| > 0.1\})$
- ROC curve: false positive rate → true positive rate.
  iAUC: area under ROC curve, up to # FPs $=$ # edges.
  Random ranking has iAUC $= 0.02$
- About 25% edges have value ≈ 0 in true $\boldsymbol{A}$ (at steady state), not detectable by linearized model. Excluded from iAUC computation

Results


LD
LM
LR
GD
GR
iAUC
Experiment number
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25
30
35
40
45
50

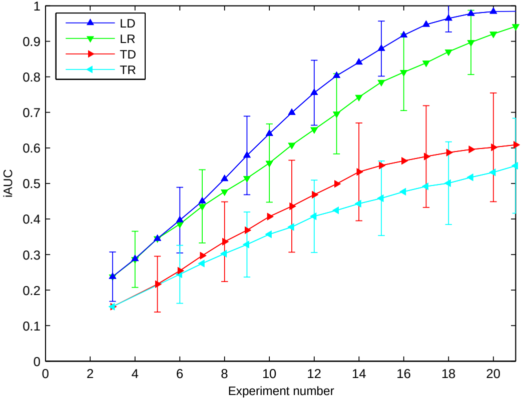
Comparison Tegnér *et.al.*

- Tegnér *et.al.*(PNAS 03): most cited work on experimental design for network identification.
- We do not use quantizations: our method works better and is 2 orders of magnitude faster
- They require node in-degree ≤ 3 (unrealistic in scale-free networks), we do not [comparison done on such graphs]



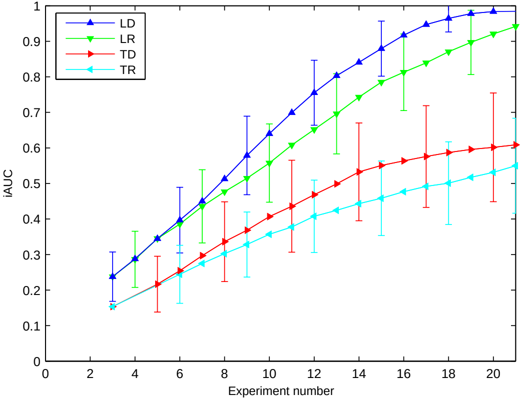
Comparison Tegnér *et.al.*

- Tegnér *et.al.*(PNAS 03): most cited work on experimental design for network identification.
- We do not use quantizations: our method works better and is 2 orders of magnitude faster
- They require node in-degree ≤ 3 (unrealistic in scale-free networks), we do not [comparison done on such graphs]



Comparison Tegnér *et.al.*

- Tegnér *et.al.*(PNAS 03): most cited work on experimental design for network identification.
- We do not use quantizations: our method works better and is 2 orders of magnitude faster
- They require node in-degree ≤ 3 (unrealistic in scale-free networks), we do not [comparison done on such graphs]



Related Work

- Much work on disturbed linearized ODE models.
  Estimation, no inference, no experimental design (except Tegnér *et.al.*)
- Sparse Bayesian Learning (Tipping, 01; Rogers, Girolami, 05)
  No experimental design. Uses non-log-concave Student-*t* prior.
  EP more general than SBL
- Markov Chain Monte Carlo (Park, Casella, 05)
  Much slower than our method (too slow for large-scale experimental design). Hard to assess convergence even for experts

Related Work

- Much work on disturbed linearized ODE models.
  Estimation, no inference, no experimental design (except Tegnér *et.al.*)
- Sparse Bayesian Learning (Tipping, 01; Rogers, Girolami, 05)
  No experimental design. Uses non-log-concave Student-*t* prior.
  EP more general than SBL
- Markov Chain Monte Carlo (Park, Casella, 05)
  Much slower than our method (too slow for large-scale experimental design). Hard to assess convergence even for experts

Related Work

- Much work on disturbed linearized ODE models.
  Estimation, no inference, no experimental design (except Tegnér *et.al.*)
- Sparse Bayesian Learning (Tipping, 01; Rogers, Girolami, 05)
  No experimental design. Uses non-log-concave Student-*t* prior.
  EP more general than SBL
- Markov Chain Monte Carlo (Park, Casella, 05)
  Much slower than our method (too slow for large-scale experimental design). Hard to assess convergence even for experts

Conclusions

- Fast accurate approximate inference, experimental design in disturbed linearized ODE setup
- Network sparsity is key prior assumption. Experimental design can lead to large savings
- Can be used with time-course measurements just as well
- Robust, easy-to-use method. Code with Matlab interface will be released
- Linearized ODE approach is limited:
  - Small, controlled u_* to stay in linearity region (experimental techniques?), but large u_* for better SNR
  - No saturation, Michaelis-Menten, *etc*
- Bayesian inference and experimental design for (simple) non-linear ODEs of biochemical kinetics?

Conclusions

- Fast accurate approximate inference, experimental design in disturbed linearized ODE setup
- Network sparsity is key prior assumption. Experimental design can lead to large savings
- Can be used with time-course measurements just as well
- Robust, easy-to-use method. Code with Matlab interface will be released
- Linearized ODE approach is limited:
  - Small, controlled u_* to stay in linearity region (experimental techniques?), but large u_* for better SNR
  - No saturation, Michaelis-Menten, *etc*
- Bayesian inference and experimental design for (simple) non-linear ODEs of biochemical kinetics?

Conclusions

- Fast accurate approximate inference, experimental design in disturbed linearized ODE setup
- Network sparsity is key prior assumption. Experimental design can lead to large savings
- Can be used with time-course measurements just as well
- Robust, easy-to-use method. Code with Matlab interface will be released
- Linearized ODE approach is limited:
  - Small, controlled u_* to stay in linearity region (experimental techniques?), but large u_* for better SNR
  - No saturation, Michaelis-Menten, *etc*
- Bayesian inference and experimental design for (simple) non-linear ODEs of biochemical kinetics?

Conclusions

- Fast accurate approximate inference, experimental design in disturbed linearized ODE setup
- Network sparsity is key prior assumption. Experimental design can lead to large savings
- Can be used with time-course measurements just as well
- Robust, easy-to-use method. Code with Matlab interface will be released
- Linearized ODE approach is limited:
  - Small, controlled u_* to stay in linearity region (experimental techniques?), but large u_* for better SNR
  - No saturation, Michaelis-Menten, *etc*
- Bayesian inference and experimental design for (simple) non-linear ODEs of biochemical kinetics?

Conclusions

- Fast accurate approximate inference, experimental design in disturbed linearized ODE setup
- Network sparsity is key prior assumption. Experimental design can lead to large savings
- Can be used with time-course measurements just as well
- Robust, easy-to-use method. Code with Matlab interface will be released
- Linearized ODE approach is limited:
  - Small, controlled $\boldsymbol{u}_*$ to stay in linearity region (experimental techniques?), but large $\boldsymbol{u}_*$ for better SNR
  - No saturation, Michaelis-Menten, *etc*
- Bayesian inference and experimental design for (simple) non-linear ODEs of biochemical kinetics?

Conclusions

- Fast accurate approximate inference, experimental design in disturbed linearized ODE setup
- Network sparsity is key prior assumption. Experimental design can lead to large savings
- Can be used with time-course measurements just as well
- Robust, easy-to-use method. Code with Matlab interface will be released
- Linearized ODE approach is limited:
  - Small, controlled $\boldsymbol{u}_*$ to stay in linearity region (experimental techniques?), but large $\boldsymbol{u}_*$ for better SNR
  - No saturation, Michaelis-Menten, *etc*
- Bayesian inference and experimental design for (simple) non-linear ODEs of biochemical kinetics?

Conclusions (II)

- Other applications of sparse (generalized) linear models, in systems biology and beyond (natural image statistics, neural spike coding, adaptive control, *etc*)
- Applications to dynamical or nonparametric models?
- Submitted for journal publication
- Details:
  M. Seeger, F. Steinke, K. Tsuda
  Bayesian Inference and Optimal Design in the Sparse Linear Model, AI and Statistics 2007
  `www.kyb.tuebingen.mpg.de/bs/people/seeger`
- Useful for your work? Do not hesitate to get in touch