CS-E5885 Modeling biological networks
Biological network structure selection: Approximative methods

Harri Lähdesmäki

Department of Computer Science
Aalto University

February 2, 2024

Outline

- Linear regression example: polynomial model selection
- Cross-validation
- Bayesian model selection
- Gradient matching
- Bayesian information criterion

- Reading:
  - This lecture follows parts of Section 7 and 8 from (Murphy, 2012) as well as parts of Section 10 from (Wilkinson, 2011)

Network structure selection

- Assume a biological system that contains n chemical species $\{x_1, \ldots, x_n\}$
- Assume ODE modeling framework
- Structure of the model, also called network structure, can be defined by a directed graph $\mathcal{G}(V, E)$, where $V = \{x_1, \ldots, x_n\}$ and $E = \{(x_s, x_t) : x_s, x_t \in V\}$ contains directed edges (from x_s to x_t) between nodes V

Network structure selection

- Assume a biological system that contains n chemical species $\{x_1, \dots, x_n\}$
- Assume ODE modeling framework
- Structure of the model, also called network structure, can be defined by a directed graph $\mathcal{G}(V, E)$, where $V = \{x_1, \dots, x_n\}$ and $E = \{(x_s, x_t) : x_s, x_t \in V\}$ contains directed edges (from x_s to x_t) between nodes V
- For each variable (node) $x_i \in V$ in an ODE model, we have a 1-D differential equation model

  $$\frac{dx_i(t)}{dt} = f_i(\hat{\mathbf{x}}_i(t)|\theta_i),$$

  where $\hat{\mathbf{x}}_i(t) = (x_{i_1}(t), \dots, x_{i_{k_i}}(t))$ defines a set of variables that regulate x_i and correspond to edges in E that point to x_i, i.e.,
  - If $\{x_{i_1}, \dots, x_{i_{k_i}}\}$ are the incoming edges to x_i in $\mathcal{G}$, then $\hat{\mathbf{x}}_i(t) = (x_{i_1}(t), \dots, x_{i_{k_i}}(t))$
- As we have discussed earlier, a highly interesting problem is the one where both the driving function f_i (or its parameters θ_i) and the (sub)set of variables $\hat{\mathbf{x}}_i(t)$ that regulate x_i are unknown

Network structure selection (2)

- For each variable x_i, there are 2^n different possible combinations/subsets of variables $\{x_1, \ldots, x_n\}$ (assuming there are no known biological constraints)
- For a full ODE system of n variables, there are $2^{(n^2)}$ different network structures
- There might also be a family of (parametric) driving functions, such as mass-action, Michaelis-Menten, linear, etc., to choose from for each variable x_i: $f^{(1)}, \ldots, f^{(\ell)}, \ldots$
- In other words, there are a very large number of variable combinations + functions to be considered

Polynomial parameter estimation

- Consider now a polynomial model

$$y = \beta_0 + \sum_{i=1}^{d} \beta_i x^i + \epsilon,$$

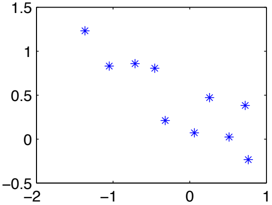
  where $\epsilon \sim N(0, \sigma^2)$

- In this linear model, the number d of x^i terms in the polynomial defines the model structure
- If X_d denotes the design matrix corresponding to the model d, then the ML parameters can be obtained using the standard formula

$$\hat{\beta}_d = (X_d^T X_d)^{-1} X_d^T \mathbf{y}$$

  assuming $X_d^T X_d$ is full rank

A polynomial fit example



A polynomial fit example

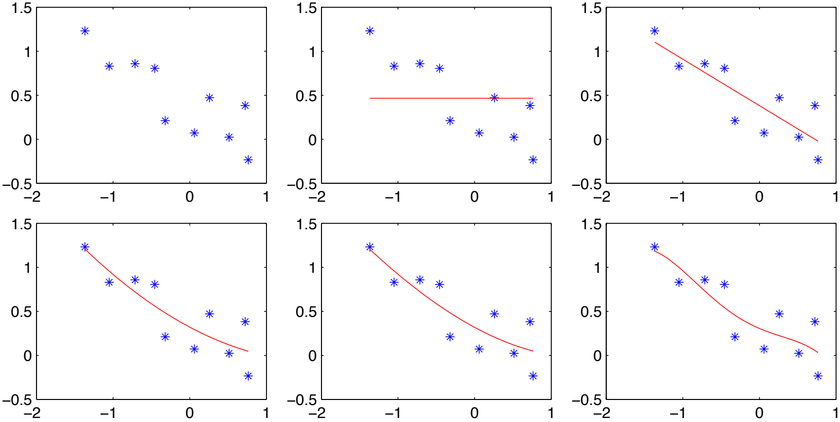
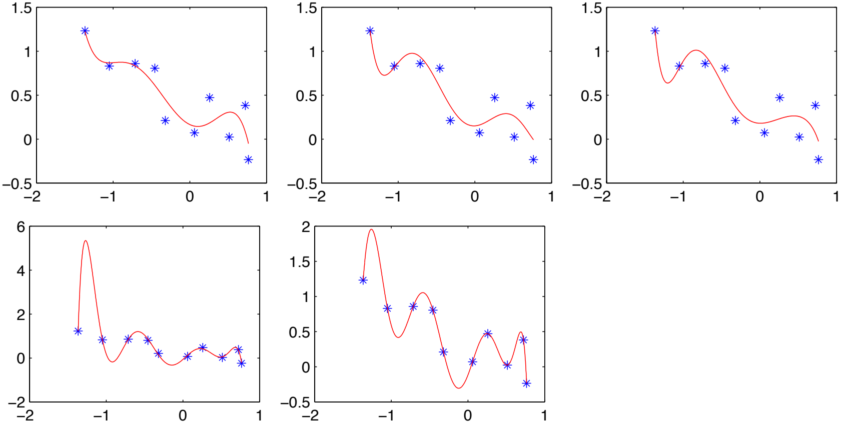


Figure: Illustration of polynomial model fitting with varying order $d \in \{0, 1, \dots, 4\}$.

A polynomial fit example (2)



A polynomial fit example (2)

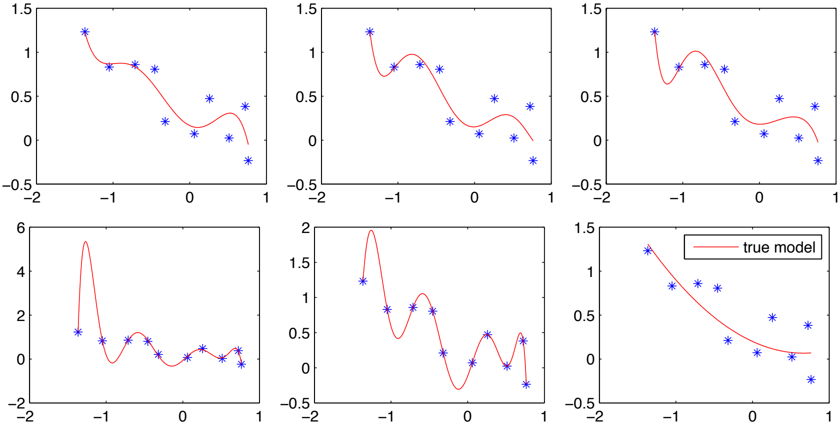


Figure: Illustration of polynomial model fitting with varying order $d \in \{5, \ldots, 9\}$.

Model selection

- How do we find the correct/best polynomial model? That is, how do we choose d?
- How do we find the correct/best model structure for our biological network model?

Model selection

- How do we find the correct/best polynomial model? That is, how do we choose d?
- How do we find the correct/best model structure for our biological network model?
- Pure error minimization, such as maximum likelihood approach, fails because
  - Parameters of a model are fitted to the whole data without taking into consideration the model complexity
  - More complex models, i.e. higher order polynomials, will decrease the error although they may be far away from the true model
  - Similarly, larger subsets of regulatory variables $\hat{\mathbf{x}}_i(t)$ in ODEs will provide increasingly better fits to data
  - Highly complex models do not generally generalize to unseen data points
  - Such a model is said to be overfitted to the given data

Model selection

- How do we find the correct/best polynomial model? That is, how do we choose d?
- How do we find the correct/best model structure for our biological network model?
- Pure error minimization, such as maximum likelihood approach, fails because
  - Parameters of a model are fitted to the whole data without taking into consideration the model complexity
  - More complex models, i.e. higher order polynomials, will decrease the error although they may be far away from the true model
  - Similarly, larger subsets of regulatory variables $\hat{\mathbf{x}}_i(t)$ in ODEs will provide increasingly better fits to data
  - Highly complex models do not generally generalize to unseen data points
  - Such a model is said to be overfitted to the given data
- Some objective and principled model selection method is needed
- Standard model selection methods include
  - Assess predictive accuracy, e.g. cross-validation
  - Bayesian model selection

Cross-validation

- Quantify predictive accuracy of a model on a separate test data, which is not used for learning the model parameters
- If such additional test data does not exist, we can use cross-validation

Cross-validation

- Quantify predictive accuracy of a model on a separate test data, which is not used for learning the model parameters
- If such additional test data does not exist, we can use cross-validation
- In k-fold cross-validation, the dataset D is split into k non-overlapping parts $D_1, D_2, \ldots, D_k$ that have approximately the same size, i.e.:

$$D_i \cap D_j = \emptyset,\ i \neq j, \quad |D_i| \approx |D_j|,\ i \neq j, \quad \text{and} \quad D = \cup_i D_i$$

Cross-validation

- Quantify predictive accuracy of a model on a separate test data, which is not used for learning the model parameters
- If such additional test data does not exist, we can use cross-validation
- In k-fold cross-validation, the dataset D is split into k non-overlapping parts $D_1, D_2, \ldots, D_k$ that have approximately the same size, i.e.:

$$D_i \cap D_j = \emptyset,\ i \neq j, \quad |D_i| \approx |D_j|,\ i \neq j, \quad \text{and} \quad D = \cup_i D_i$$

- Each set D_i is left out from the training data in turn and the model parameters are estimated using

  $$D_{-i} = \{D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_k\},$$

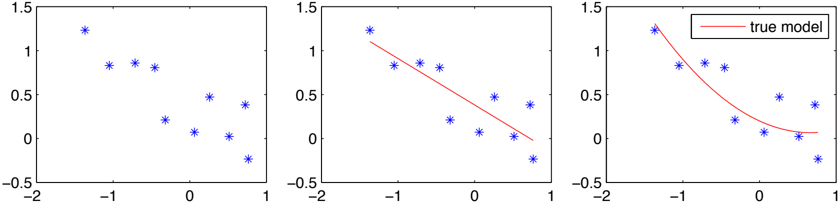
  and the accuracy of the model is tested on the left out set D_i
  - Accuracy measure can be based on e.g. mean squared error, likelihood, posterior predictive distribution, etc.
- This process is repeated for all k data folds and the average accuracy across the k repetitions is computed

Leave-one-out cross-validation

- If $k = N$ where N is the number of data points this corresponds to the leave-one-out cross-validation (LOOCV)
- Cross-validation gives an approximately unbiased prediction accuracy estimate for a model that is trained from data set that has size $N - N/k$
- Computationally rather expensive for large values of k

A polynomial fit example (cont'd)

- Lets get back to the polynomial fit example
- LOOCV estimated mean squared prediction errors are shown below for different model structures
  - $d = 0$: 0.23458789, $d = 1$: 0.079221035, $d = 2$: 0.096859549, $d = 3$: 0.13213058, $d = 4$: 0.64508982, $d = 5$: 0.76196395, $d = 6$: 3.8143803, $d = 7$: 1635.9915, $d = 8$: 1197.8935



An ODE model selection

An idealistic/brute-force approach for (small) biological networks

- Assume N time-course data sets and use the LOOCV approach (i.e., $k = N$ data folds)
- Fix a biological network model structure for now, call it M_1

An ODE model selection

An idealistic/brute-force approach for (small) biological networks

- Assume N time-course data sets and use the LOOCV approach (i.e., $k = N$ data folds)
- Fix a biological network model structure for now, call it M_1
- Iterate over N data folds
    1. Train ODE model parameters (e.g. sum of squared errors, maximum likelihood or posterior) for M_1 on D_{-i} using tools from previous lectures
    2. Test prediction accuracy (e.g. sum of squared errors, likelihood or predictive posterior) on left-out data D_i
    3. Compute average prediction accuracy over all data folds

An ODE model selection

An idealistic/brute-force approach for (small) biological networks

- Assume N time-course data sets and use the LOOCV approach (i.e., $k = N$ data folds)
- Fix a biological network model structure for now, call it M_1
- Iterate over N data folds
  1. Train ODE model parameters (e.g. sum of squared errors, maximum likelihood or posterior) for M_1 on D_{-i} using tools from previous lectures
  2. Test prediction accuracy (e.g. sum of squared errors, likelihood or predictive posterior) on left-out data D_i
  3. Compute average prediction accuracy over all data folds
- Repeat for all 2^{n^2} biological network models
- Requires solving the system and optimizing the parameters excessively many times
- Can be computationally very challenging!

An ODE model selection (2)

- The brute-force search can be made faster by
  - Using a search algorithm: e.g., start from the empty model (empty graph; no directed edges between variables) and sequentially add (but do not remove) more edges
  - Incorporating biological constraint: in best case this reduces the number of possible ODE models to something manageable (recall the example in the previous lecture with 12 models)
  - Using approximative model fitting methods, such as gradient matching

An ODE model selection: gradient matching

- Gradient matching is a commonly used heuristic that approximates time derivatives with finite differences
- Assume $N + 1$ measurements $(x_i(t_0), x_i(t_1), \dots, x_i(t_N))$ for the ith variable
- For time-series measurements, the gradient matching corresponds to

$$\frac{\mathrm{d}x_i(t_n)}{\mathrm{d}t} \simeq \Delta x_i(t_n) = \frac{x_i(t_{n+1}) - x_i(t_n)}{t_{n+1} - t_n}$$

and for steady state measurements

$$\frac{\mathrm{d}x_i(t_n)}{\mathrm{d}t} \simeq \Delta x_i(t_n) = 0$$

An ODE model selection: gradient matching

- Gradient matching is a commonly used heuristic that approximates time derivatives with finite differences
- Assume $N + 1$ measurements $(x_i(t_0), x_i(t_1), \ldots, x_i(t_N))$ for the ith variable
- For time-series measurements, the gradient matching corresponds to

$$\frac{\mathrm{d}x_i(t_n)}{\mathrm{d}t} \simeq \Delta x_i(t_n) = \frac{x_i(t_{n+1}) - x_i(t_n)}{t_{n+1} - t_n}$$

  and for steady state measurements

$$\frac{\mathrm{d}x_i(t_n)}{\mathrm{d}t} \simeq \Delta x_i(t_n) = 0$$

- Thus, ODE model fitting reduces to a regression model using N data points (x_n, y_n), where $x_n = \hat{\mathbf{x}}_i(t_n)$ and $y_n = \Delta x_i(t_n)$
- In other words, find function f_i (or its parameters θ_i) so that

$$\Delta x_i(t_n) = \frac{x_i(t_{n+1}) - x_i(t_n)}{t_{n+1} - t_n} \approx f_i(\hat{\mathbf{x}}_i(t_n)|\theta_i)$$

An ODE model selection: gradient matching (2)

- Sometime model is approximated further by assuming the RHS is linear in parameters

$$\Delta x_i(t_n) = \frac{x_i(t_{n+1}) - x_i(t_n)}{t_{n+1} - t_n} \approx \beta_0 + \beta_{i_1} x_{i_1}(t_n) + \ldots + \beta_{i_{k_i}} x_{i_{k_i}}(t_n)$$

- This would further reduce ODE model fitting to a linear regression model

An ODE model selection: gradient matching (3)

- Gradient matching results in significant reduction in time complexity because
  - Each variable x_i can be analyzed independently: $O(n2^n)$ time complexity instead of $O(2^{(n^2)})$
  - Linear or non-linear regression instead of ODE model fitting
- Using gradient matching with linear approximation, we can find relatively efficiently (at least for small networks):
  - Optimal parameters $\{\beta_0, \beta_{i_1}, \ldots, \beta_{i_{k_i}}\}$
    - Linear model fitting with ML/ordinary least squares estimation
  - Optimal regulators $\hat{\mathbf{x}}_i(t)$ for each variable i
    - Linear regression based model selection

An ODE model selection: gradient matching (3)

- Gradient matching results in significant reduction in time complexity because
  - Each variable x_i can be analyzed independently: $O(n2^n)$ time complexity instead of $O(2^{(n^2)})$
  - Linear or non-linear regression instead of ODE model fitting
- Using gradient matching with linear approximation, we can find relatively efficiently (at least for small networks):
  - Optimal parameters $\{\beta_0, \beta_{i_1}, \ldots, \beta_{i_{k_i}}\}$
    - Linear model fitting with ML/ordinary least squares estimation
  - Optimal regulators $\hat{\mathbf{x}}_i(t)$ for each variable i
    - Linear regression based model selection
- Many extensions have been proposed:
  - Use e.g. basis function extension to model nonlinearities
- Can provide an efficient approach also for non-linear models:
  - Mass-action kinetics, Michealis-Menten, etc.
  - Black-box and non-parametric models, such as neural networks and Gaussian processes
  - etc.

An ODE example (1)

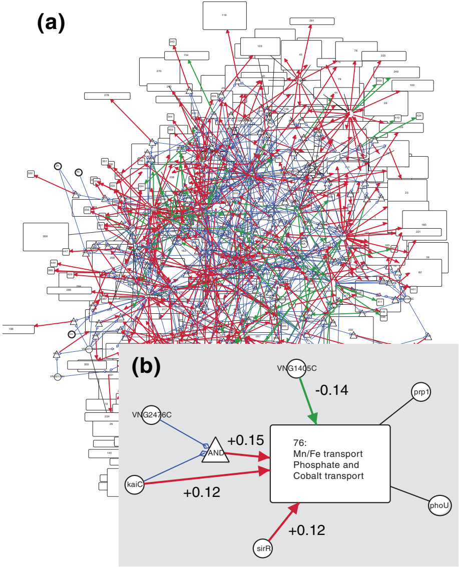
- Network structure selection example from (Bonneau et al., 2006)
- Learn transcriptional regulatory networks in halobacterium from gene expression data using an ODE model of the form

$$\frac{dY}{dt} = f(\beta_{i_1} X_{i_1} + \ldots + \beta_{i_k} X_{i_k}) - \tau Y$$

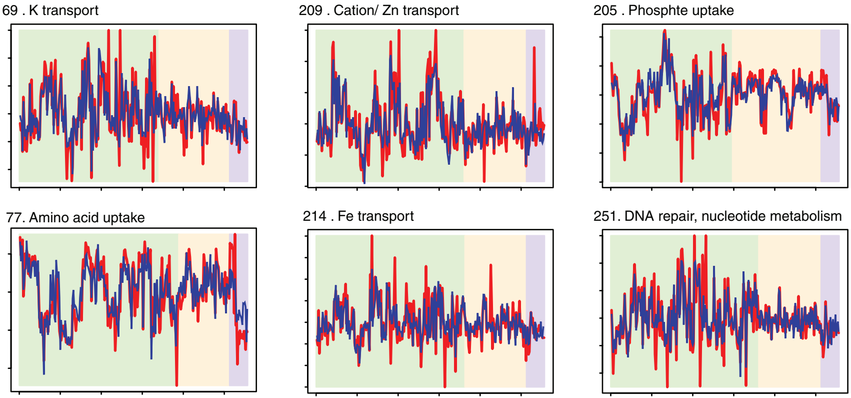
  where f is a sigmoidal type of function

  - Y is the target gene and $X_{i_1}, \ldots, X_{i_k}$ are a subset of other genes in halobacterium
- Gradient matching and model selection using cross-validation

An ODE example (2)



An ODE example (3)



69 . K transport

209 . Cation/ Zn transport

205 . Phosphte uptake

77. Amino acid uptake

214 . Fe transport

251. DNA repair, nucleotide metabolism

Figure from (Bonneau et al., 2006)

An SDE model fitting and selection: gradient matching

- Recall the chemical Langevin equation SDE model

  $$dX_t = \mu(X_t, c)dt + \sqrt{\beta(X_t, c)}dW_t,$$

  where $\mu(x, c) = Sh(x, c)$ and $\beta(x, c) = S\mathrm{diag}\{h(x, c)\}S^T$
- Model structure defined by S (and $h(\cdot)$), c denotes parameters

An SDE model fitting and selection: gradient matching

- Recall the chemical Langevin equation SDE model

  $$dX_t = \mu(X_t, c)dt + \sqrt{\beta(X_t, c)}dW_t,$$

  where $\mu(x, c) = Sh(x, c)$ and $\beta(x, c) = S\mathrm{diag}\{h(x, c)\}S^T$
- Model structure defined by S (and $h(\cdot)$), c denotes parameters
- Assume noise-free data $x = (x_0, x_{\Delta t}, x_{2\Delta t}, \ldots, x_{N\Delta t})$ collected with a small time step Δt
- When Δt is small, then the Euler-Maruyama approximation is accurate

  $$\Delta X_t \triangleq X_{t+\Delta t} - X_t \approx \mu(X_t, c)\Delta t + \sqrt{\beta(X_t, c)}\Delta W_t$$

An SDE model fitting and selection: gradient matching

- Recall the chemical Langevin equation SDE model

$$dX_t = \mu(X_t, c)dt + \sqrt{\beta(X_t, c)}dW_t,$$

  where $\mu(x, c) = Sh(x, c)$ and $\beta(x, c) = S\mathrm{diag}\{h(x, c)\}S^T$
- Model structure defined by S (and $h(\cdot)$), c denotes parameters
- Assume noise-free data $x = (x_0, x_{\Delta t}, x_{2\Delta t}, \ldots, x_{N\Delta t})$ collected with a small time step Δt
- When Δt is small, then the Euler-Maruyama approximation is accurate

$$\Delta X_t \triangleq X_{t+\Delta t} - X_t \approx \mu(X_t, c)\Delta t + \sqrt{\beta(X_t, c)}\Delta W_t$$

- The above equation can be written as

$$X_{t+\Delta t}|X_t, c \sim N(X_t + \mu(X_t, c)\Delta t, \beta(x, c)\Delta t)$$

An SDE model fitting and selection: gradient matching

- Recall the chemical Langevian equation SDE model

$$dX_t = \mu(X_t, c)dt + \sqrt{\beta(X_t, c)}dW_t,$$

where $\mu(x, c) = Sh(x, c)$ and $\beta(x, c) = S\mathrm{diag}\{h(x, c)\}S^T$

- Model structure defined by S (and $h(\cdot)$), c denotes parameters
- Assume noise-free data $x = (x_0, x_{\Delta t}, x_{2\Delta t}, \ldots, x_{N\Delta t})$ collected with a small time step Δt
- When Δt is small, then the Euler-Maruyama approximation is accurate

$$\Delta X_t \triangleq X_{t+\Delta t} - X_t \approx \mu(X_t, c)\Delta t + \sqrt{\beta(X_t, c)}\Delta W_t$$

- The above equation can be written as

$$X_{t+\Delta t}|X_t, c \sim N(X_t + \mu(X_t, c)\Delta t, \beta(x, c)\Delta t)$$

- The likelihood model for the data x can now be written as

$$L(c|x) = p(x_0|c)\prod_{t=1}^{N} N(x_{t+\Delta t}|x_t, c)$$

Bayesian model comparison

- As discussed in the previous lecture, in Bayesian model comparison, we would like to compute the posterior probability of a model $\mathcal{M}_k$, given data $\mathcal{D}$

$$p(\mathcal{M}_k|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{M}_k)p(\mathcal{M}_k)}{p(\mathcal{D})}$$

Bayesian model comparison

- As discussed in the previous lecture, in Bayesian model comparison, we would like to compute the posterior probability of a model $\mathcal{M}_k$, given data $\mathcal{D}$

$$p(\mathcal{M}_k|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{M}_k)p(\mathcal{M}_k)}{p(\mathcal{D})}$$

- Recall polynomial model fitting: for the kth order model $\mathcal{D} = (X_k, \mathbf{y})$ and $\mathcal{M}_k \triangleq k$

$$p(k|\mathbf{y}, X_k) = \frac{p(\mathbf{y}|k, X_k)p(k)}{p(\mathbf{y})}$$

Bayesian model comparison

- As discussed in the previous lecture, in Bayesian model comparison, we would like to compute the posterior probability of a model $\mathcal{M}_k$, given data $\mathcal{D}$

$$p(\mathcal{M}_k|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{M}_k)p(\mathcal{M}_k)}{p(\mathcal{D})}$$

- Recall polynomial model fitting: for the kth order model $\mathcal{D} = (X_k, \mathbf{y})$ and $\mathcal{M}_k \triangleq k$

$$p(k|\mathbf{y}, X_k) = \frac{p(\mathbf{y}|k, X_k)p(k)}{p(\mathbf{y})}$$

- One needs to compute the marginal likelihood $p(\mathbf{y}|k, X_k)$

$$p(\mathbf{y}|k, X_k) = \int_{\beta_k} p(\mathbf{y}|k, X_k, \beta_k)p(\beta_k|k, X_k)d\beta_k,$$

where $\beta_k = (\beta_0, \beta_1, \ldots, \beta_k)^T$ and $p(\beta_k|k, X_k)$ is prior probablity of β

Bayesian model comparison (2)

- Instead of finding and using a point estimate $\hat{\beta}$, one has to average over all parameter values weighted according to a prior

Bayesian model comparison (2)

- Instead of finding and using a point estimate $\hat{\beta}$, one has to average over all parameter values weighted according to a prior
- Bayes model selection via the marginal likelihood has a built-in "Occam's razor"
  - Models that are too complex are automatically penalized

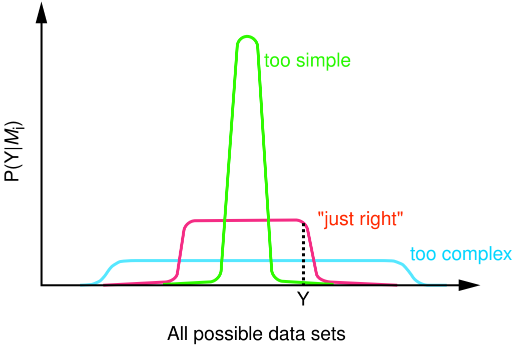


Figure from (Rasmussen, 2004)

Bayesian analysis for the linear model

- For a specific (very limited) choice of likelihood model and prior, the marginal likelihood can be solved analytically for a linear model (conjugate)

Bayesian analysis for the linear model

- For a specific (very limited) choice of likelihood model and prior, the marginal likelihood can be solved analytically for a linear model (conjugate)
- Let us assume the standard normal gamma conjugate prior for β_k and σ^2

$$\beta_k \sim N(\mu_k, \sigma^2 I) \quad \text{and} \quad \frac{\nu\lambda}{\sigma^2} \sim \chi^2_\nu,$$

where μ_k, ν and λ are hyperparameters

Bayesian analysis for the linear model

- For a specific (very limited) choice of likelihood model and prior, the marginal likelihood can be solved analytically for a linear model (conjugate)
- Let us assume the standard normal gamma conjugate prior for β_k and σ^2

$$\beta_k \sim N(\mu_k, \sigma^2 I) \quad \text{and} \quad \frac{\nu\lambda}{\sigma^2} \sim \chi^2_\nu,$$

where μ_k, ν and λ are hyperparameters

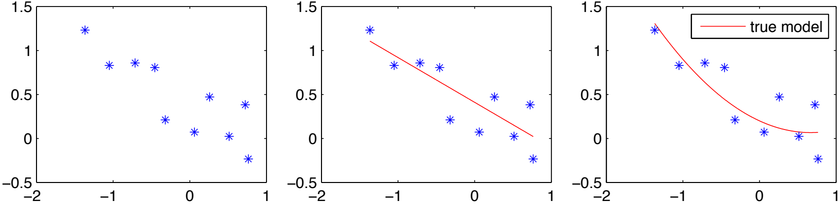
- The marginal likelihood, see e.g. (Raftery et al., 1997)

$$\begin{aligned} p(\mathbf{y}|k, X_k) &= \int_{\beta_k} p(\mathbf{y}|k, X_k, \beta_k) p(\beta_k|k, X_k, \mu, \nu, \lambda) d\beta_k \\ &= \dots \\ &= \frac{\Gamma(\frac{\nu+n}{2})(\nu\lambda)^{\nu/2}}{\pi^{n/2}\Gamma(\frac{\nu}{2})|I + \sigma^2 X_k X_k^T|^{1/2}} \\ &\quad \times \left(\lambda\nu + \mathbf{r}^T (I + \sigma^2 X_k X_k^T)^{-1} \mathbf{r}^T\right)^{-(\nu+n)/2}, \quad (*) \end{aligned}$$

where $\mathbf{r} = \mathbf{y} - X_k \mu_k$

Bayesian analysis for the linear/polynomial model (2)

- Marginal likelihood for the first seven models are shown below
  - $k = 0$: 0.00062424, $k = 1$: 0.0077628, $k = 2$: 0.0016004, $k = 3$: 0.00058334, $k = 4$: 0.00027658, $k = 5$: 0.00013991, $k = 6$: 7.246e-05, ...



Biological network structure selection using Bayesian methods

- Let M_k denote a biological network structure, and the associated ODE dynamics are

$$\frac{dx_i(t)}{dt} = f_i(\hat{\mathbf{x}}_{i_k}(t)|M_k, \theta_k), \quad i \in \{1, \ldots n\}$$

- Given data D, the marginal likelihood can be computed (by integrating out parameters) for the ODE models: $P(D|M_k)$
- This can be numerically approximated e.g. using population MCMC and thermodynamic integration
- Lets briefly go through the gradient matching approximation where computation can be done more efficiently and another example which also provides non-linearity and accurate inference (Äijö and Lähdesmäki, 2009)

Gradient matching with Bayesian methods

- Approximations: gradient matching and linear model assumption for the *i*th variable

$$\Delta x_i(t_k) \triangleq y_i(t_k) = \beta_0 + \beta_{i_1} x_{i_1}(t_k) + \ldots + \beta_{i_{k_i}} x_{i_{k_i}}(t_k) - \lambda_i x_i(t_k) + \epsilon_i(t_k),$$

where $y_i(t_k)$ is interpreted as a measurement of the finite difference and $\epsilon_i \sim \mathcal{N}(0, \sigma)$ i.i.d.

Gradient matching with Bayesian methods

- Approximations: gradient matching and linear model assumption for the *i*th variable

$$\Delta x_i(t_k) \triangleq y_i(t_k) = \beta_0 + \beta_{i_1} x_{i_1}(t_k) + \ldots + \beta_{i_{k_i}} x_{i_{k_i}}(t_k) - \lambda_i x_i(t_k) + \epsilon_i(t_k),$$

  where $y_i(t_k)$ is interpreted as a measurement of the finite difference and $\epsilon_i \sim \mathcal{N}(0, \sigma)$ i.i.d.

- For non time-course data

$$0 \triangleq y_i(t_k) = \beta_0 + \beta_{i_1} x_{i_1}(t_k) + \ldots + \beta_{i_{k_i}} x_{i_{k_i}}(t_k) - \lambda_i x_i(t_k) + \epsilon_i(t_k)$$

Gradient matching with Bayesian methods

- Approximations: gradient matching and linear model assumption for the *i*th variable

$$\Delta x_i(t_k) \triangleq y_i(t_k) = \beta_0 + \beta_{i_1} x_{i_1}(t_k) + \ldots + \beta_{i_{k_i}} x_{i_{k_i}}(t_k) - \lambda_i x_i(t_k) + \epsilon_i(t_k),$$

  where $y_i(t_k)$ is interpreted as a measurement of the finite difference and $\epsilon_i \sim \mathcal{N}(0, \sigma)$ i.i.d.

- For non time-course data

$$0 \triangleq y_i(t_k) = \beta_0 + \beta_{i_1} x_{i_1}(t_k) + \ldots + \beta_{i_{k_i}} x_{i_{k_i}}(t_k) - \lambda_i x_i(t_k) + \epsilon_i(t_k)$$

- Alternatively

$$\begin{aligned} y_i(t_k) &= \underbrace{(1, \hat{\mathbf{x}}_i(t_k), x_i(t_k))}_{\mathbf{x_k^T}} \boldsymbol{\beta} + \epsilon_i(t_k) \\ &= \mathbf{x_k^T} \boldsymbol{\beta} + \epsilon_i(t_k) \end{aligned}$$

  where $\hat{\mathbf{x}}_i(t) = (x_{i_1}(t), \ldots, x_{i_{k_i}}(t))$ and $\boldsymbol{\beta} = (1, \beta_0, \beta_{i_1}, \ldots, \beta_{i_{k_i}}, \lambda_i)^T$

Gradient matching with Bayesian methods (2)

- Collectively, for all time points

$$\mathbf{y}_i = X_i \boldsymbol{\beta} + \varepsilon_i$$

  where $\mathbf{y}_i$ contain $y_i(t_k)$ for different values of t_k and X_i contains $\mathbf{x}_k^T = (1 \ \hat{\mathbf{x}}(t_k), x_i(t_k))$ as rows
- Marginal likelihood for the ith variable in model M_k can be computed as in (*): $p(\mathbf{y}_i | M_k, X_i)$
- Each variable x_i is analyzed independently

Gradient matching with Bayesian methods (2)

- Collectively, for all time points

$$\mathbf{y}_i = X_i\boldsymbol{\beta} + \varepsilon_i$$

  where $\mathbf{y}_i$ contain $y_i(t_k)$ for different values of t_k and X_i contains $\mathbf{x}_k^T = (1\ \hat{\mathbf{x}}(t_k), x_i(t_k))$ as rows
- Marginal likelihood for the ith variable in model M_k can be computed as in (*): $p(\mathbf{y}_i|M_k, X_i)$
- Each variable x_i is analyzed independently
- Compute and combine results for all the variables in M_k

$$p(\mathbf{y}|M_k, X) = \prod_{i=1}^{n} p(\mathbf{y}_i|M_k, X_i)$$

- Bayesian posterior probability for the network model M is then

$$P(M_k|\mathbf{y}) = \frac{p(\mathbf{y}|M_k, X)P(M_k)}{P(\mathbf{y})}$$

Model averaging approach

- For each biological network model M_k we get the probability

$$P(M_k|\mathbf{y})$$

- Often (unfortunately) no network model stands out as unique, but rather several networks have a similar score
- To assess the overall evidence for a directed edge from x_k to x_l we can use an approach called Bayesian model averaging

$$P(x_k \to x_l|\mathbf{y}) = \sum_{M \,:\, (x_k,x_l)\in E} P(M|\mathbf{y})$$

Model averaging approach

- For each biological network model M_k we get the probability

$$P(M_k|\mathbf{y})$$

- Often (unfortunately) no network model stands out as unique, but rather several networks have a similar score
- To assess the overall evidence for a directed edge from x_k to x_l we can use an approach called Bayesian model averaging

$$P(x_k \rightarrow x_l|\mathbf{y}) = \sum_{M \,:\, (x_k, x_l) \in E} P(M|\mathbf{y})$$

- Results shown on the next slides are obtained by a non-linear approximation

An ODE model selection: a nonlinear approximation

- A transcriptional regulation model for gene x_i

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = \alpha_i + f_i(\hat{\mathbf{x}}_i(t)) - \lambda_i x_i(t)$$

  - α_i is the basal transcription rate
  - f_i is an unknown and non-parametric regulation function (in technical terms, f_i has a Gaussian process prior)
  - $\hat{\mathbf{x}}_i(t) = (x_{i_1}(t), \ldots, x_{i_k}(t))$ denotes the expressions of genes/TFs that regulate gene x_i
  - λ_i is the decay rate of the mRNA

An ODE model selection: a nonlinear approximation

- A transcriptional regulation model for gene x_i

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = \alpha_i + f_i(\hat{\mathbf{x}}_i(t)) - \lambda_i x_i(t)$$

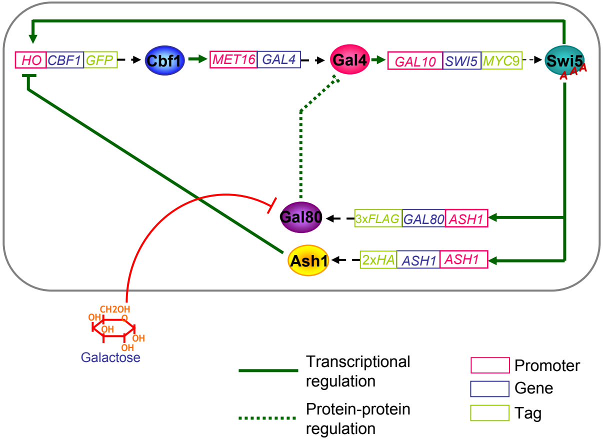
  - α_i is the basal transcription rate
  - f_i is an unknown and non-parametric regulation function (in technical terms, f_i has a Gaussian process prior)
  - $\hat{\mathbf{x}}_i(t) = (x_{i_1}(t), \ldots, x_{i_k}(t))$ denotes the expressions of genes/TFs that regulate gene x_i
  - λ_i is the decay rate of the mRNA
- For time-series and steady state measurements

$$\frac{\mathrm{d}x_i(t_k)}{\mathrm{d}t} \simeq \Delta x_i(t_k) \quad \text{and} \quad \frac{\mathrm{d}x_i(t)}{\mathrm{d}t} \simeq 0$$

- Model averaging: $P(x_k \to x_l | \mathbf{y})$

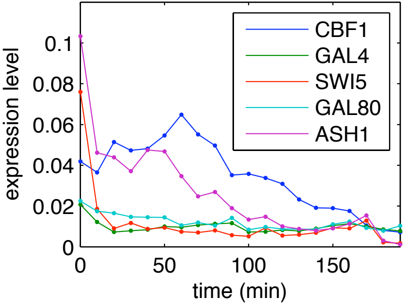
Synthetic IRMA network

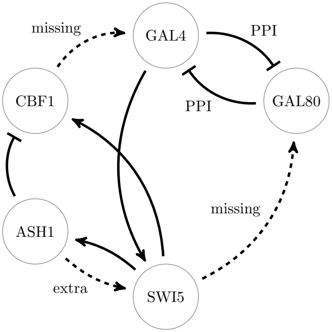
- mRNA measurements from *in vivo* reverse-engineering and modeling assessment (IRMA) network (Cantone et al., 2009)



Results for IRMA

- Inferred regulatory connections for the IRMA network

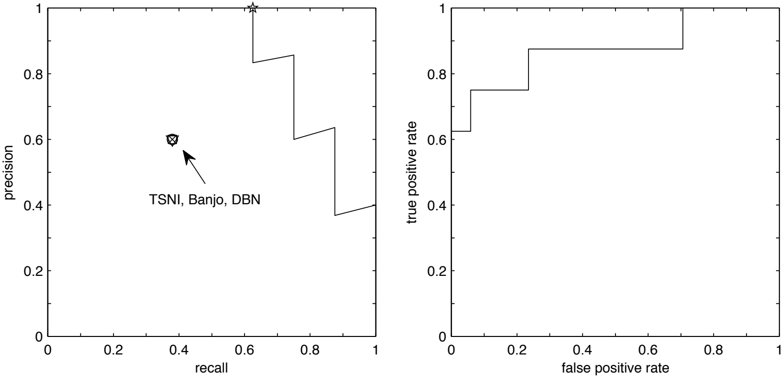




(Äijö and Lähdesmäki, 2009)

Results for IRMA (2)

- Precision-recall and receiver operating characteristics curves (P-ROC and ROC)



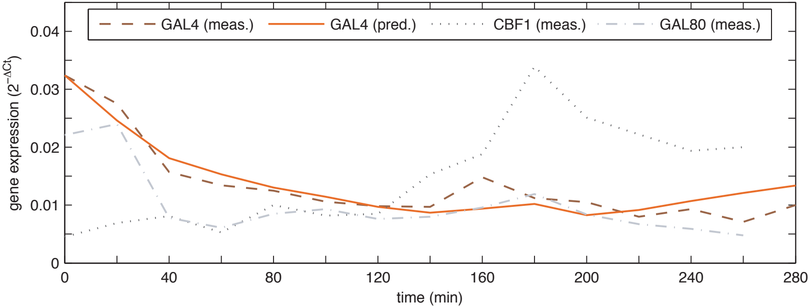
(a) P-ROC curve

(b) ROC curve

(Äijö and Lähdesmäki, 2009)

Results for IRMA (3)

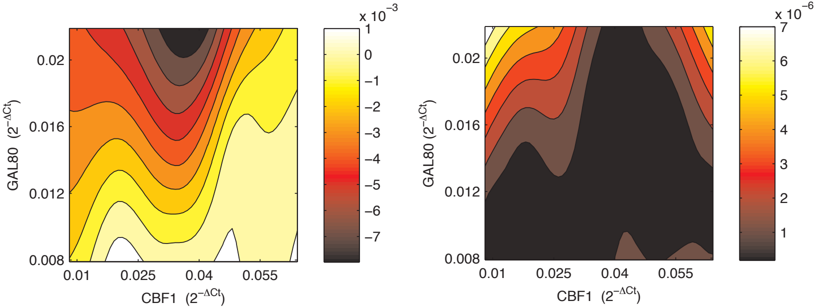
- Predictive behavior for GAL4 gene (independent validation data)



(Äijö and Lähdesmäki, 2009)

Results for IRMA (4)

- Inferred regulatory function f_i for GAL4



(a) Estimated regulatory function,

(b) Variance of the estimate.

(Äijö and Lähdesmäki, 2009)

Bayesian information criterion for model structure selection

- Bayesian model structure selection involves the marginal likelihood term that is generally difficult to compute

$$p(\mathcal{D}|\mathcal{M}_k) = \int_{\theta_k} p(\mathcal{D}|\mathcal{M}_k, \theta_k) p(\theta_k|\mathcal{M}_k) d\theta_k$$

- Next we look at a commonly used approximation technique for the marginal likelihood, so-called Bayesian information criterion (BIC) score

Bayesian information criterion for model structure selection

- Bayesian model structure selection involves the marginal likelihood term that is generally difficult to compute

$$p(\mathcal{D}|\mathcal{M}_k) = \int_{\theta_k} p(\mathcal{D}|\mathcal{M}_k, \theta_k) p(\theta_k|\mathcal{M}_k) d\theta_k$$

- Next we look at a commonly used approximation technique for the marginal likelihood, so-called Bayesian information criterion (BIC) score
- We will show that the approximation for the logarithm of the marginal likelihood has the following form:

$$\ln p(\mathcal{D}|\mathcal{M}_k) \approx \ln p(\mathcal{D}|\mathcal{M}_k, \hat{\theta}_k) - \frac{d}{2} \ln N,$$

  where $\hat{\theta}_k$ denotes the maximum likelihood or maximum a posteriori (MAP) parameter value, $d = \dim(\theta_k)$ and N denotes the number of data points
- Note that this approximation uses point estimate $\hat{\theta}_k$ which can be efficiently obtained using the gradient-based optimization and sensitivity equations or adjoints
- In the following derivation (from (Murphy, 2012)), we will drop off $\mathcal{M}_k$ from the notation for simplicity

Laplace approximation to integral

- Assume parameters $\theta \in \mathbb{R}^d$ and a (posterior) distribution

  $$p(\theta|\mathcal{D}) = \frac{p(\theta, \mathcal{D})}{p(\mathcal{D})} = \frac{1}{Z}\exp(-E(\theta)),$$

  where $E(\theta) = -\ln p(\theta, \mathcal{D})$ and $Z = p(\mathcal{D})$

Laplace approximation to integral

- Assume parameters $\theta \in \mathbb{R}^d$ and a (posterior) distribution

  $$p(\theta|\mathcal{D}) = \frac{p(\theta, \mathcal{D})}{p(\mathcal{D})} = \frac{1}{Z}\exp(-E(\theta)),$$

  where $E(\theta) = -\ln p(\theta, \mathcal{D})$ and $Z = p(\mathcal{D})$
- We can apply Taylor series expansion around the mode θ^* (i.e., the highest probability value)

  $$\hat{E}(\theta) \approx E(\theta^*) + (\theta - \theta^*)^T \mathbf{g} + \frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*),$$

  where $\mathbf{g}$ is the gradient of E and H is the hessian of E evaluated at θ^*

  $$\begin{aligned} \mathbf{g} &= \nabla E(\theta)\big|_{\theta=\theta^*} \\ H &= \left.\frac{\partial^2 E(\theta)}{\partial\theta\partial\theta^T}\right|_{\theta=\theta^*} \end{aligned}$$

Laplace approximation to integral (2)

- Because the gradient at the mode is zero, we obtain

$$\begin{aligned} p(\theta, \mathcal{D}) &\approx \hat{p}(\theta, \mathcal{D}) = \exp(-\hat{E}(\theta)) \\ &= \exp\left(-E(\theta^*) - \frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)\right) \\ &= \exp(-E(\theta^*)) \exp\left(-\frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)\right) \end{aligned}$$

Laplace approximation to integral (2)

- Because the gradient at the mode is zero, we obtain

$$\begin{aligned} p(\theta, \mathcal{D}) &\approx \hat{p}(\theta, \mathcal{D}) = \exp(-\hat{E}(\theta)) \\ &= \exp\left(-E(\theta^*) - \frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)\right) \\ &= \exp(-E(\theta^*)) \exp\left(-\frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)\right) \end{aligned}$$

- We also get

$$\begin{aligned} \hat{p}(\theta|\mathcal{D}) &= \frac{1}{Z}\hat{p}(\theta, \mathcal{D}) = \frac{1}{Z}\underbrace{\exp(-E(\theta^*))}_{\text{constant w.r.t. } \theta} \exp\left(-\frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)\right) \\ &\propto \mathcal{N}(\theta|\theta^*, H^{-1}) \end{aligned}$$

Laplace approximation to integral (2)

- Because the gradient at the mode is zero, we obtain

$$\begin{aligned} p(\theta, \mathcal{D}) &\approx \hat{p}(\theta, \mathcal{D}) = \exp(-\hat{E}(\theta)) \\ &= \exp\left(-E(\theta^*) - \frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)\right) \\ &= \exp(-E(\theta^*)) \exp\left(-\frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)\right) \end{aligned}$$

- We also get

$$\begin{aligned} \hat{p}(\theta|\mathcal{D}) &= \frac{1}{Z}\hat{p}(\theta, \mathcal{D}) = \frac{1}{Z}\underbrace{\exp(-E(\theta^*))}_{\text{constant w.r.t. } \theta} \exp\left(-\frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)\right) \\ &\propto \mathcal{N}(\theta|\theta^*, H^{-1}) \end{aligned}$$

- The normalization constant is

$$Z = p(\mathcal{D}) \approx \int \hat{p}(\theta|\mathcal{D}) d\theta = \exp(-E(\theta^*))(2\pi)^{-d/2}|H|^{-1/2}$$

Bayesian information criterion

- Using the normal approximation to the marginal likelihood we get

$$
\begin{aligned}
\ln p(\mathcal{D}) &\approx \ln\left(\exp(-E(\theta^*))(2\pi)^{-d/2}|H|^{-1/2}\right) \\
&\propto -E(\theta^*) - \frac{1}{2}\ln|H| \\
&= \ln p(\theta^*, \mathcal{D}) - \frac{1}{2}\ln|H| \\
&= \ln p(\mathcal{D}|\theta^*) + \ln p(\theta^*) - \frac{1}{2}\ln|H|
\end{aligned}
$$

Bayesian information criterion

- Using the normal approximation to the marginal likelihood we get

$$
\begin{aligned}
\ln p(\mathcal{D}) &\approx \ln\left(\exp(-E(\theta^*))(2\pi)^{-d/2}|H|^{-1/2}\right) \\
&\propto -E(\theta^*) - \frac{1}{2}\ln|H| \\
&= \ln p(\theta^*, \mathcal{D}) - \frac{1}{2}\ln|H| \\
&= \ln p(\mathcal{D}|\theta^*) + \ln p(\theta^*) - \frac{1}{2}\ln|H|
\end{aligned}
$$

- If we assume uniform prior, we can drop the second term $\ln p(\theta^*)$

Bayesian information criterion

- Using the normal approximation to the marginal likelihood we get

$$\begin{aligned}
\ln p(\mathcal{D}) &\approx \ln\left(\exp(-E(\theta^*))(2\pi)^{-d/2}|H|^{-1/2}\right)\\
&\propto -E(\theta^*) - \frac{1}{2}\ln|H|\\
&= \ln p(\theta^*, \mathcal{D}) - \frac{1}{2}\ln|H|\\
&= \ln p(\mathcal{D}|\theta^*) + \ln p(\theta^*) - \frac{1}{2}\ln|H|
\end{aligned}$$

- If we assume uniform prior, we can drop the second term $\ln p(\theta^*)$
- We can write $H = \sum_{i=1}^{N} H_i$, where N is the number of data points, $\mathcal{D}_i$ is the ith data point and

$$H_i = \frac{\partial^2 \ln p(\mathcal{D}_i|\theta)}{\partial\theta\partial\theta^T}$$

Bayesian information criterion (2)

- If we further assume that each $H_i = \hat{H}$ is fixed we have

$$\ln|H| = \ln|N\hat{H}| = \ln N^d|\hat{H}| = d\ln N + \ln|\hat{H}|,$$

where $d = \mathrm{dim}(\theta)$

Bayesian information criterion (2)

- If we further assume that each $H_i = \hat{H}$ is fixed we have

  $$\ln|H| = \ln|N\hat{H}| = \ln N^d|\hat{H}| = d \ln N + \ln|\hat{H}|,$$

  where $d = \mathrm{dim}(\theta)$
- Finally, because $\ln|\hat{H}|$ does not depend on N, an asymptotic approximation to the marginal likelihood can be written as

  $$\ln p(\mathcal{D}) \approx \ln p(\mathcal{D}|\hat{\theta}) - \frac{d}{2} \ln N$$

- This is called the Bayesian information criterion

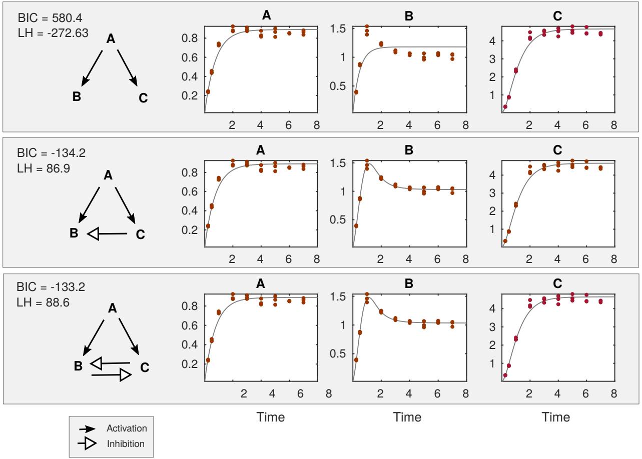
Bayesian information criterion: illustration

- Consider a simply (gene regulatory) network model consisting of three genes whose dynamics are governed by the following ODE model

$$
\begin{aligned}
\frac{d[A]}{dt} &= k_{\mathrm{bas}}^{A} - k_{\mathrm{dec}}^{A}[A] \\
\frac{d[B]}{dt} &= k_{\mathrm{bas}}^{B} + k_{\mathrm{act}}^{AB}[A] - k_{\mathrm{inh}}^{CB}[B][C] - k_{\mathrm{dec}}^{B}[B] \\
\frac{d[C]}{dt} &= k_{\mathrm{bas}}^{C} + k_{\mathrm{act}}^{AC}[A] - k_{\mathrm{inh}}^{BC}[B][C] - k_{\mathrm{dec}}^{C}[C]
\end{aligned}
$$

- We will consider three different model structure
  - Model 1: $k_{\mathrm{inh}}^{CB} = 0$ and $k_{\mathrm{inh}}^{BC} = 0$
  - Model 2: $k_{\mathrm{inh}}^{BC} = 0$
  - Model 3: All params. are assumed to be non-zero
- Three replicated time-series experiments:
  - 9 time points
  - Additive Gaussian noise

Bayesian information criterion: illustration (2)



- BIC is computed here as -BIC, i.e., small is better (Figure credit to Juho Timonen)

References

- Bonneau R, et al. (2006). The Inferelator: a procedure for learning parsimonious regulatory networks from systems-biology data-sets de novo, *Genome Biol.* 7(5):R36.
- Murphy K (2012) Machine learning: a probabilistic perspective, MIT Press.
- Raftery, A.E., Madigan, D. and Hoeting, J.A. (1997) Bayesian model averaging for regression models. *Journal of the American Statistical Association*, 92, 179-191.
- Rasmussen CE (2004) Gaussian processes in machine learning. In: Advanced Lectures on Machine Learning. Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence, 3176 . Springer, Germany, pp. 63-71.
- Rasmussen CE and Williams CKI (2006) *Gaussian processes for machine learning.* MIT Press, Cambridge, MA, USA.
- Äijö T and Lähdesmäki H (2009) Learning gene regulatory networks from gene expression measurements using non-parametric molecular kinetics. *Bioinformatics*, 25(22):2937–2944.
- Äijö T, Granberg K, Lähdesmäki H, (2013) Sorad: A systems biology approach to predict and modulate dynamic signaling pathway response from phosphoproteome time-course measurements. *Bioinformatics*.