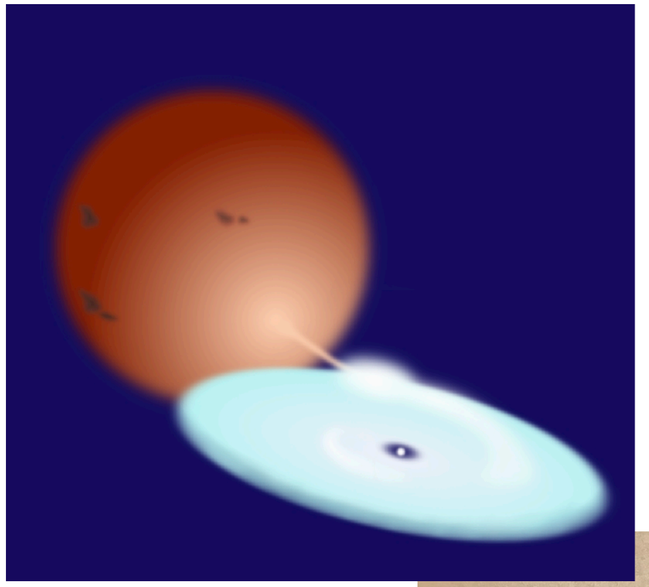
Statistical modeling + Shallow ML

AstroAI Asian Network Summer School, Sept 2-6, 2024
DAY 1　Lecture

Makoto Uemura (Hiroshima University)

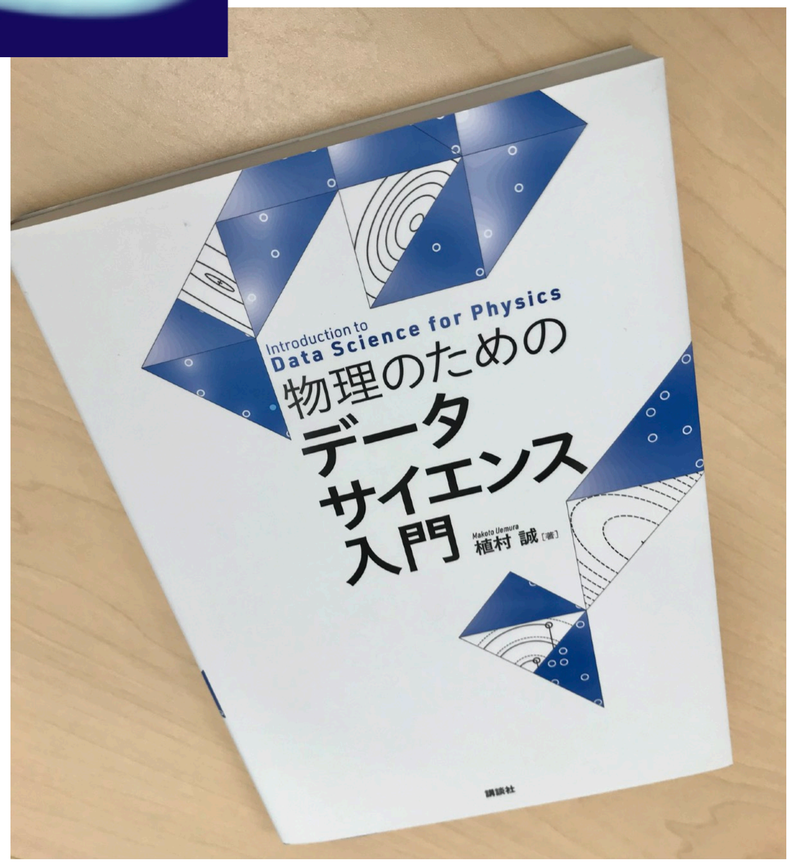
About Me



- Time Domain Astronomy (the physics of accretion & ejection around compact objects)

- Data Science for Astronomy

  - “Introduction to Data Science for Physics” 2023 Kodansha
    - Section 0: Data Science and Machine Learning, What excites you?
    - Section 1: Estimation and Test
    - Section 2: High dimensional model
    - **Section 3: Bayesian modeling**
    - Section 4: Markov-chain Monte Carlo (MCMC)
    - **Section 5: Regularization and Sparse modeling**
    - **Section 6: Classification models**
    - **Section 7: Gaussian process**
    - Section 8: Neural network
    - Appendix A: Python programs



Contents

4 topics x (30-min lecture + 15-min hands-on exercise)

- Statistical modeling
- Regularization
- Gaussian process
- Classification models

1. Statistical modeling

1-1 Bayes' theorem, posterior, prior, likelihood

1-2 Generalization error, information criterion, cross-validation

Terms and Notation

- Scalar x, Vector $\mathbf{x}$ (bold), Matrix $\mathbf{A}$: $\mathbf{y} = (y_1, y_2, \cdots, y_N)$, $\mathbf{y} = \mathbf{A}\mathbf{x}$
- Gaussian (normal) distribution, and its sample :
  - $p(z) = \dfrac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\dfrac{(z-\mu)^2}{2\sigma^2}\right\} = \mathcal{N}(\mu, \sigma^2), \quad z \sim \mathcal{N}(\mu, \sigma^2)$
- Conditional probability of y given θ : $p(y \mid \theta)$
- Norm of a vector : $\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$, $\|\mathbf{x}\|_2 = \sqrt{x_1 + x_2 + \cdots + x_N}$
- $\mathbf{y} = f(\mathbf{x}; \theta)$: $\mathbf{y}$ target variables, $\mathbf{x}$ explanatory variables (, or predictors or features), θ model parameters

If you find any terms unclear later in this lecture, please refer back to this slide.

1-1 Bayes' theorem, posterior, prior, likelihood

Bayes' Theorem and its Cast

Posterior probability and Likelihood

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})}$$

- Posterior probability $p(\theta \mid \mathbf{y})$
  - The goal of Bayesian inference is to estimate the posterior distribution.
- Likelihood function $p(\mathbf{y} \mid \theta)$
  - joint probability of observing the data **y** given the parameters $\boldsymbol{\theta}$
  - In frequentist statistics, the best model is that maximizes the likelihood.

Bayes' Theorem and its Cast

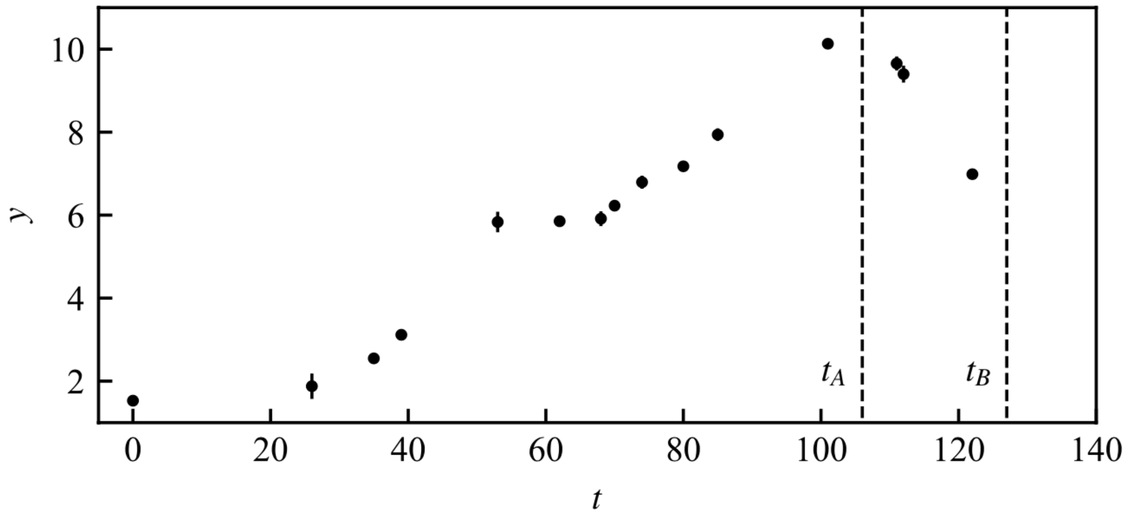
Prior probability

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})}$$

- Prior probability $p(\theta)$
  - A key component of Bayesian models
  - allows us to incorporate existing knowledge or assumptions about the parameters into the model.
  - helps to articulate relationships between parameters more clearly.

Ex. Smooth curve

- Time-series data (15 data points in t=0–140)
- Equally-spaced 141 μ
- Current slope is similar to the last slope
- Inputs: 15 Data (t_i, y_i)
- Parameters: 141 μ_i & λ
- Estimating the posterior of the parameters with MCMC (λ is given by hand)



$$p(y_i|\mu_j) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{-\frac{(y_i-\mu_j)^2}{2\sigma_i^2}\right\} \quad (\text{ただし、} t_i = t_j)$$

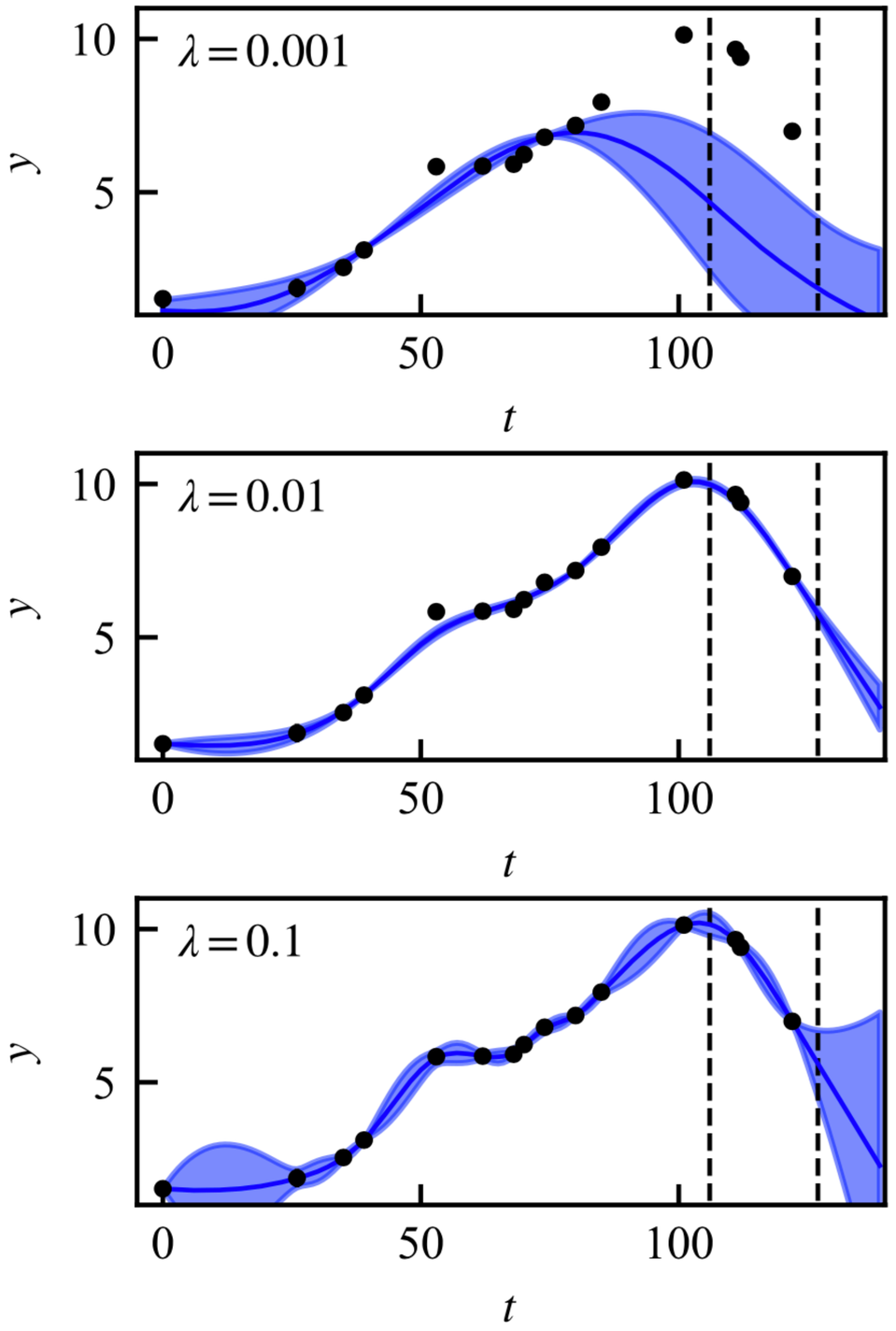
$$p(\mu_j) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left\{-\frac{((\mu_j-\mu_{j-1})-(\mu_{j-1}-\mu_{j-2}))^2}{2\lambda^2}\right\}$$

$$= \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left\{-\frac{(\mu_j-2\mu_{j-1}+\mu_{j-2})^2}{2\lambda^2}\right\}$$

Ex. Smooth curve: Result

$$p(\mu_j) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left\{-\frac{((\mu_j - \mu_{j-1}) - (\mu_{j-1} - \mu_{j-2}))^2}{2\lambda^2}\right\}$$
$$= \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left\{-\frac{(\mu_j - 2\mu_{j-1} + \mu_{j-2})^2}{2\lambda^2}\right\}$$

- For small λ
  - Too rigid & failing to capture the underlying trend in the data.
- For large λ
  - Too flexible → Overfit (see the next section)
- For moderate λ
  - Smooth and capturing the trend



Ex. Smooth curve: lessons learned

$$p(\mu_j) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left\{-\frac{(\mu_j - \mu_{j-1})^2}{2\lambda^2}\right\}$$

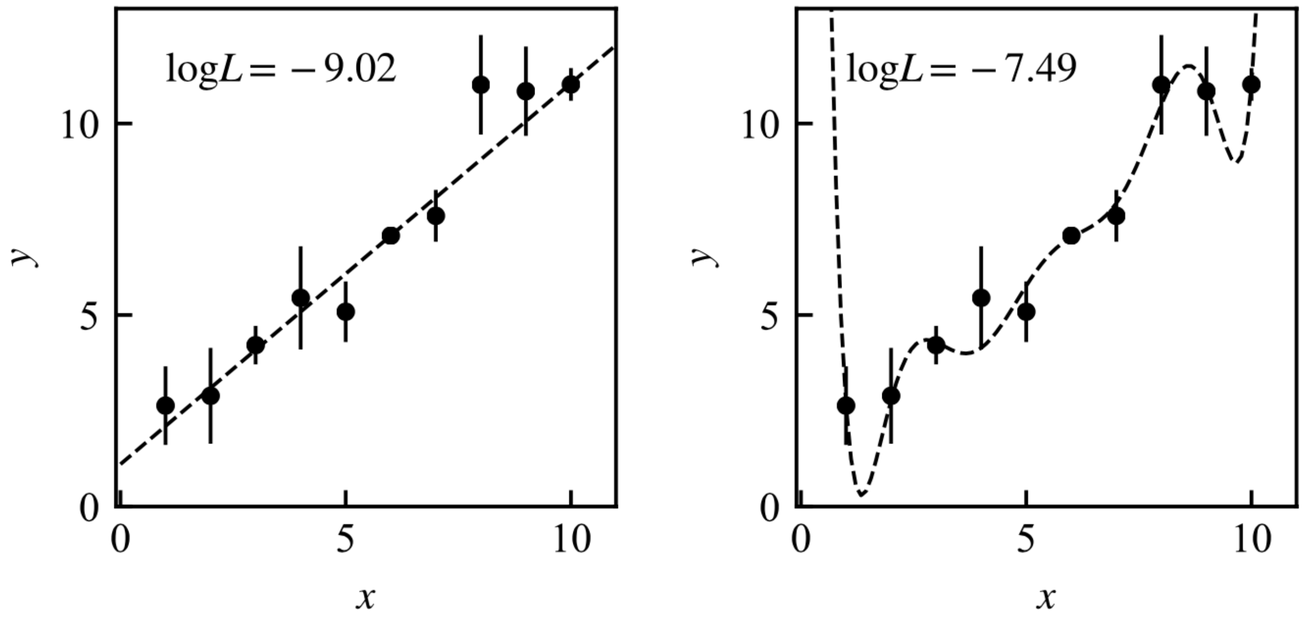
$$p(\mu_j) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left\{-\frac{((\mu_j - \mu_{j-1}) - (\mu_{j-1} - \mu_{j-2}))^2}{2\lambda^2}\right\}$$
$$= \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left\{-\frac{(\mu_j - 2\mu_{j-1} + \mu_{j-2})^2}{2\lambda^2}\right\}$$

- Expressing complex relationships between the model parameters through the prior.
- not lines or polynomials, but more nuanced trends.
- Local trend model
  - It can be extended to
    - Auto-regressive model = the coefficients of μ_{j-1}, μ_{j-2} are also estimated from the data.
    - State-space model

1-2 Generalization error, information criteria, cross-validation

Polynomial Regression, as an example

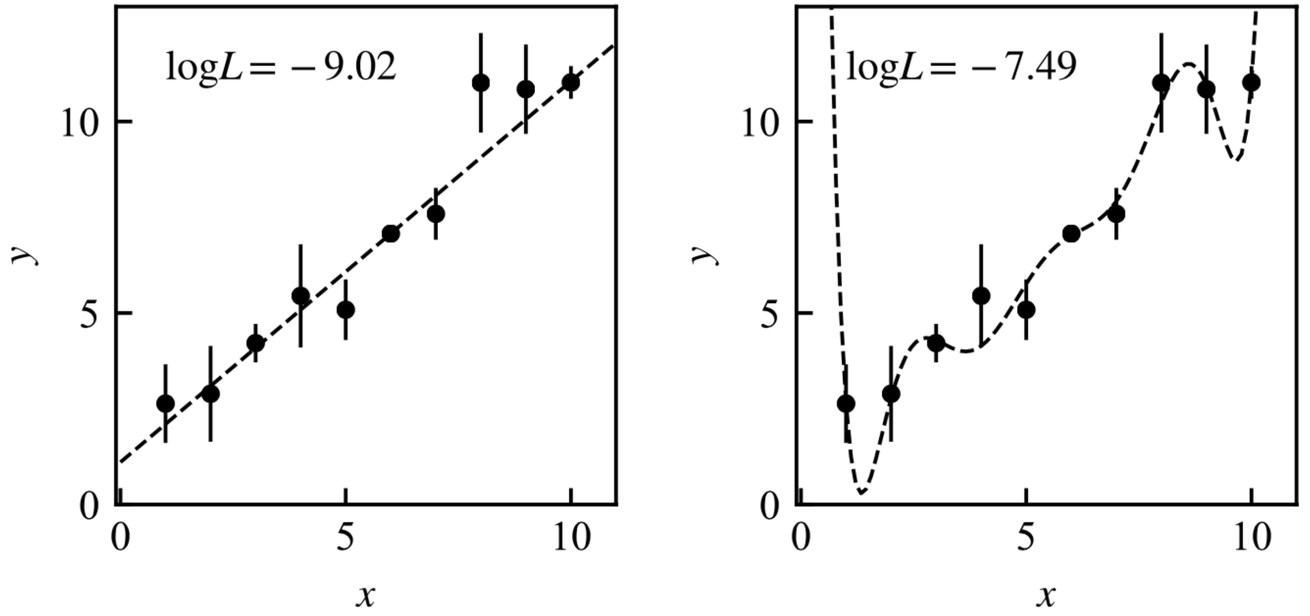
- Left panel: 1st degree = line
- Right panel: 8th degree polynomial
- The eighth-degree polynomial has a higher (log-)likelihood.
- Do you prefer the 8th order model?



Overfit and model selection

- Overfit = a statistical model describes random noise
  - The likelihood of a model often increases with the number of model parameters.
  - In high-dimensional problems, evaluating a model solely based on likelihood becomes inappropriate.
- Model selection:
  - If the model is too simple → it cannot adequately explain the data.
  - If the model is too complex → overfitting occurs.
  - It's essential to choose a model with the right level of complexity. = model selection
- Two approaches: Information criteria & cross-validation.

Overfit =
Low generalization error

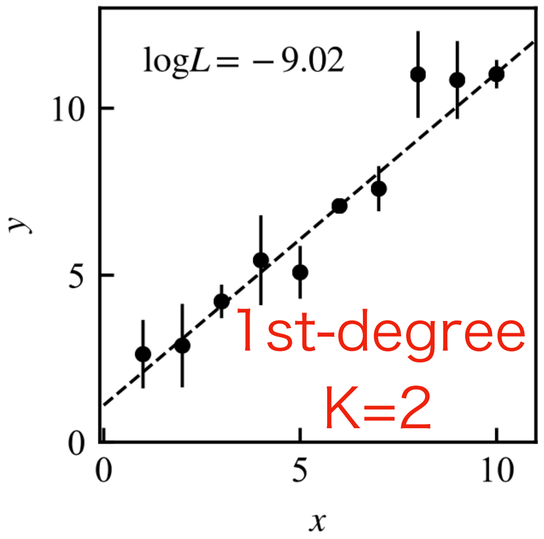


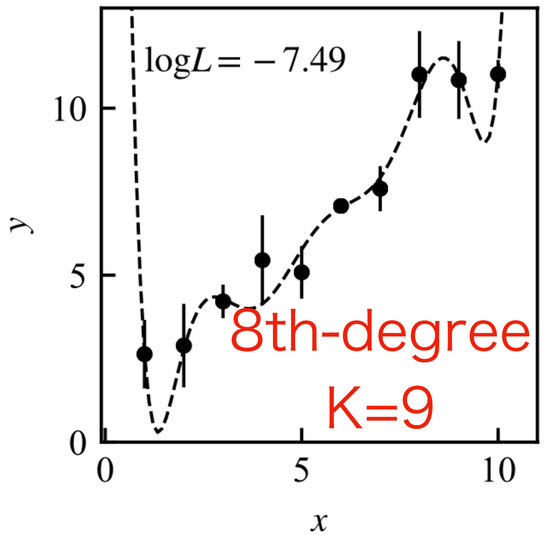
AIC（Akaike information criterion）

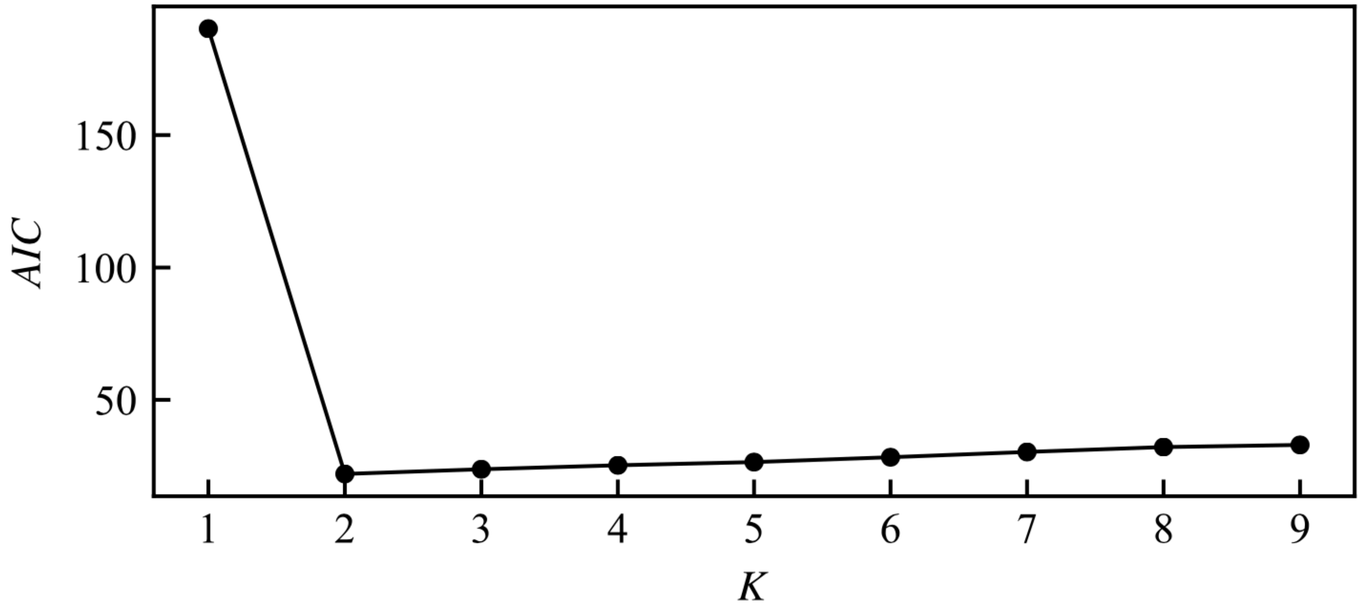
$$\mathrm{AIC} = -2(\log L(\hat{\boldsymbol{\theta}}) - K) = -2\log L(\hat{\boldsymbol{\theta}}) + 2K$$

- The log-likelihood $\log L(\hat{\theta})$ of the data can become biased when the model is too complex.
- To correct this bias, AIC introduces a penalty term, the number of model parameters K.
- Even if two models have the same likelihood, the model with fewer parameters — resulting in a smaller K — will have a lower AIC, and be considered the better model.
- Focusing not just on how well a model fits the current data, but also on how well it can predict new, unseen data. = model predictive performance or generalization performance.

AIC model selection for the polynomial fitting



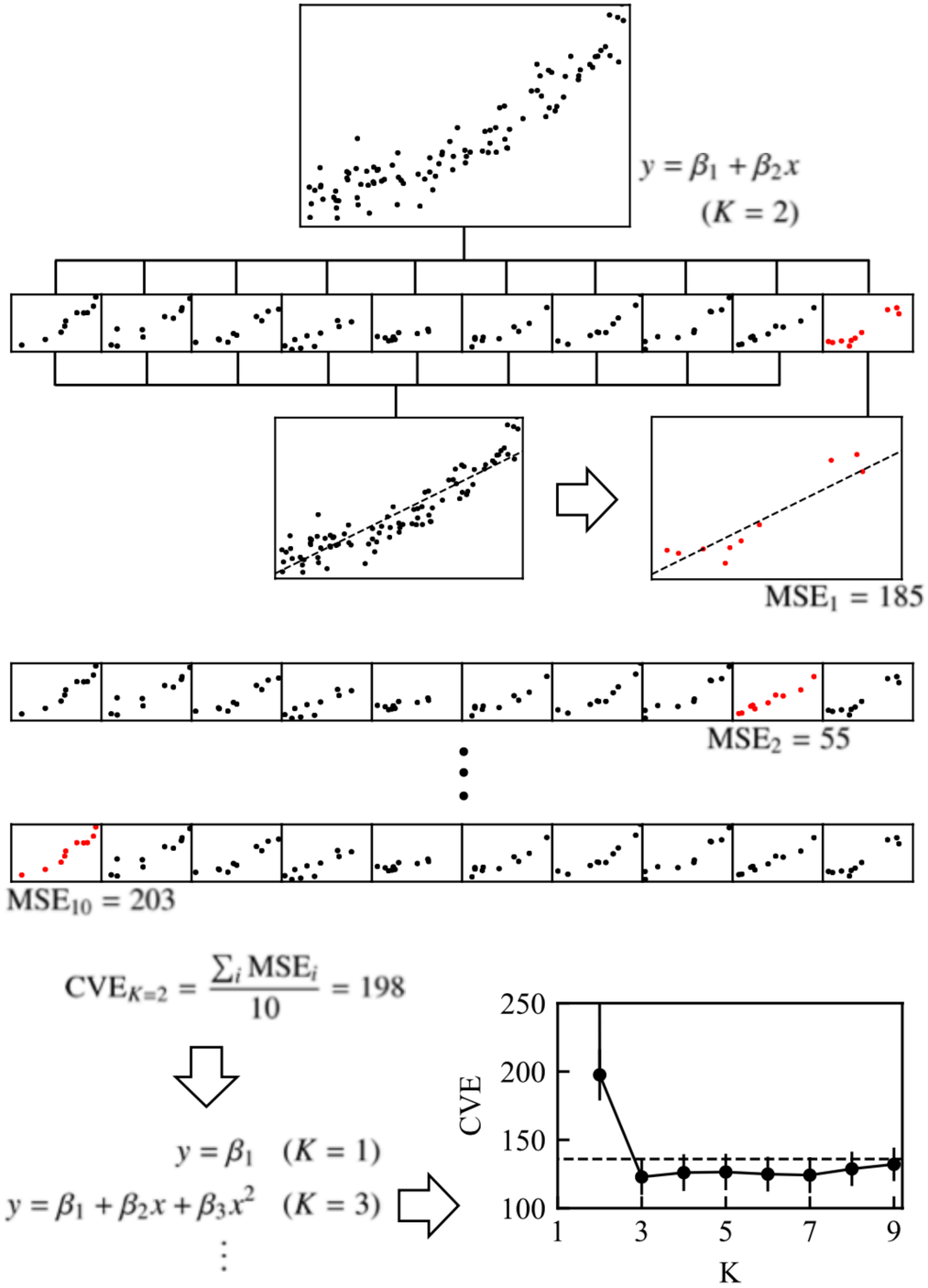




- Models are created → the model parameters are determined using maximum likelihood estimation → the likelihood is obtained → AIC can be calculated based on the number of model parameters (K)
- When K = 2 , which corresponds to a line, the AIC is minimized.
- The K=2 model provides the best balance between fitting the data and avoiding unnecessary complexity.

Cross-Validation (CV)

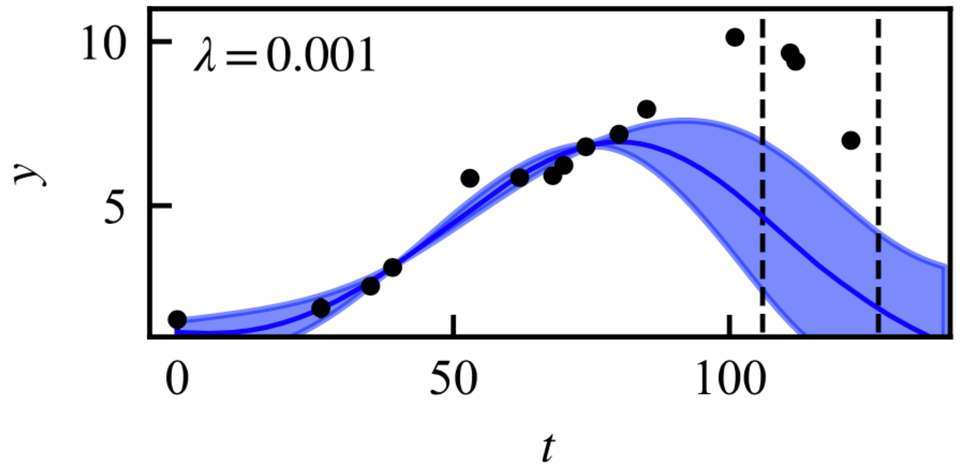
- K-fold cross-validation (right figure)
  - Divide the data into K equally-sized subsets (10 folds is common)
  - Use one subset k for validation and the remaining subsets for training
  - Obtain K metrics (such as squared errors or mean squared errors (MSE))
  - Calculate the average and standard error of the cross-validation errors (CVE)

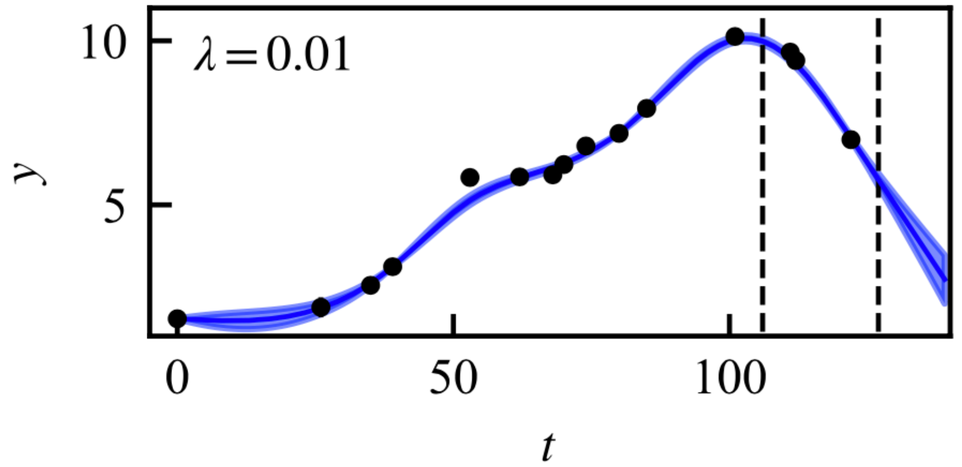


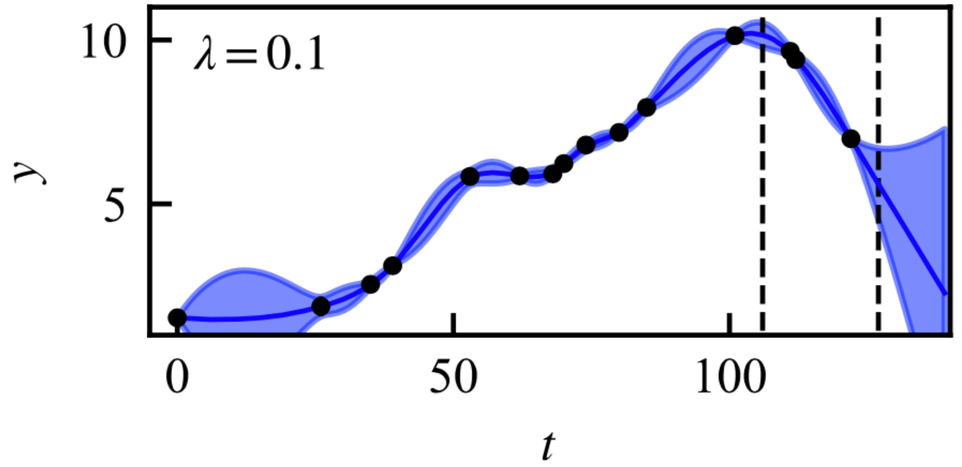
Hands-on exercise #1

Local trend model

Lecture_Day1_Uemura/01_Bayes.ipynb





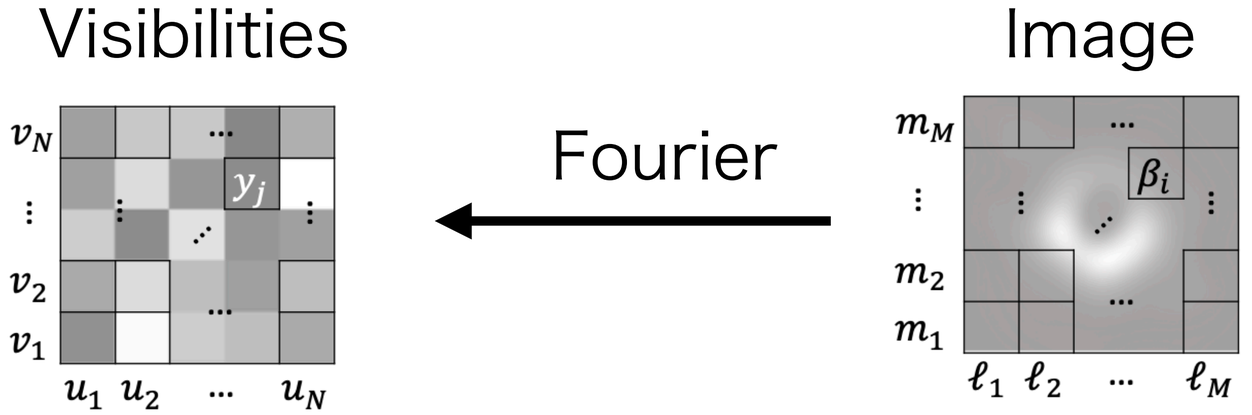


2. Regularization

Key words: LASSO (l1-penalty), sparsity

Image reconstruction in VLBI

- Data: Visibilities (complex numbers)
- Estimation: Brightness distribution on the celestial sphere = 2D image
- Matrix = Fourier transform
- Number of data points N < Number of coefficients M
  - Underdetermined system
  - = ill-posed problem

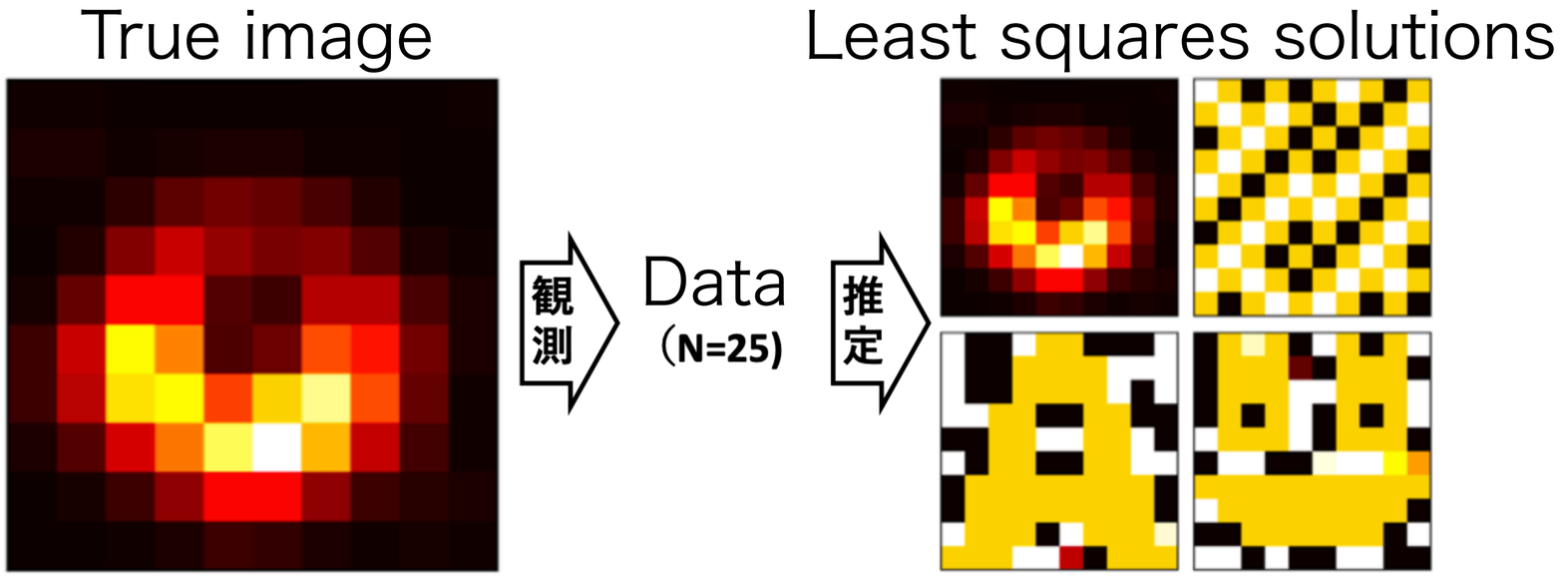


$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1M} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NM} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_M \end{pmatrix}$$

$$\boldsymbol{y} = \mathcal{F}\boldsymbol{\beta}$$

$$a_{ij} = \exp\{-2\pi i(u_i \ell_j + v_i m_j)\}$$

In maximum likelihood estimation, the solution is indeterminate



- Estimate the brightness of 100 pixels (left figure)
- Data = 25 complex visibilities = 50 pieces of information (real part + imaginary part or amplitude + phase)
- Results (right figure)
  - All four images are “least squares solutions” = perfectly match the data

- So, what should we do next?
  - Does it really make sense to claim that any of the other three are “astronomical images”? —> No !
  - Why do many people consider the other three to be impossible as astronomical images? —> Because they have prior knowledge about what astronomical images should look like.
  - Can we use that prior knowledge as a Bayesian prior probability to fill in the gaps left by the missing information?

Least Squares Method and Regularization

Basic terms

- Least Squares Method : $\min_{\theta} E(\theta)$
  - The problem of finding the parameter θ that minimizes the error function, $E(\theta) = \|y - f(\theta)\|_2^2$

- Regularized (Penalty) Least Squares Method : $\min_{\theta} E(\theta) + \lambda\Phi(\theta)$
  - $\lambda\Phi(\theta)$ is the regularization term
  - We aim to minimize this combined function.
  - λ is the regularization coefficient.
    - When λ is large, the method prioritizes minimizing the regularization term over fitting the data.
    - When λ is small, the regularization term has less influence, allowing the model to fit the data more closely.

Regularization and Bayesian Methods

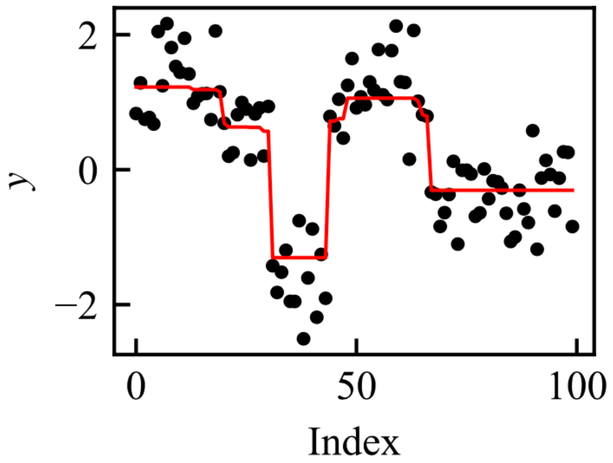
- If we think of the objective function in regularized least squares as the negative log-posterior probability:

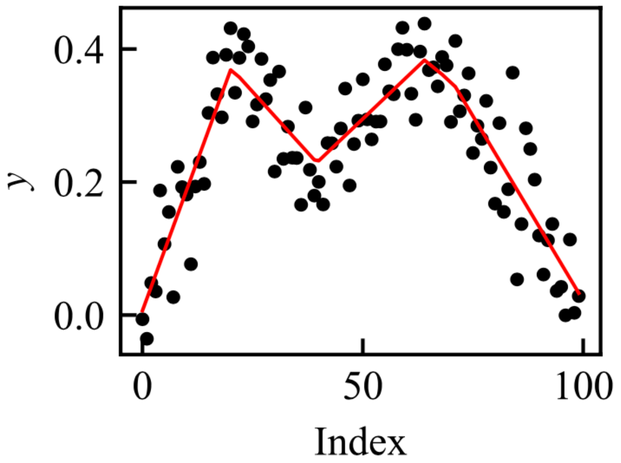
$$-\log p(\beta \,|\, \mathbf{y}) = \frac{1}{2} \sum_i \frac{(y_i - \mathbf{x}_i \beta)^2}{\sigma_i^2} + \lambda \Phi(\beta)$$

$$p(\boldsymbol{\beta}|\boldsymbol{y}) \propto \prod_i \exp\left\{ -\frac{(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta})^2}{2\sigma_i^2} \right\} \exp\{-\lambda \Phi(\boldsymbol{\beta})\}$$
$$\propto p(\boldsymbol{y}|\boldsymbol{\beta}) p(\boldsymbol{\beta})$$

- The regularization term corresponds to the prior distribution.
- The form of the regularization term reflects our prior knowledge about the data.

Various regularizers



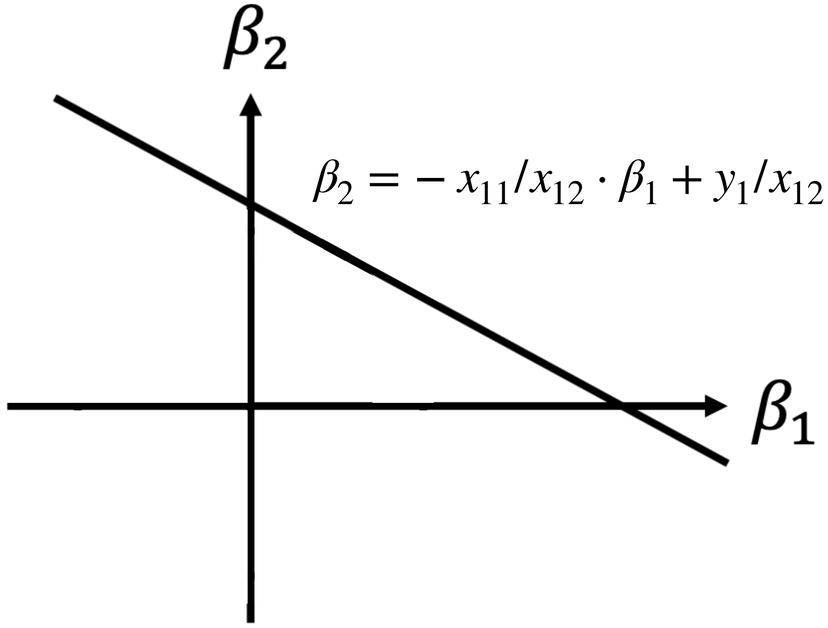


- Ridge regression : $\lambda\|\beta\|_2^2$
- LASSO regression : $\lambda\|\beta\|_1 \rightarrow$ sparse solution (next slides)
- Elastic net : $\lambda(\alpha\|\beta\|_1 + (1-\alpha)\|\beta\|_2^2)$
  - By adjusting the two regularization parameters λ and α, Elastic net can sometimes yield better results than LASSO.
- Total variation minimization : $\lambda \sum_i |\beta_{i+1} - \beta_i|$
  - l1 norm in differential space.
- ℓ_1 trend filter : $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{\beta}\|_2^2 + \lambda \sum_i |\beta_{i+2} - 2\beta_{i+1} + \beta i|$
  - sparse in second-order difference
  - Represent the data with the same line as much as possible, estimating points of trend changes.
- Group LASSO : $\lambda \sum_k \|\boldsymbol{\beta}^{(k)}\|_2$

  $\{\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \cdots, \boldsymbol{\beta}^{(K)}\}$
- Maximum Entropy Method (MEM) : $-\lambda \sum_i \beta_i \log \beta_i$
  - Classical regularizer often used in physics.

LASSO regression:
A simple example of undetermined problems

- Consider a low-dimensional problem that is easy to visualize
- Two model parameters (β_1, β_2)
- One data point = One equation $y_1 = \beta_1 x_{11} + \beta_2 x_{12}$
- Estimate (β_1, β_2)
- All points on the line, $\beta_2 = -x_{11}/x_{12} \cdot \beta_1 + y_1/x_{12}$, represent valid solutions = Underdetermined



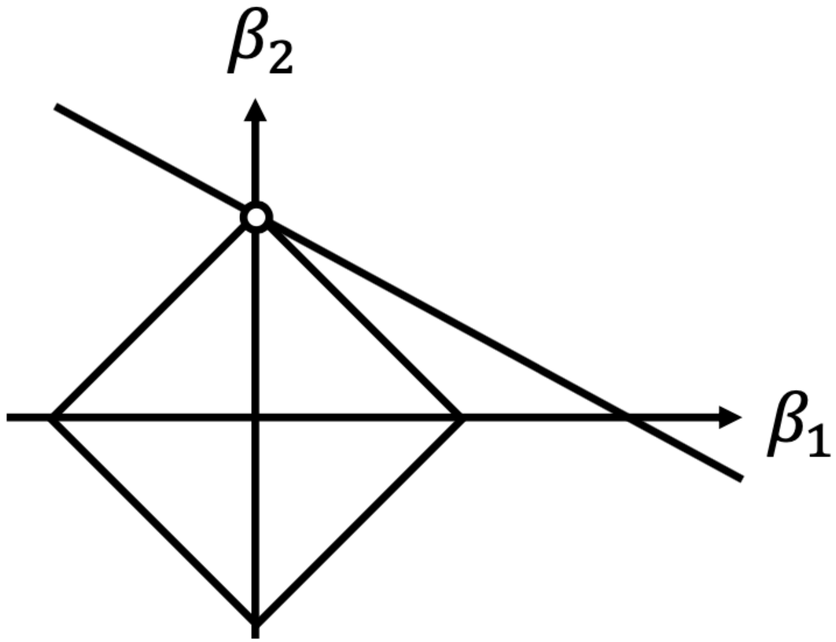
LASSO regression
(Least Absolute Shrinkage and Selection Operator; Tibshirani 1996)

- Enforces sparsity in the coefficient vector $\boldsymbol{\beta}$ by using the L1 norm for regularization

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}$$

- Constant L1 norm:

  $\|\boldsymbol{\beta}\|_1 = |\beta_1| + |\beta_2| + \cdots + |\beta_M| = \text{const.}$ is a “diamond” shape in 2D and an octahedron in 3D



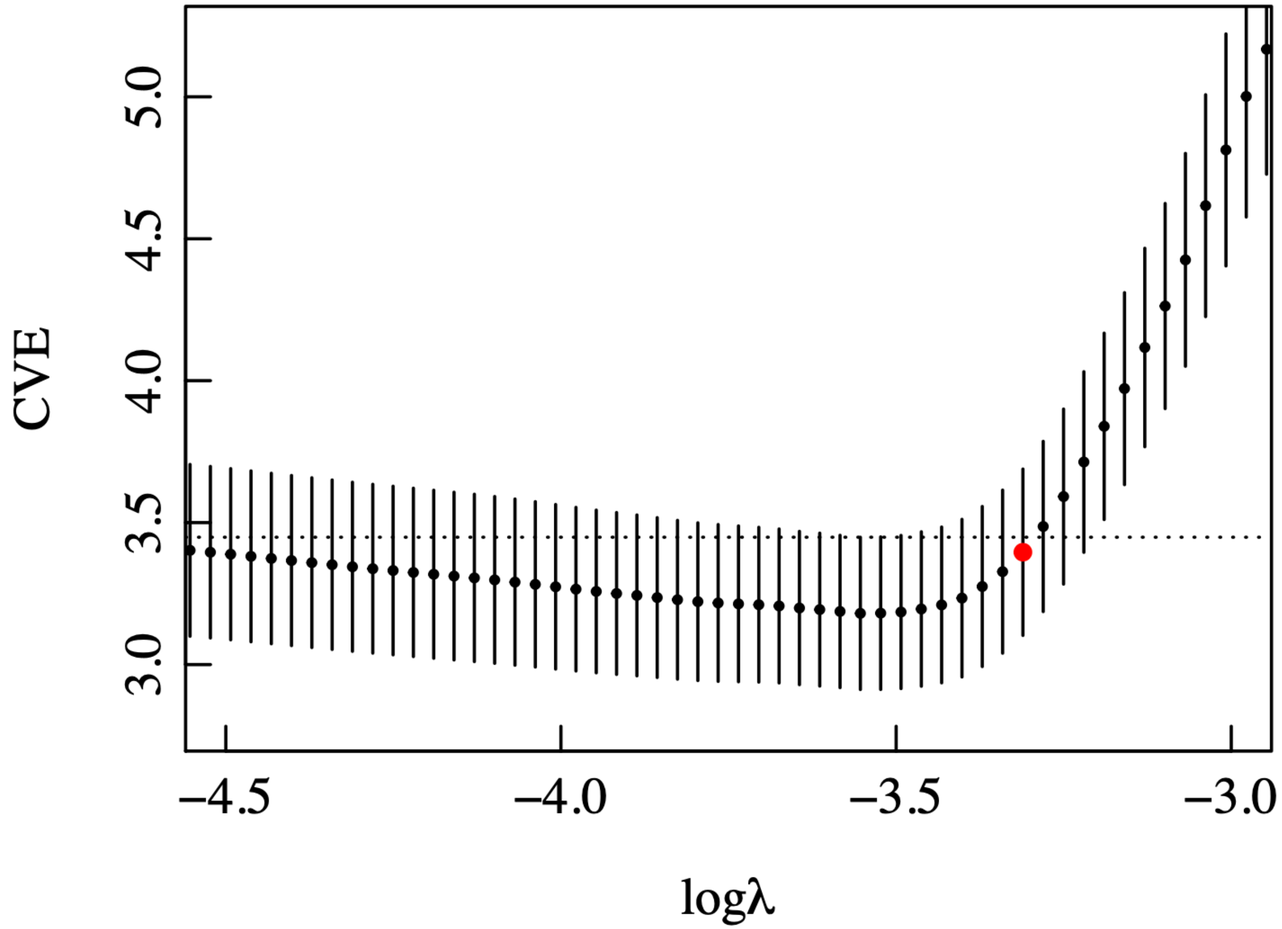
- Powerful in situations where we have more features than observations and when we believe that $\boldsymbol{\beta}$ is sparse = only a few features are truly important.

- The level of sparsity is determined by λ.

The optimal model is determined by λ

Choosing λ correctly is crucial

- Choose λ that maximizes the model's generalization performance.
- Cross-validation is a straightforward and widely used method.
- Right figure: Example from a practical case discussed later
  - If λ is too large, the resulting model may become excessively sparse, which means it oversimplifies the data, leading to a large discrepancy between the model and the actual data.
  - if λ is too small, the model may overfit, capturing noise in the data, and thus reducing its ability to generalize to new, unseen data.



Estimate the period of a variable star from time-series data

- Estimate a sparse power spectrum from data.
- The observed time-series data **y** is modeled as a linear combination of cos and sin components at each frequency.
- A total of 2M components (both sin and cos) for M different frequencies.
- Number of data points = N
- According to the sampling theorem, if N = 2M , the problem is uniquely solvable.
- If some data points are missing or the sampling intervals are non-uniform → Underdetermined system with N < 2M

$$\begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \cos(2\pi t_1 \nu_1) & \cdots & \cos(2\pi t_1 \nu_M) \\ \vdots & \vdots & \vdots \\ \cos(2\pi t_N \nu_1) & \cdots & \cos(2\pi t_N \nu_M) \\ \sin(2\pi t_1 \nu_1) & \cdots & \sin(2\pi t_1 \nu_M) \\ \vdots & \vdots & \vdots \\ \sin(2\pi t_N \nu_1) & \cdots & \sin(2\pi t_N \nu_M) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_M \\ b_1 \\ \vdots \\ b_M \end{pmatrix}$$

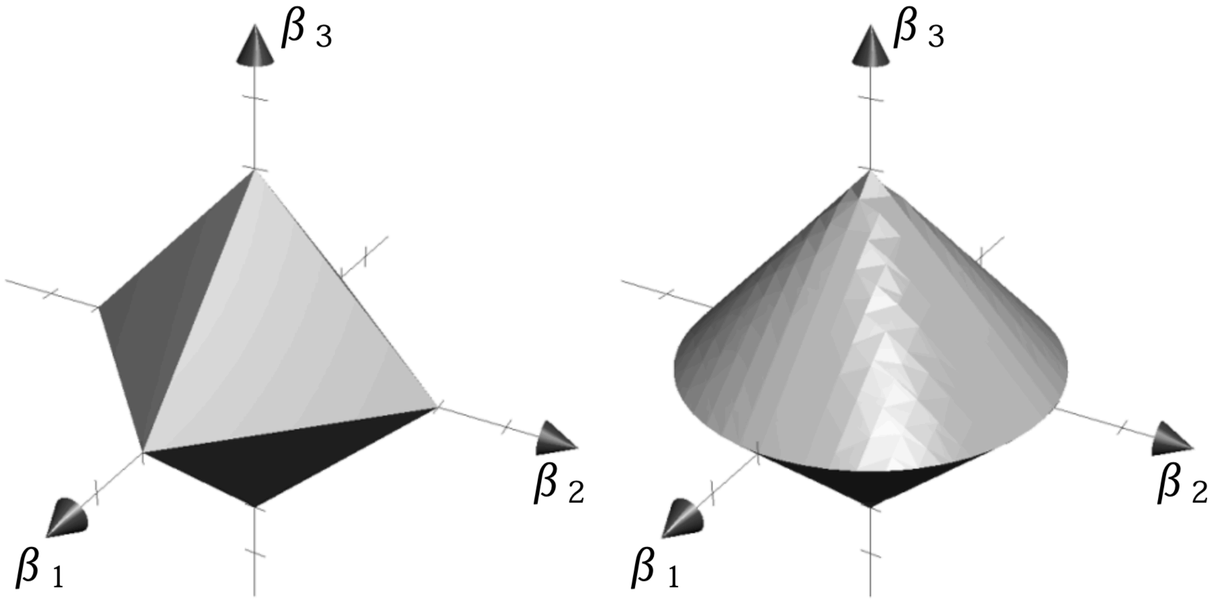
$$\boldsymbol{y} = \mathcal{F}^{-1}\boldsymbol{\beta}$$

Solve using Group LASSO

- Assume that the power spectrum is sparse.
- Using standard LASSO, the sin and cos coefficients for each frequency would be treated independently (left figure).
- With Group LASSO, coefficients are grouped by frequency.
  - Bottom-right figure: $\beta 1$ and $\beta 2$, representing the sine and cosine components for a particular frequency, are grouped together.
  - Group LASSO enforces sparsity at the group level, meaning that either the entire group (both $\beta 1$ and $\beta 2$) or another coefficient, such as $\beta 3$, is set to zero.

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \mathcal{F}^{-1}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$$

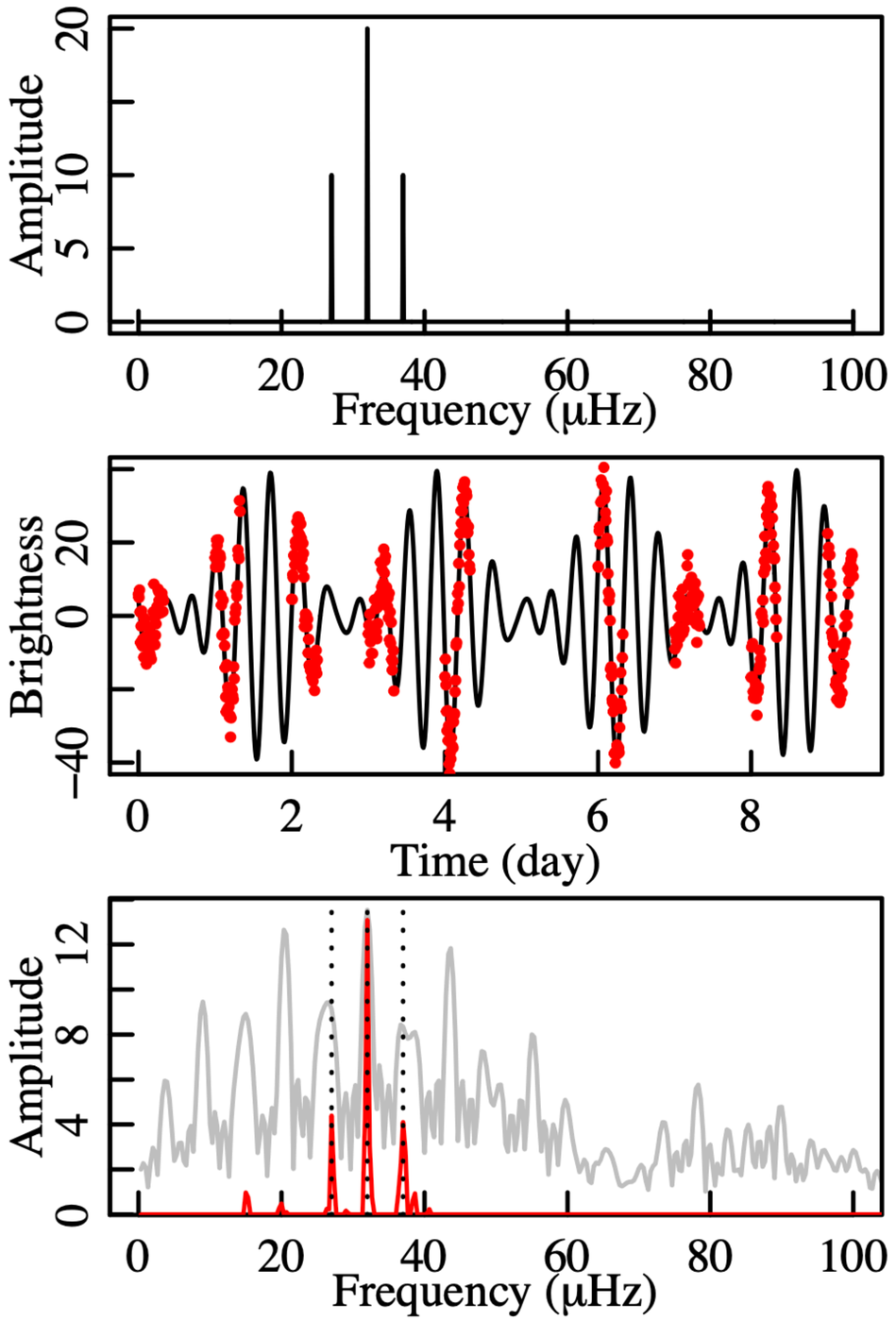
$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \mathcal{F}^{-1}\boldsymbol{\beta}\|_2^2 + \lambda\sum_i \sqrt{a_i^2 + b_i^2}$$



$$\sqrt{\beta_3^2} + \sqrt{\beta_1^2 + \beta_2^2}$$

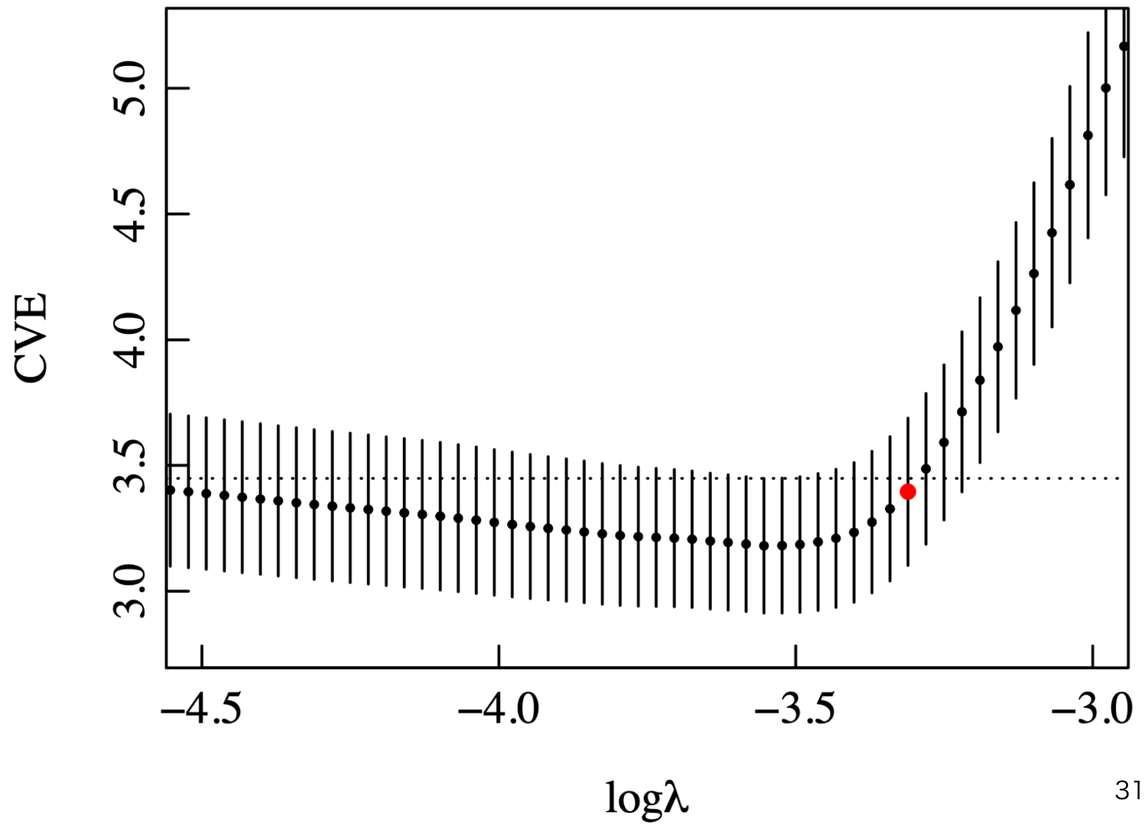
Experiments

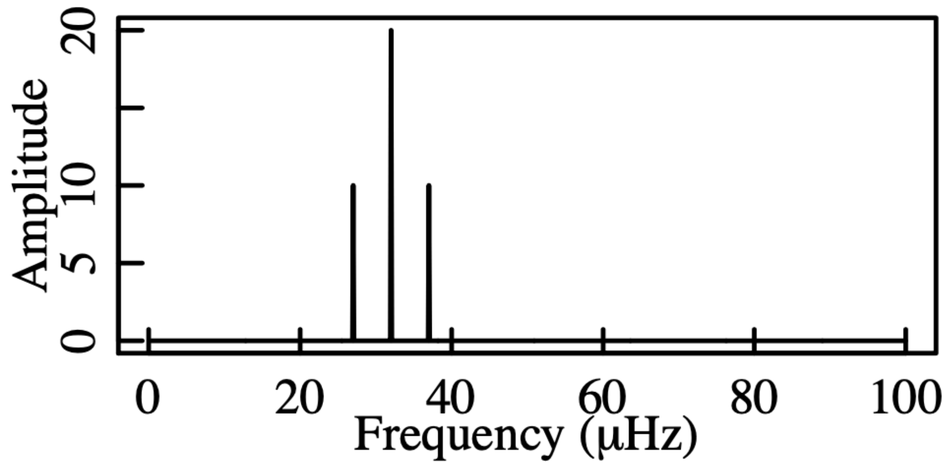
- Assume three distinct frequency signals (top panel)
- Perform an inverse Fourier transform to create a time series dataset (black line in the middle panel)
- Downsample the data and add Gaussian noise (red dots)
- Perform a standard Fourier transform with the red dots → the gray power spectrum (bottom panel)
  - While the strongest signal is detected, the other weaker signals are buried under aliasing
- Perform a group LASSO regression → the red curve
  - Aliasing is significantly reduced, and the true signals become much clearer.

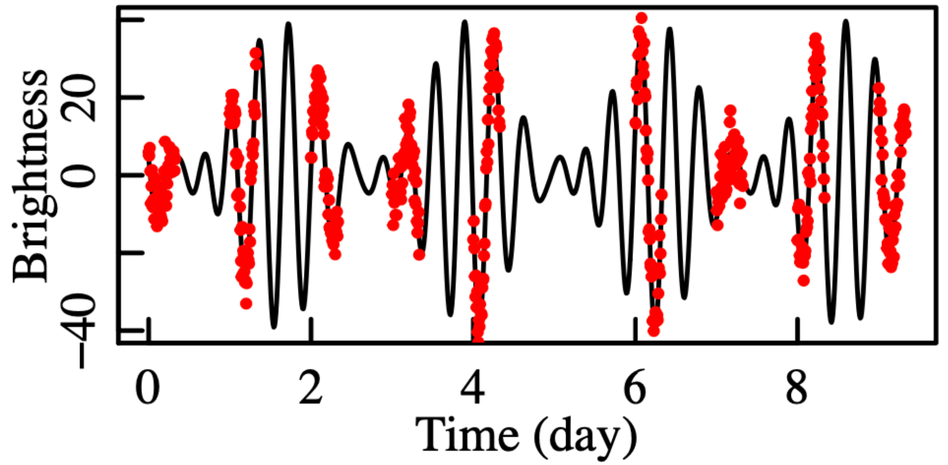


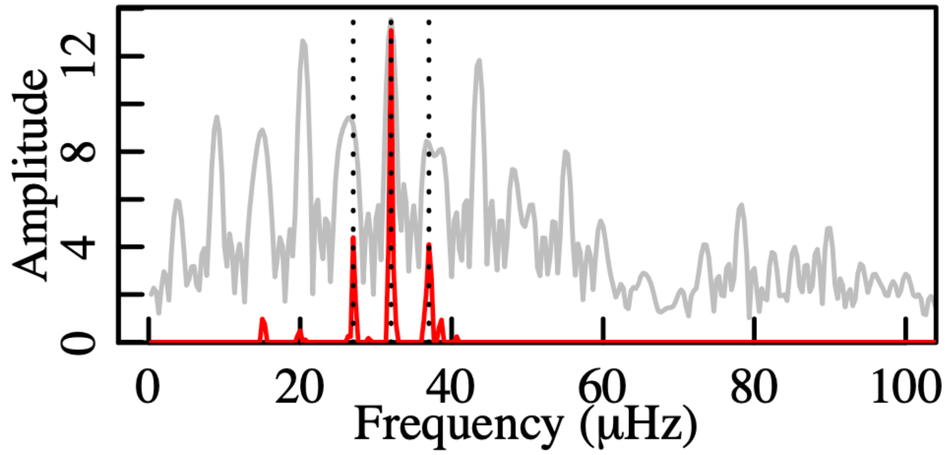
Experiments

- The regularization parameter λ is determined using cross-validation and the one-standard-error rule



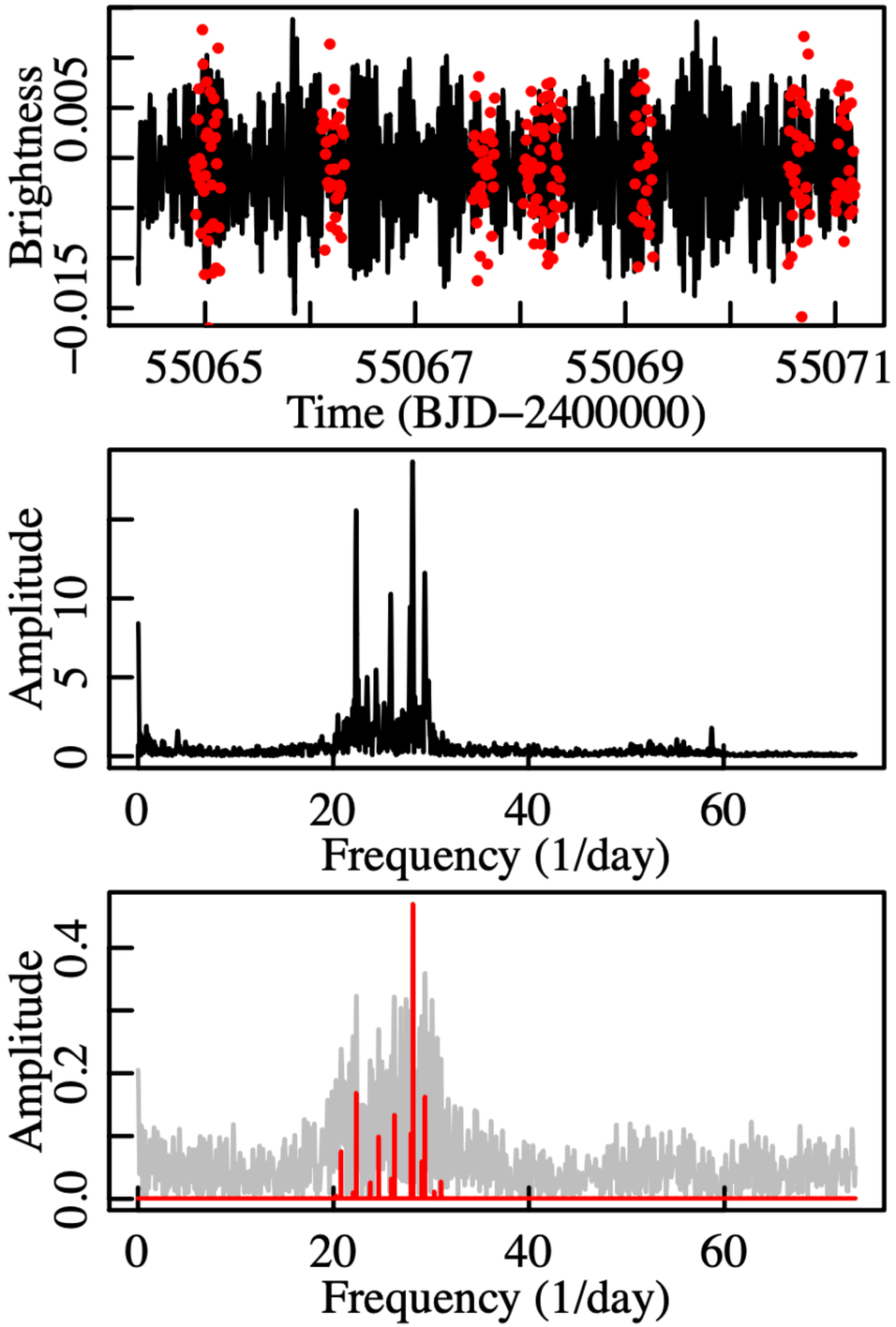






Apply to real data

- δ Sct type pulsating star observed with the Kepler satellite (black line in the top panel)
  - Nearly complete data. Obtaining such comprehensive data from ground-based observations is challenging.
  - The power spectrum is nearly complete (middle panel)
- Simulating ground-based observations, downsample the data and add noise (red dots in the top panel)
- Performing a standard Fourier transform results in the gray line in the bottom panel → Full of aliasing artifacts
- Results from Group LASSO (red line in the bottom panel)
  - Group LASSO effectively detects the strong signals present in the original power spectrum.



Event Horizon Telescope, again

as a regularized least-squares

©EHT

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left[\frac{1}{2} \| \boldsymbol{v} - F\boldsymbol{x} \|_{\ell_2}^2 + \lambda_1 \| \boldsymbol{x} \|_{\ell_1} + \lambda_{\mathrm{TSV}} \mathrm{TSV}(\boldsymbol{x}) \right],$$

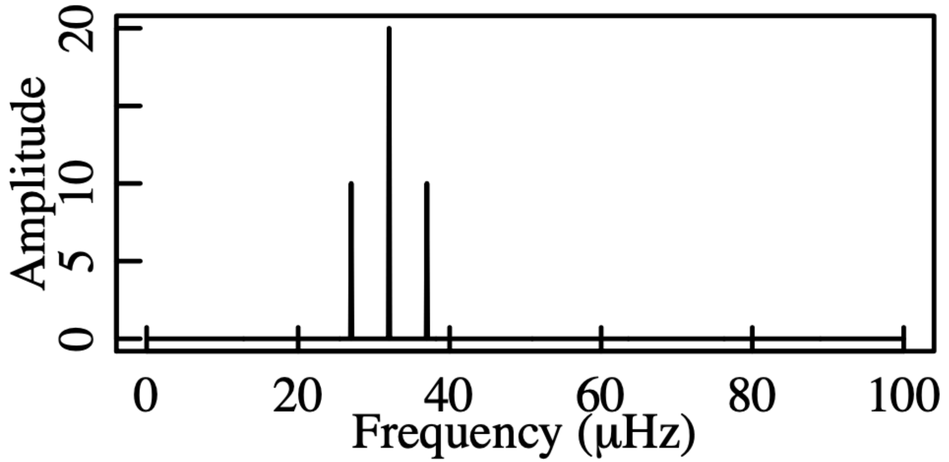
$$\mathrm{TSV}(\boldsymbol{x}) = \sum_{ij} \left[(x_{ij} - x_{i,j-1})^2 + (x_{ij} - x_{i-1,j})^2 \right].$$

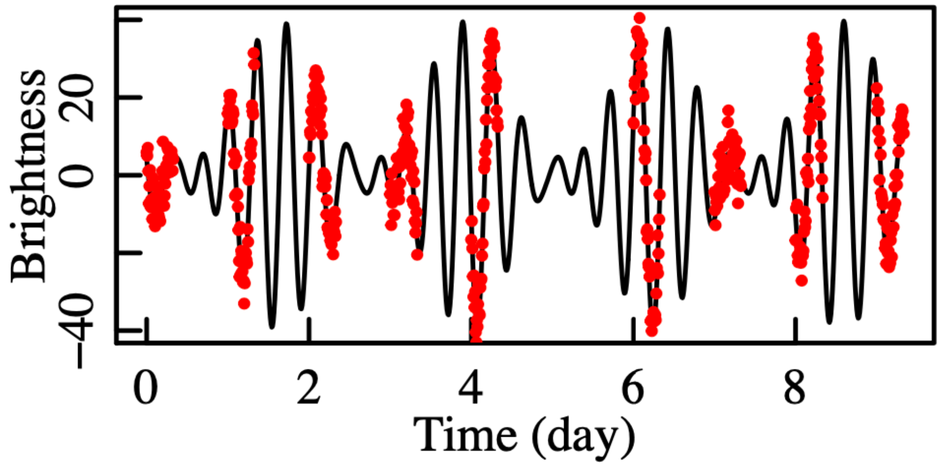
Event Horizon Telescope Collaboration (2019)

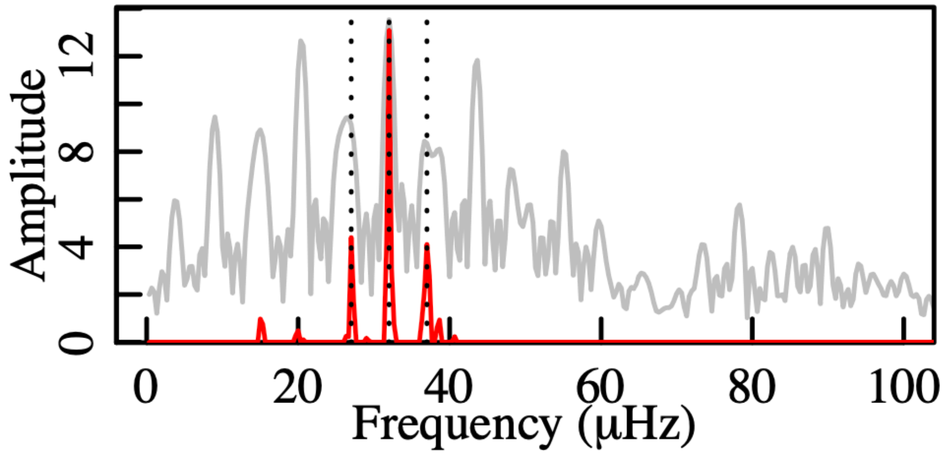
Hands-on exercise #2

Power spectrum estimation with LASSO

Lecture_Day1_Uemura/02_LASSO.ipynb





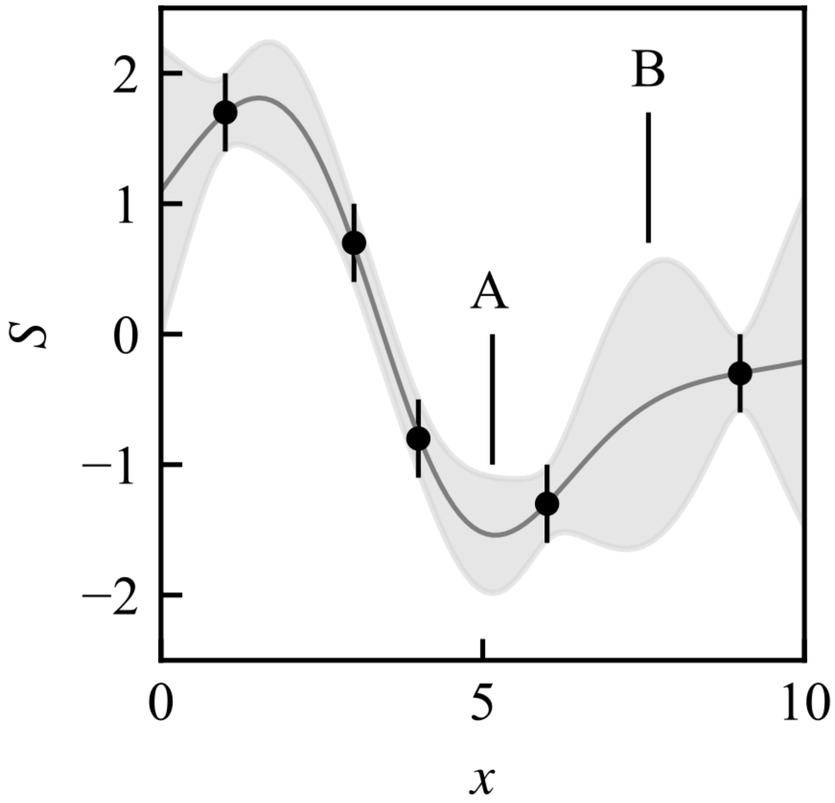


3. Gaussian processes

Key words: Kernel function, Bayesian optimization

What’s Gaussian processes (GP) useful for

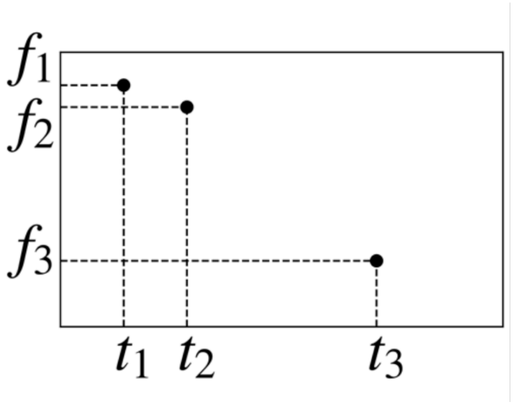
- Providing plausible curves (or surfaces) when the functional form of the data is unknown
  - Useful for interpolation tasks
- Also offering plausible uncertainty regions
  - Useful for finding the maxima and minima of functions and constructing surrogate models
- Computationally efficient
  - Relatively straightforward to implement from scratch
  - Python provides convenient modules for working with GPs, such as GPy and sklearn.gaussian_process.

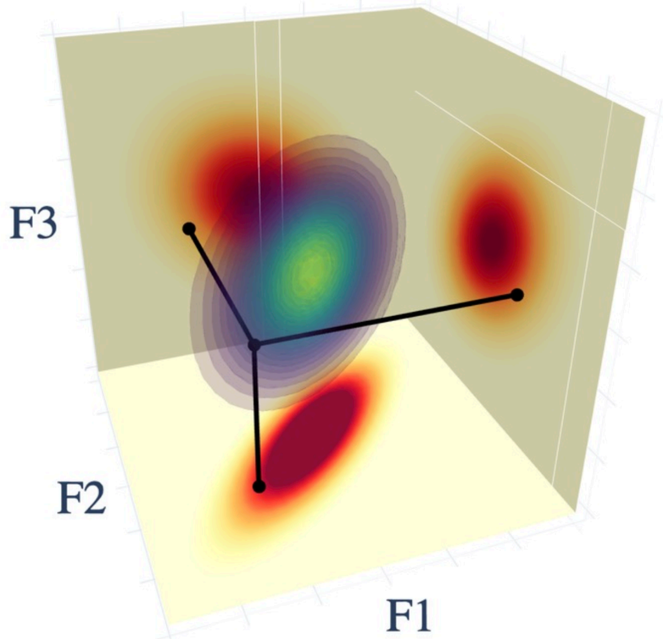


Example of Gaussian Process regression. If you want to find the minimum value of the function, you should measure at point “A” next. If you want to create a more accurate surrogate model, you should measure at point “B” next.

Definition of Gaussian Processes

Consider the N data points as a sample from an N-dimensional normal distribution





$$p(\boldsymbol{z}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\right\}$$

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

- Set the mean μ to zero by data preprocessing
- You need to define the covariance matrix = a core of GPs.
- Example: When there are 3 data points

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}$$

  - Each element is a function of $\sigma_{12} = k(t_1, t_2)$

- If there are N data points, you need N(N-1)/2 elements → Too many parameters

Kernel functions

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}$$

- Replacing covariances to the kernel functions: $\sigma_{ij} = k(x_i, x_j)$
- RBF (Gaussian) kernel
  - The distance between two points $\mathbf{x}_i, \mathbf{x}_j$ ~ their correlation.
  - Powerful for capturing smooth and continuous relationships in the data.
  - $\theta 2$ = length-scale
- Estimating a few kernel parameters from N data points → Gaussian Process Regression

RBF kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta_1 \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{\theta_2}\right\}$

Exponential kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta_1 \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2}{\theta_2}\right\}$

Periodic kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta_1 \exp\left\{-\theta_2 \cos\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2}{\theta_3}\right\}$

Linear kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta_1 \boldsymbol{x}_i^T \boldsymbol{x}_j$

Matern kernel $K(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|x-x'|}{\rho}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}|x-x'|}{\rho}\right)$

K ν is the modified Bessel function of the second kind

Gaussian Process Regression

Estimation of the kernel parameters from data

Likelihood $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I})$

Prior $p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$

Marginalized likelihood

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}) d\boldsymbol{f}$$

$$= \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}) = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{C}|}} \exp\left\{ -\frac{1}{2} \boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y} \right\}$$

$$\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}$$

Log-likelihood $\log p(\boldsymbol{y}) = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\boldsymbol{C}| - \frac{1}{2} \boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y}$

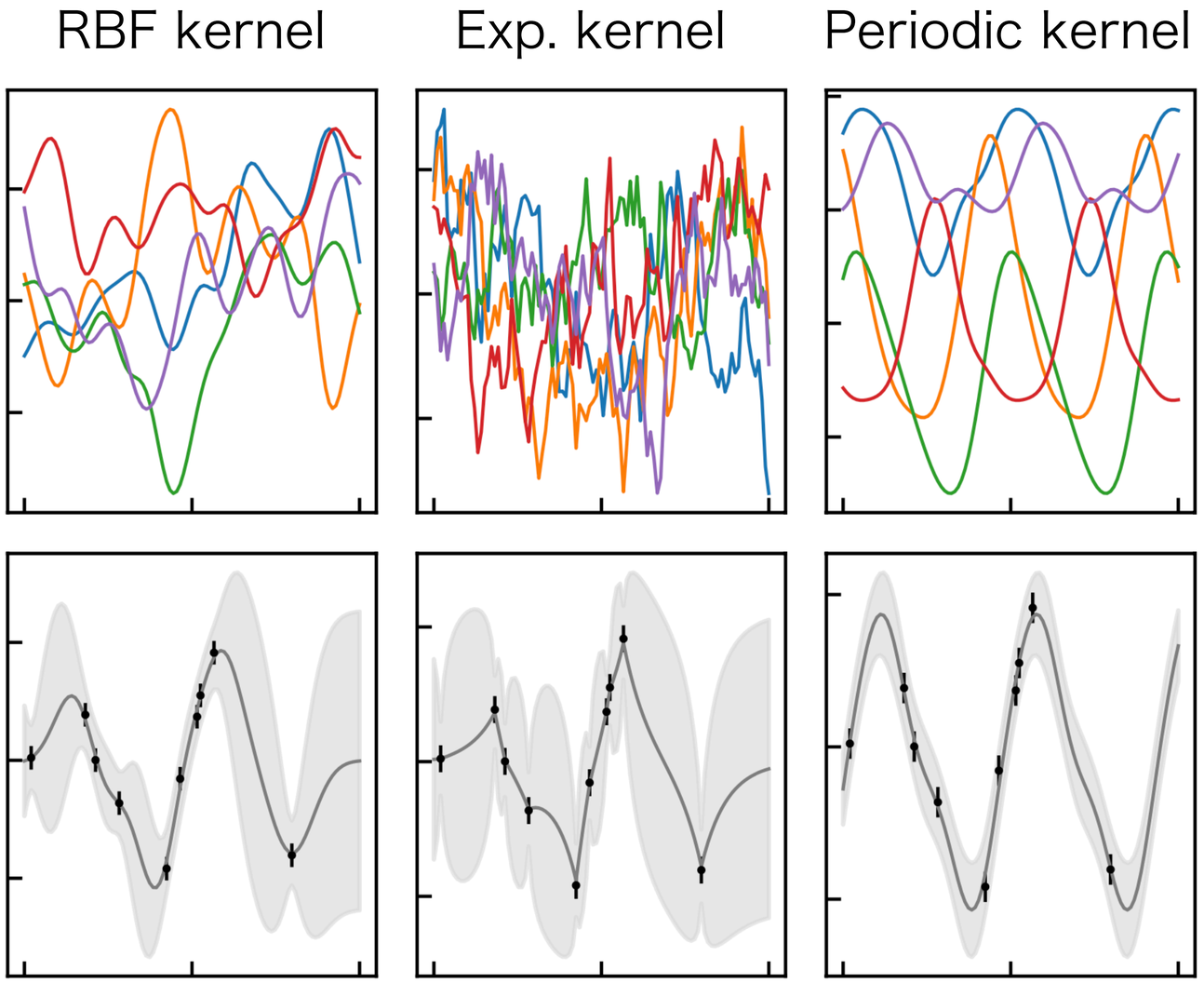
- A Bayesian model where a sample **f** from a Gaussian process defined by the covariance matrix **K** is subject to measurement error σ.
- Marginalize over **f**.
  - Even with Gaussian noise added to the Gaussian process, it remains a Gaussian process.
- Maximize the log (marginal) likelihood using gradient methods or similar → Determines the kernel parameters.

Prior and Posterior of kernels

$$p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I})$$

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$$

- **Top:** Samples from a Gaussian process with a given set of kernel parameters. = Prior $p(\mathbf{f})$
- **Bottom:** Predictions under the conditions where data has been obtained. = Posterior $p(\mathbf{f}|\mathbf{y})$



Prediction of y* at x*

- We want to know $p(y_*|\boldsymbol{y})$
- Joint probability follows the same Gaussian process.

Joint Prob. $p(\boldsymbol{y}' = (\boldsymbol{y}, y_*))$

Same GP $p(\boldsymbol{y}') = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}')$

$$\boldsymbol{C}' = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^T & c \end{pmatrix} \quad \boldsymbol{k}_*^T = (k(\boldsymbol{x}_*, \boldsymbol{x}_1), k(\boldsymbol{x}_*, \boldsymbol{x}_2), \cdots, k(\boldsymbol{x}_*, \boldsymbol{x}_N)) \quad c = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$$

- Use a theorem for the conditional multivariate normal distribution.

Theorem for the conditional multivariate Gaussian

$$\begin{pmatrix} \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}^{(1)} \\ \boldsymbol{\mu}^{(2)} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} \right)$$

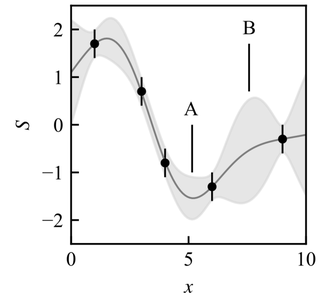
- The predictive distribution of y* is also a normal distribution. The mean and variance can be obtained through simple matrix calculations.

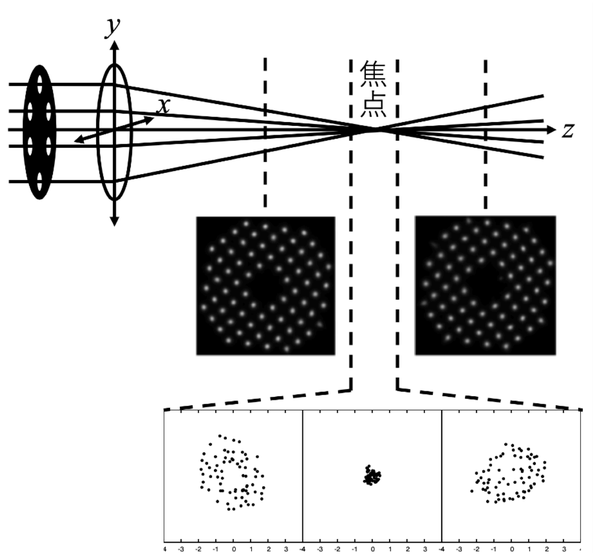
$$p(\boldsymbol{y}^{(2)}|\boldsymbol{y}^{(1)}) = \mathcal{N}(\boldsymbol{\mu}^{(2)} + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\boldsymbol{y}^{(1)} - \boldsymbol{\mu}^{(1)}), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12})$$

Prediction $\boxed{p(y_*|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{k}_*^T\boldsymbol{C}^{-1}\boldsymbol{y}, c - \boldsymbol{k}_*^T\boldsymbol{C}^{-1}\boldsymbol{k}_*)}$

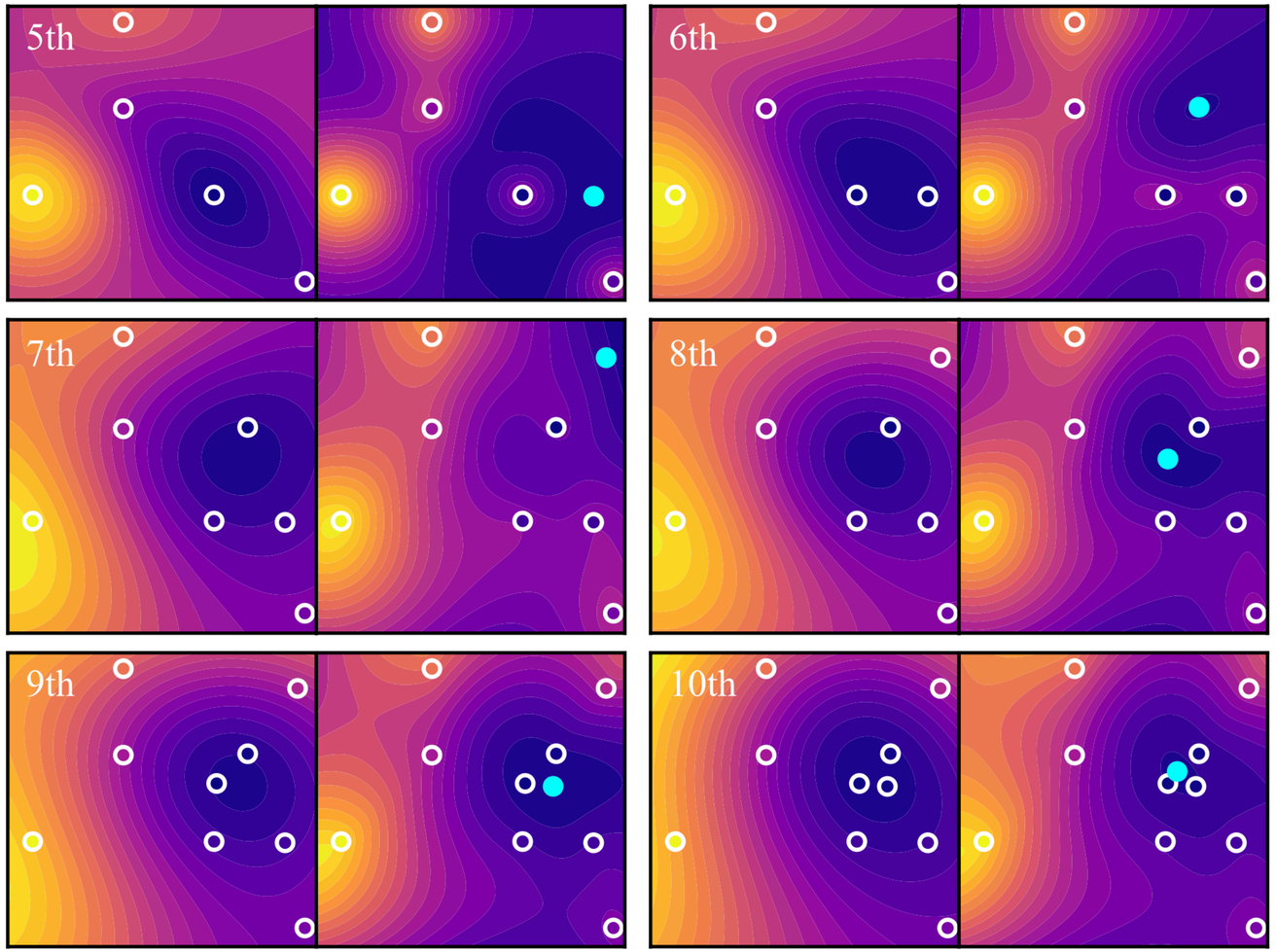
Practical Example 1

Hartmann test for the Kanata telescope



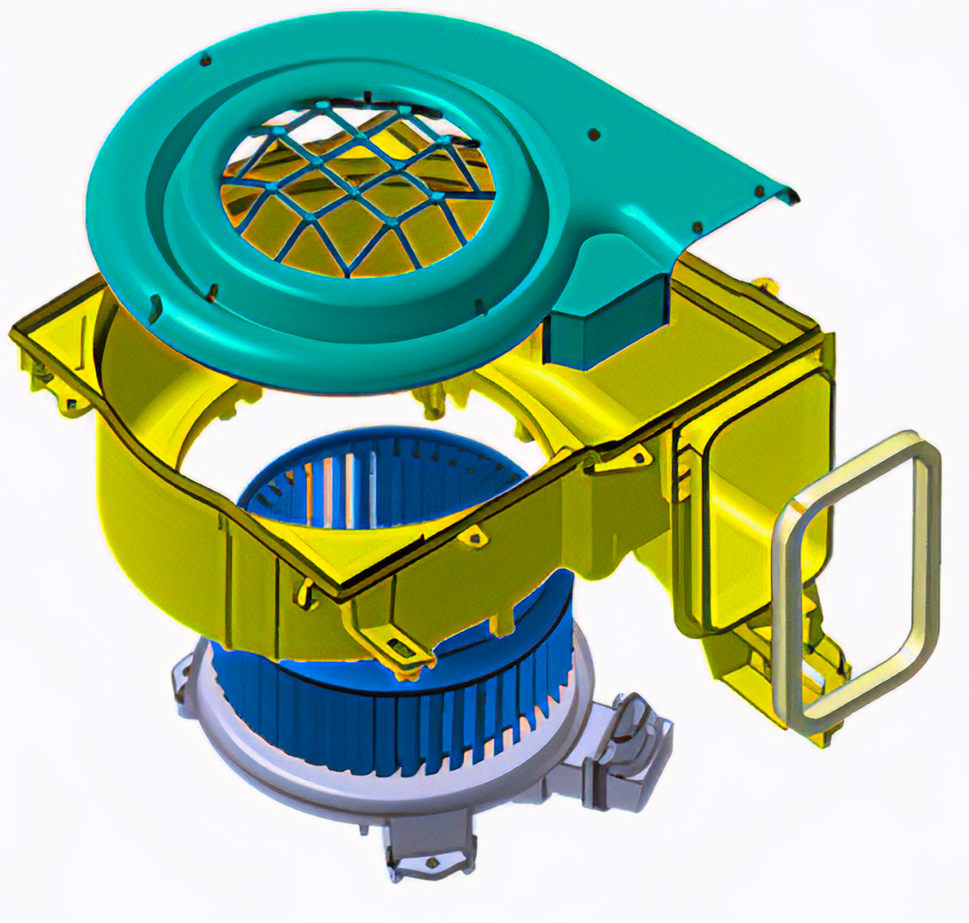


- Find the optimal (x, y) position of the secondary mirror where the light converges most effectively.
- Exhaustive search in the 2D space of (x, y) is time-consuming.
- Use “Bayesian optimization” with a Gaussian process to perform the search.



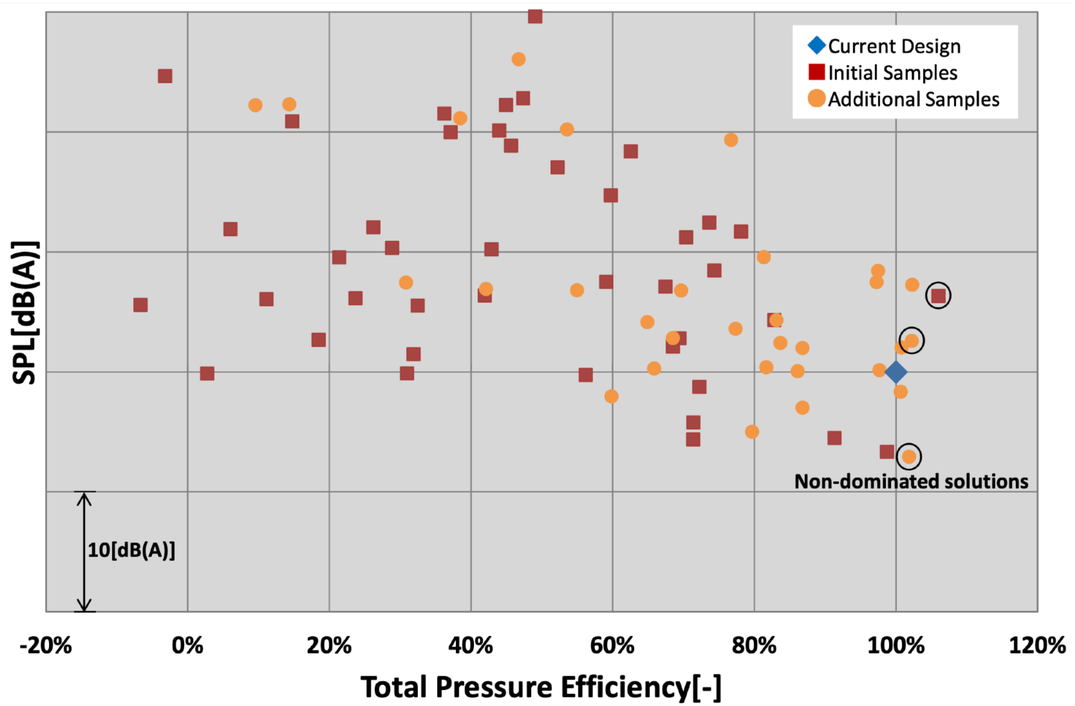
Practical Example 2 : Car air conditioner design

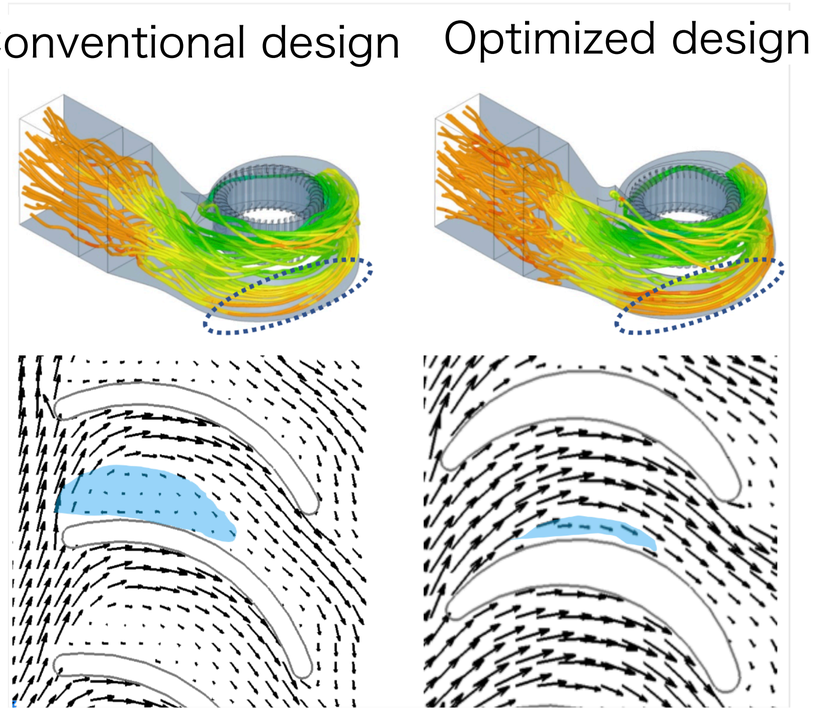
Masaru Kamada, Koji Shimoyama, Fumito Sato, Junya Washiashi, and Ya- sufumi Konishi. Multi-objective design optimization of a high efficiency and low noise blower unit of a car air-conditioner. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 233 (13):3493–3503, 2019.



- Determine the optimal design that maximizes airflow performance while minimizing noise.
- Perform fluid simulations → Evaluate performance
- 15 explanatory variables, such as the length and angle of the blades.

Practical Example 2 : car air conditioner design





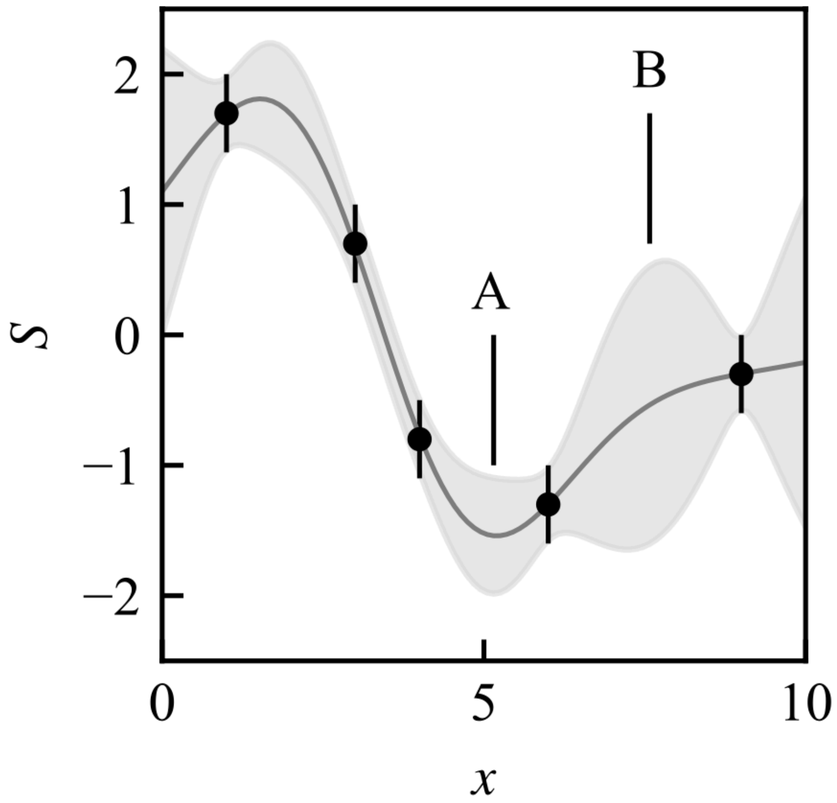
- Bayesian Optimization
  1. Simulate 45 different sets of design (= 15 explanatory variables) and measure the airflow performance and noise.
  2. Gaussian process regression across the 15-dimensional space of the explanatory variables.
  3. Based on EI (Expected Improvement), simulate the next 4 sets of designs with high airflow performance and low noise.
  4. Repeat this process 8 times.

$$\mathrm{EI}(\boldsymbol{x}) = \int_{-\infty}^{y_{\min}} (y_{\min} - y(\boldsymbol{x}))\mathcal{N}(y_{\min} - \mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x}))dy$$

Hands-on exercise #3

Gaussian process regression

Lecture_Day1_Uemura/03_GaussianProcess.ipynb



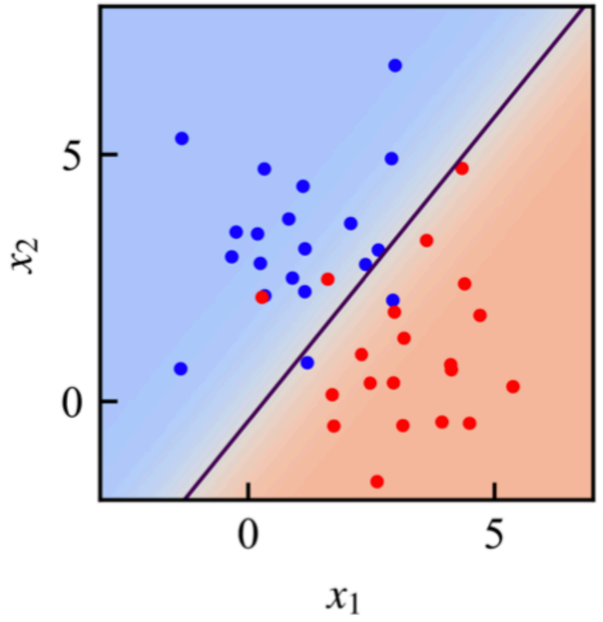
4. Classification models

4-1 Binary classification, evaluation metrics, confusion matrix
4-2 Models: Logistic regression & Support Vector Machine (SVM)

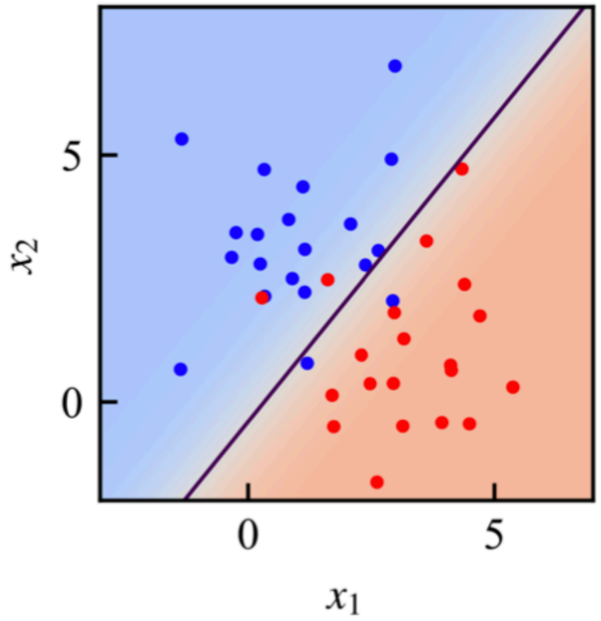
4-1 Binary classification, evaluation metrics, confusion matrix

Binary classification problems

- Binary classification: “Positive” or “Negative,” “Real” or “Fake,” or “Blue” or “Red”
  - While multi-class classification is an important topic, we won’t be covering it in this lecture due to time constraints.
- The target variable y has two possible classes, C_A and C_B,
  - In SVM (as discussed later), denoted as, y = +1 for C_A and y = -1 for C_B
- Determined by the explanatory variables **x** (, also known as **the feature vector**).
- Develop a classification model: $p(C_A|\mathbf{x})$, y=f(**x**)
  - Maximizing or minimizing an evaluation metric.
  - While in regression problems we often use an evaluation function like $\|\mathbf{y} - f(\mathbf{x})\|_2^2$, in classification, we use different metrics.



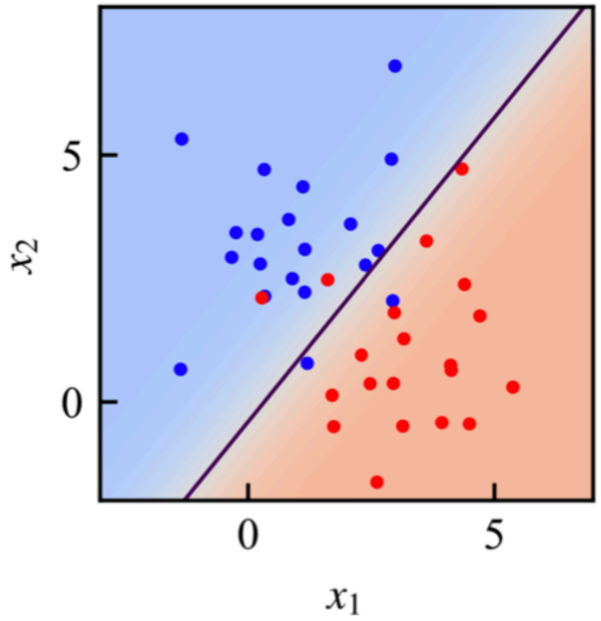
Error matrix



		True classes (label)	
		C_A	C_B
Estimates	C_A	18 (TP)	2 (FP)
	C_B	2 (FN)	18 (TN)

- Figure: An Example of a binary Classification
  - Two features
- Table: Error matrix (confusion matrix)
  - Of the 20 samples that are truly Class A:
    - 18 are correctly classified as Class A by the model (True Positive, TP)
    - 2 are incorrectly classified as Class B (False Negative, FN)
  - Of the 20 samples that are truly Class B:
    - 2 are incorrectly classified as Class A by the model (False Positive, FP)
    - 18 are correctly classified as Class B (True Negative, TN)
- Hereafter, when we refer to TP, we are referring to the number of TP samples (ex. TP=18).

Evaluation metrics based on the error matrix



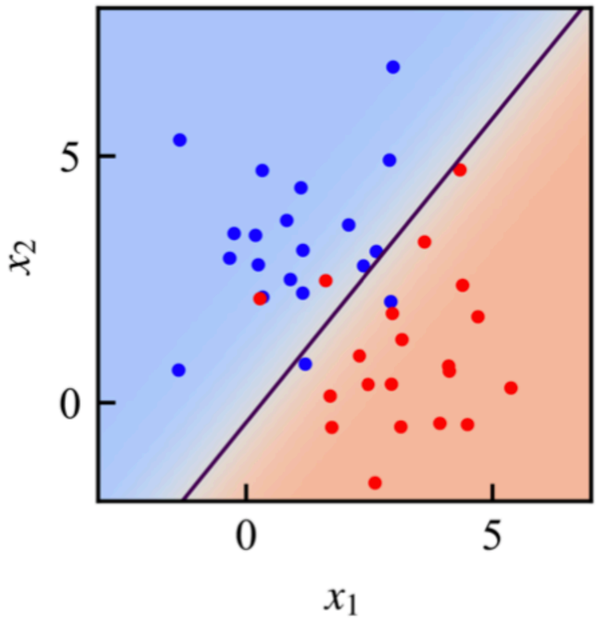
		True classes (label)	
		C_A	C_B
Estimates	C_A	18 (TP)	2 (FP)
	C_B	2 (FN)	18 (TN)

- Accuracy
  - The proportion of samples correctly classified out of all samples
  - (TP+TN)/(Total number of samples) : (18+18)/40 = 0.9
- Recall (True Positive Rate: TPR)
  - The proportion of samples correctly classified as Class A out of class A samples.
  - TP/(TP+FN) : 18/(18+2)=0.9
- False Positive Rate (FPR)
  - The proportion of samples incorrectly classified as Class A when they actually belong to Class B
  - FP/(FP+TN) : 2/(18+2)=0.1

- Accuracy and TPR : Higher values indicate better models
- FPR : Lower values indicate better models

Problems with Unequal Sample Sizes across Classes

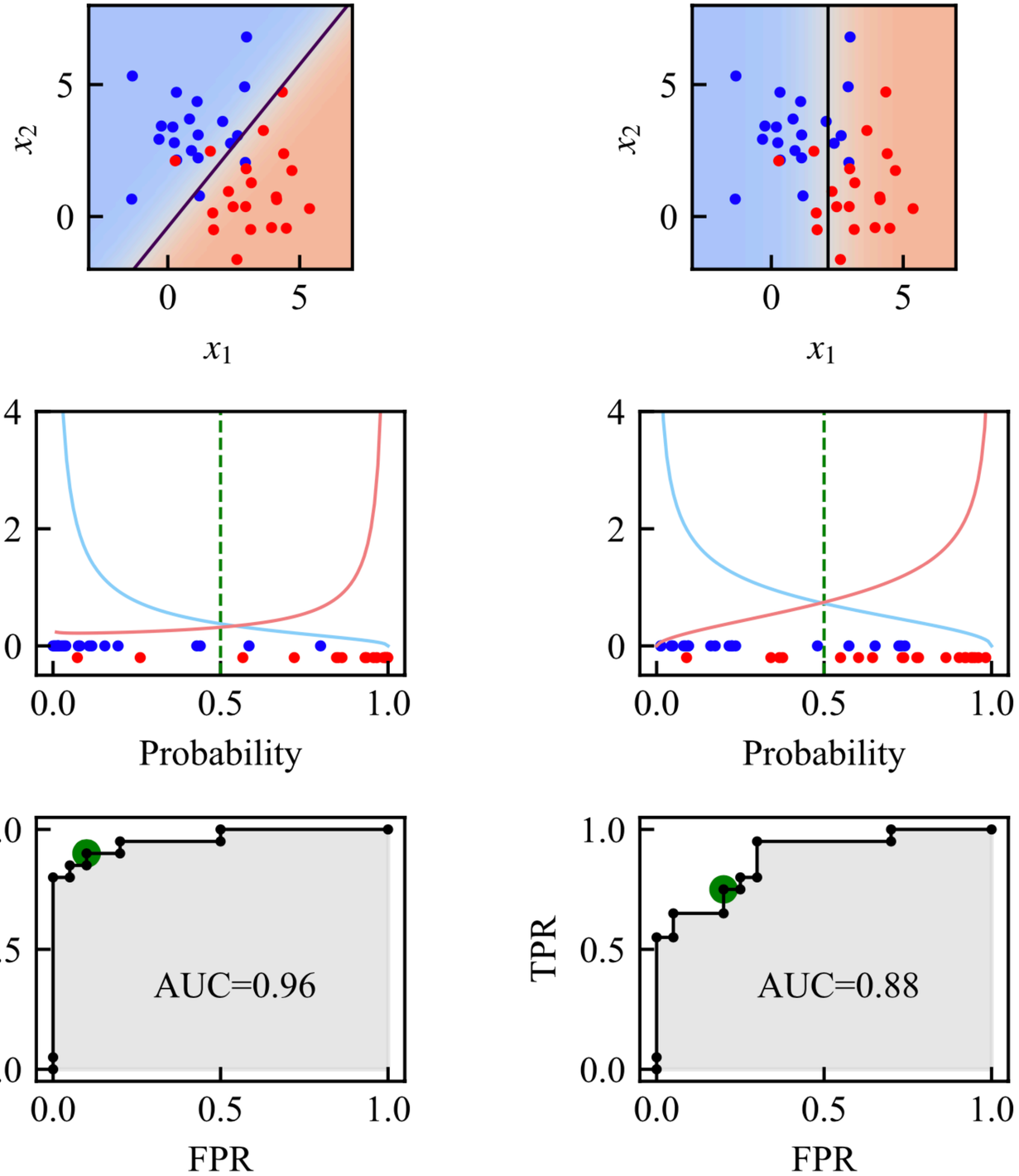
- Ideally, we would have 20 samples each for class A and class B.
- What if we have 10,000 samples for class A and only 10 for class B?
- A model that classifies all samples as class A:
  - Accuracy: 10000/10010 ~ 0.999 & TPR = 1 → Looks like a good model
  - FPR = 10/10 = 1 → a significant problem
- When there is a significant imbalance in class sizes, it's crucial to consider both TPR and FPR.



		True classes (label)	
		C_A	C_B
Estimates	C_A	18 (TP)	2 (FP)
	C_B	2 (FN)	18 (TN)

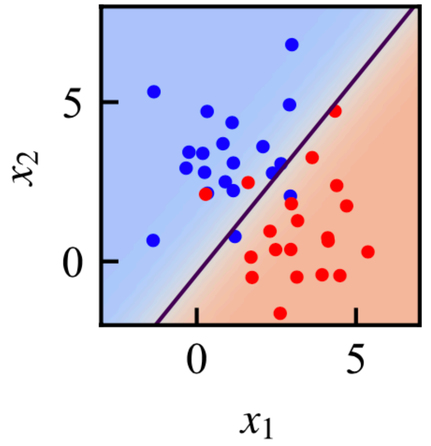
Receiver Operating Characteristic (ROC) Curve

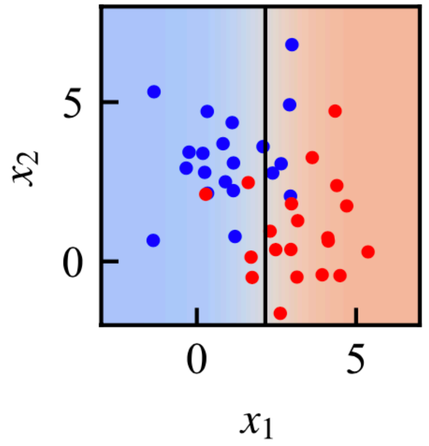
- Example on the right:
  - With p=0.5 as the decision boundary: TPR=18/20=0.9, FPR=2/20=0.1
  - On the bottom left plot: This point is represented by a green dot, with FPR on the x-axis and TPR on the y-axis.
  - With p=0.2 as the decision boundary: TPR=19/20=0.95, FPR=4/20=0.2
  - On the bottom left plot: The green dot moves to a point further up and to the right.
  - By varying the threshold probability used for decision-making, you can plot a curve on the TPR-FPR plane → the ROC curve.

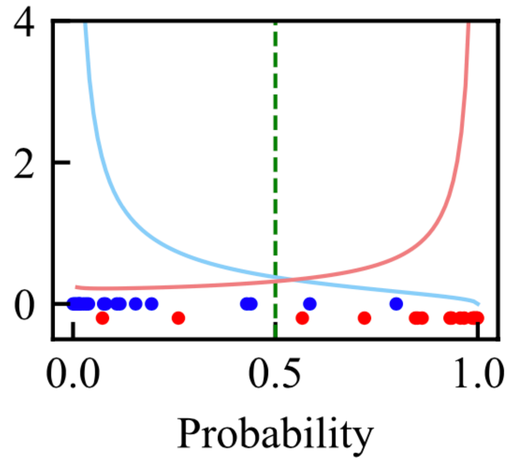


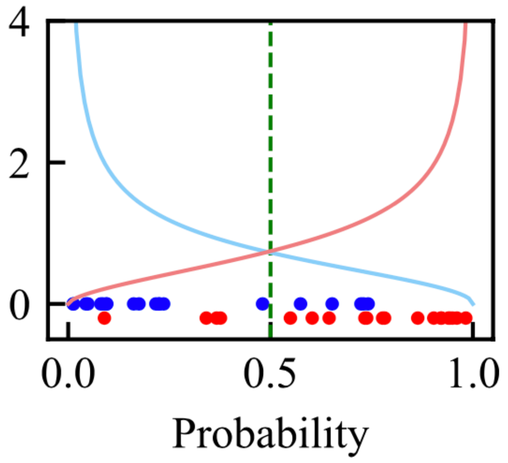
Evaluating ROC Curves and AUC

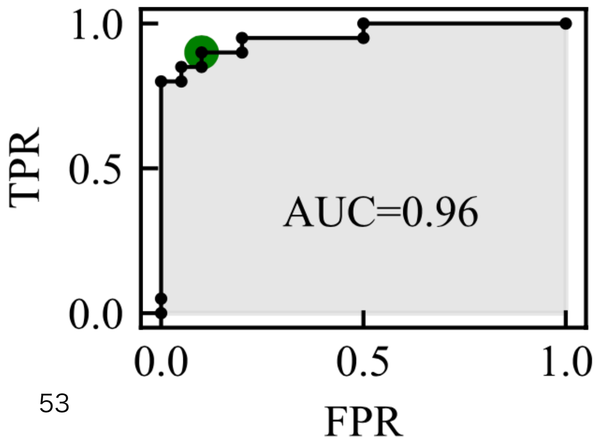
- An ideal classification model has FPR=0 & TPR=1
- The proximity of the ROC curve to the point (0,1) indicates the model's performance.
- Example in the right panels:
  - The model using only x_1 results in an ROC curve that is farther from (0,1) → Indicates a lower-performance model.
- The Area Under the ROC Curve (AUC) is a commonly used performance metric:
  - Larger the AUC, Better the model.
  - Left panel: AUC = 0.96, Right panel: AUC = 0.88 → The model in the left figure performs better.

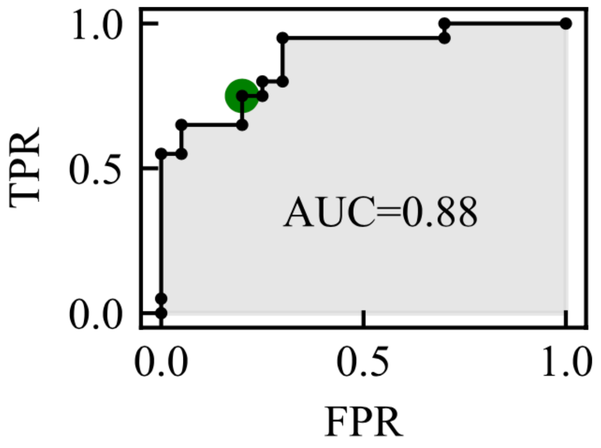








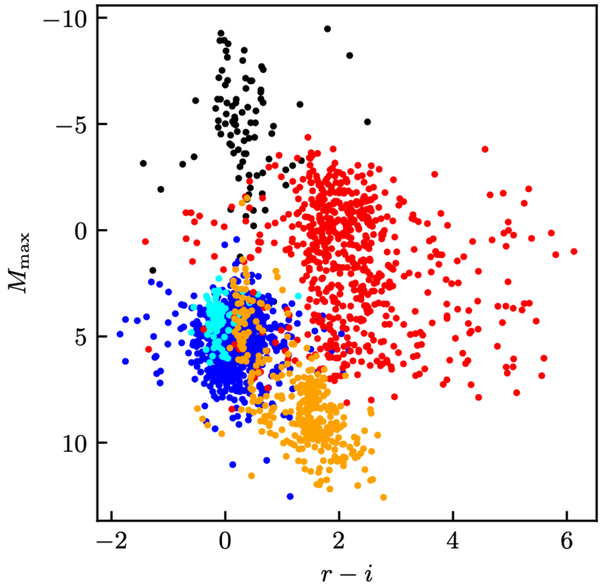




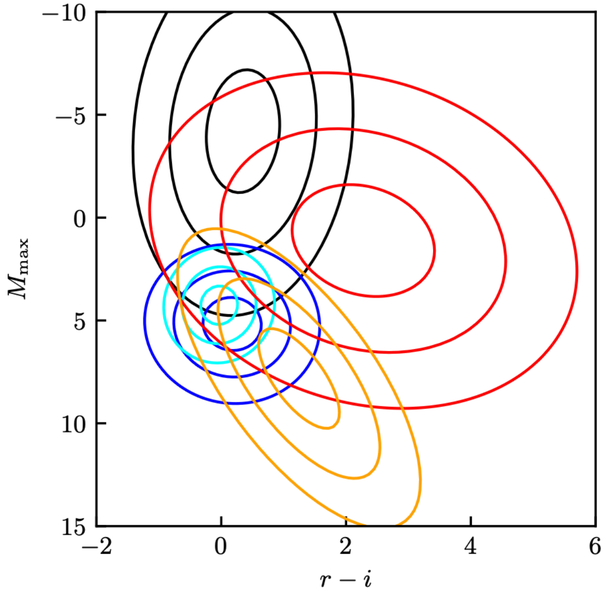
4-2 Models: Logistic regression & Support Vector Machine (SVM)

Probabilistic Classification Models

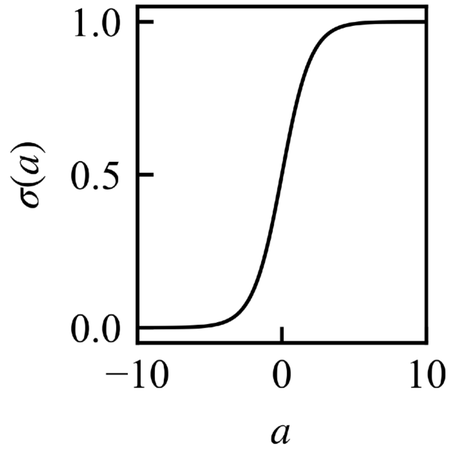
- The probability that a sample with feature vector **x** belongs to $y = C_A$ is given by $p(y = C_A \,|\, x) = p(C_A \,|\, x)$.
- Using Bayes' theorem: $$p(C_A|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_A)p(C_A)}{\sum p(\boldsymbol{x}|y)p(y)} = \frac{p(\boldsymbol{x}|C_A)p(C_A)}{p(\boldsymbol{x}|C_A)p(C_A) + p(\boldsymbol{x}|C_B)p(C_B)}$$



- Bayesian classifier: A model that computes the posterior probability by providing all elements on the right side of Bayes' theorem.
  - $p(x \,|\, C_A)$: The distribution of **x** for samples belonging to class C_A (e.g., approximated by a Gaussian distribution).
  - $p(C_A)$: The prior probability that a sample belongs to class C_A (e.g., the proportion of C_A samples in the training data).
  - Requires adjusting many parameters (e.g., the covariance matrix in the case of a multivariate normal distribution).



Logistic regression



- σ (a) ：Logistic sigmoid function
- a is a function of the feature vector **x** (linear combination)
- Estimate **w** (weights) from the data using maximum likelihood estimation = Logistic Regression.
- Target variable: Expressed as y = 1 for class A and y = 0 for class B.
- Negative log-likelihood : E(**w**)
  - Takes the form of cross-entropy between **y** and $\boldsymbol{\sigma}$.

$$p(C_A|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_A)p(C_A)}{p(\boldsymbol{x}|C_A)p(C_A) + p(\boldsymbol{x}|C_B)p(C_B)}$$

$$= \frac{1}{1+\exp(-a)} = \sigma(a)$$

$$a = \log \frac{p(\boldsymbol{x}|C_A)p(C_A)}{p(\boldsymbol{x}|C_B)p(C_B)}$$

$$a = \boldsymbol{w}_1^T \boldsymbol{x} + w_0 = \boldsymbol{w}^T \boldsymbol{x}$$

$$p(y_i|\boldsymbol{w}) = \sigma(\boldsymbol{w}^T \boldsymbol{x}_i)$$

$$p(\boldsymbol{y}|\boldsymbol{w}) = \prod_i \sigma(\boldsymbol{w}^T \boldsymbol{x}_i)^{y_i} \{1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i)\}^{1-y_i}$$

$$E(\boldsymbol{w}) = -\log p(\boldsymbol{y}|\boldsymbol{w}) = -\sum_i \{y_i \log \sigma(\boldsymbol{w}^T \boldsymbol{x}_i) + (1-y_i) \log(1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i))\}$$

Deterministic Classification Models

- Probabilistic models = based on Bayes' theorem.
- Deterministic models construct a discriminant function $f(x;\theta)$ and a decision boundary $f(x;\theta)=c$.
- There are various models, but here we introduce the representative Support Vector Machine (SVM).

Hard-Margin SVM (Perfectly Separable Problems)

- Linear discriminant function for features :

  $f(x) = w^T x + b$

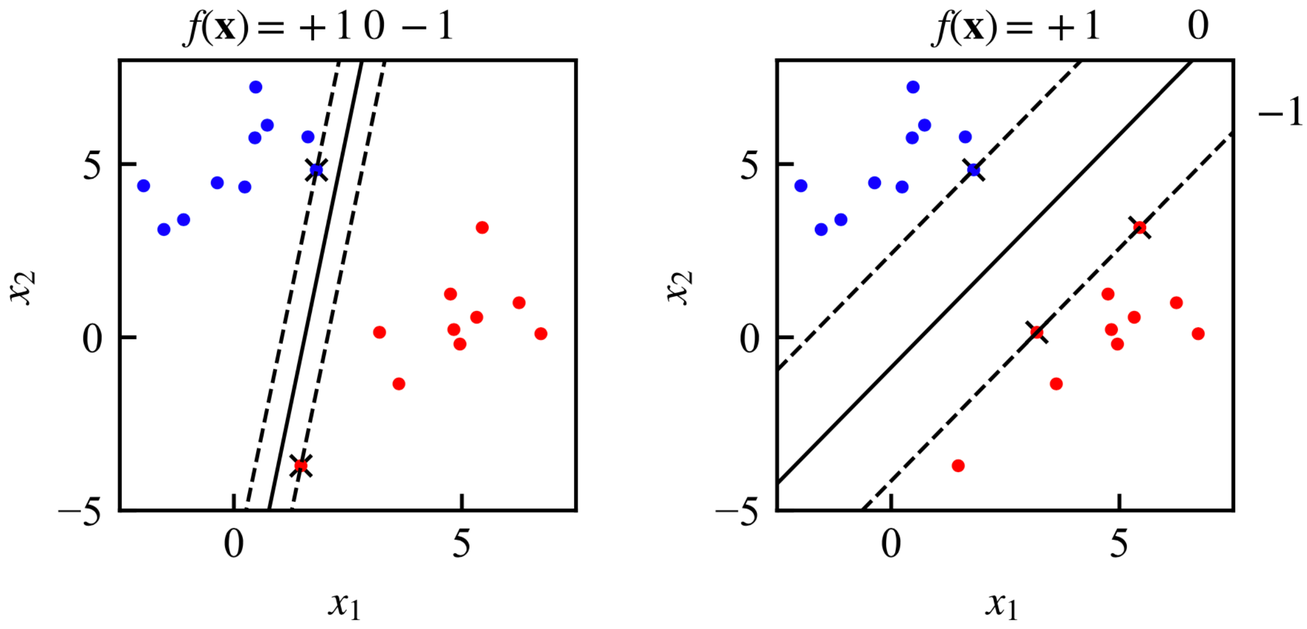
  - Later, we'll extend this to nonlinear discriminant functions.

- Decision Boundary: $f(x) = 0$

- Target Variable: y=1 if class A, y=-1 if class B

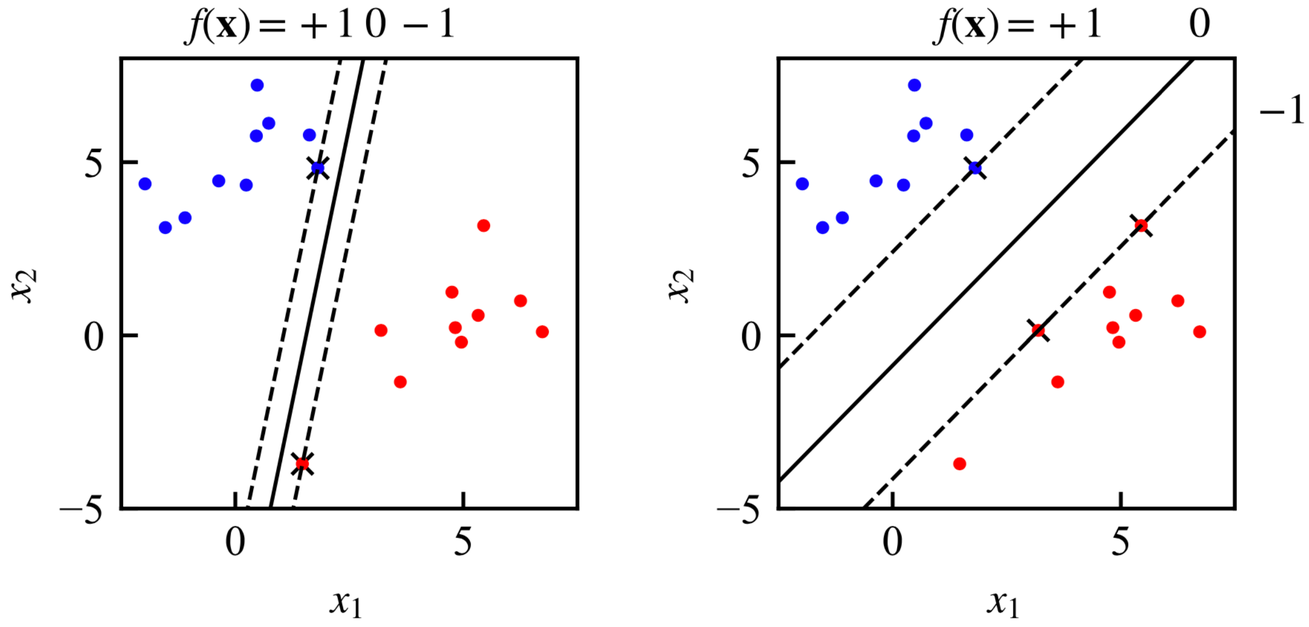
- The model classifies a sample as class A if $f(x) > 0$, class B if $f(x) < 0$.

- As illustrated in the right figure, there are numerous possible discriminant functions.



SVM Chooses the Model with the Maximum Margin

- Margin: The distance between the decision boundary and the closest samples x_{SV}.
  - , called the support vectors.
- In general, the distance between x' and the hyperplane $f(x) = w^T x = 0$ is $|f(\boldsymbol{x'})|/\|\boldsymbol{w}\|_2$
- If we construct the decision function such that $f(\boldsymbol{x}_{\text{SV}}) = \pm 1$, the margin becomes $1/\|\boldsymbol{w}\|_2$
- For Class A (y=1), f(x) > 1 , and for Class B (y=-1), f(x) < -1 . Therefore, the SVM problem reduces to the quadratic programming problem, as shown on the right.



$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$\text{subject to} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

Wanting a Non-Linear Boundary

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$\text{subject to} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

- Apply a non-linear mapping ϕ(x) ?
- Transform the quadratic programming problem into a single objective function using the Lagrange multipliers method.
  - Introduce new variables, α
- Minimize the objective function L by finding the conditions where the derivative of L is zero.
- Substitute the conditions for w and b back into the objective function.
- This transforms into a problem of maximizing L with respect to α.
- The inner product of feature vectors x_i and x_j appears → Kernel trick.

$$L = \frac{1}{2}\|\boldsymbol{w}\|_2 - \sum_i \alpha_i \left\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1\right\}$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{0} \Rightarrow \boldsymbol{w} - \sum_i \alpha_i y_i \boldsymbol{x}_i = \boldsymbol{0}$$

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow -\sum_i \alpha_i y_i = 0$$

$$\sum_i \alpha_i y_i = 0$$

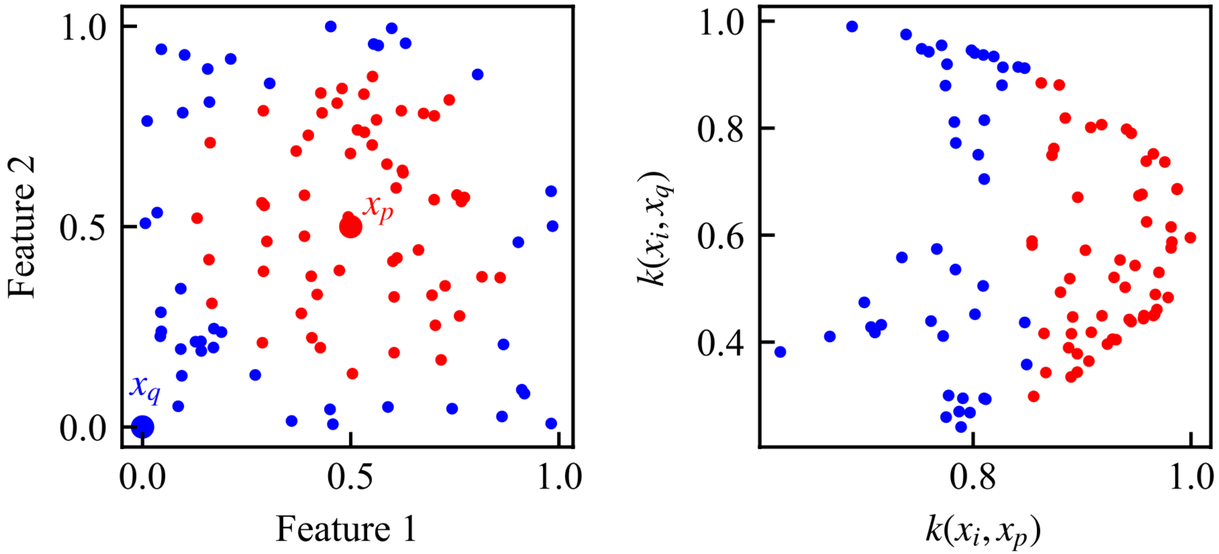
$$L = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

The Kernel Trick

- The Kernel Trick: Replace the inner product of feature vectors with a kernel function which represent a measure of similarity between features.
- For example:
  - the RBF (Radial Basis Function) kernel, or Gaussian kernel.
  - When x_i and x_j are close in feature space, k=1; when they are far apart, k=0.
- Kernel functions typically have adjustable parameters (e.g., σ for the RBF kernel).

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2}\right)$$

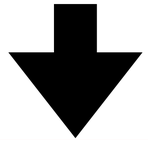


Soft Margin SVM

- In real-world problems, it's rare to have perfectly separable data where hard margin SVM is applicable. Measurement errors and other influences often lead to inevitable misclassification → Soft Margin SVM.
- The right formula:
  - Hard Margin: For $y = 1$, $f(x)$ $(=wx+b) > 1$ must hold.
  - Soft Margin: Relaxing the constraints → introducing new variables ξ (>0), allowing $f(x) < 1$ even for $y = 1$
- If many ξ are non-zero, the performance of the classifier may degrade → add constraints on ξ.
- Adjust the regularization parameter C (using cross-validation, etc.) to build a model with high generalization performance.

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$\text{subject to} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

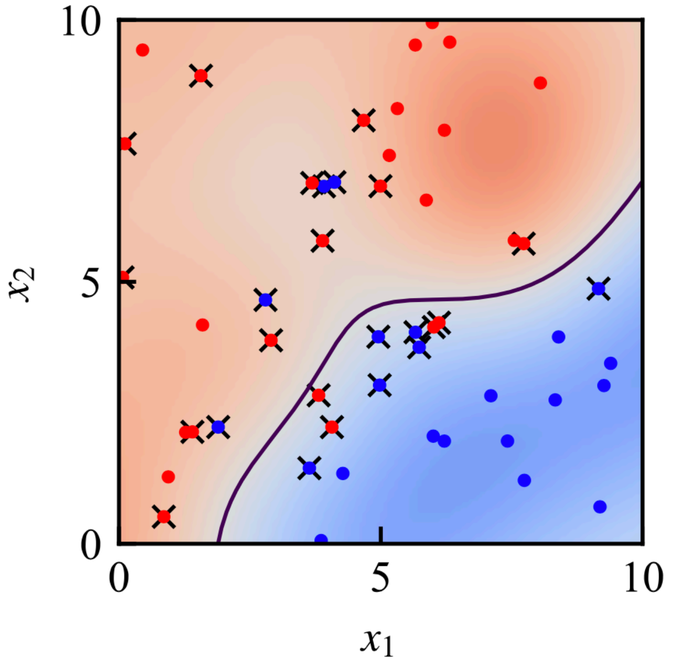


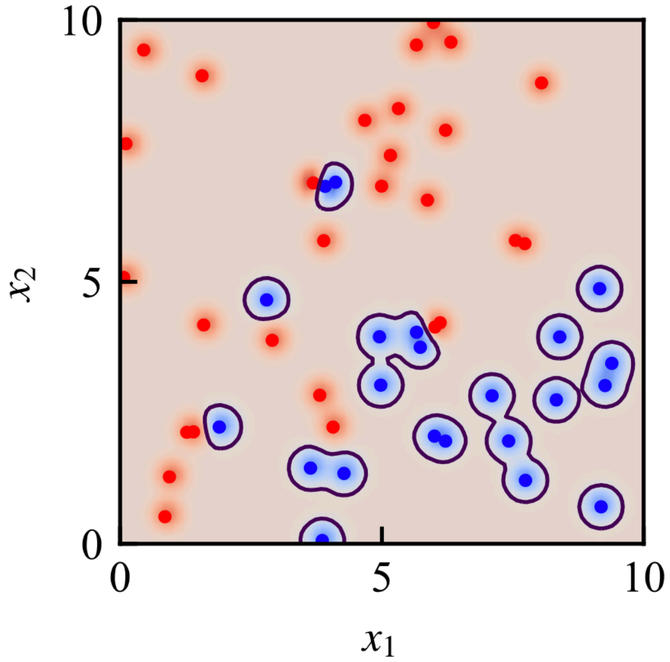
$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i$$

$$\text{subject to} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i$$

Classifying with Synthetic Data

- Use RBF kernel
- Determine regularization parameter C and kernel parameter σ by evaluating with ROC-AUC through cross-validation
- Left Figure: Best model
- Right Figure: Overfitted model
  - Kernel variance is small, and C is large.





Hands-on exercise #4

Support Vector Machine

Lecture_Day1_Uemura/04_SVM.ipynb