FE540 금융공학 인공지능 및 기계학습

Regression

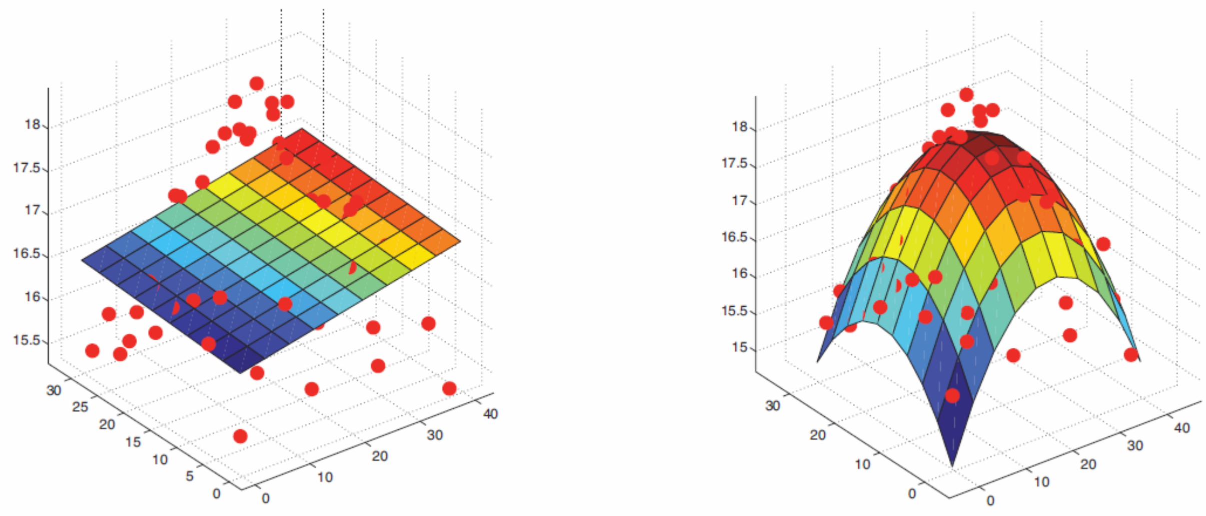
Kee-Eung Kim
Department of Computer Science
KAIST

Korea Advanced Institute of Science and Technology
한국과학기술원

Regression

- Would like to write numeric output as a function of input
  - Output: dependent variable
  - Input: independent variable
- Assume: $y = \mathbf{w}^\top \mathbf{x} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$
  - likelihood: $p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^\top \mathbf{x}, \sigma^2)$
  - can handle non-linear relationships via basis function expansion:

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}), \sigma^2) \quad \text{e.g. } \boldsymbol{\phi}(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2]$$



(a) (b)

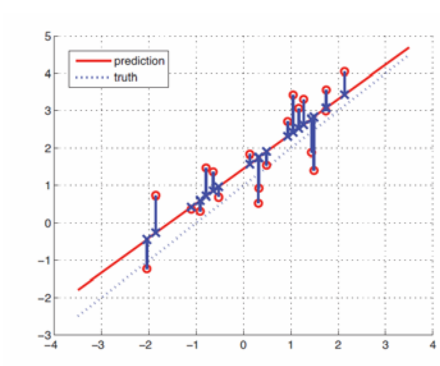
Korea Advanced Institute of Science and Technology
한국과학기술원

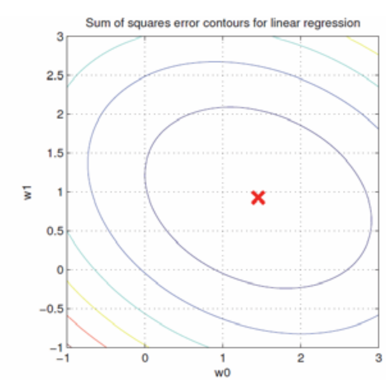
Maximum Likelihood Estimation

- MLE is equivalent to least squares
  - $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} \equiv \mathrm{argmax}_{\boldsymbol{\theta}} \log p(\mathcal{D}|\boldsymbol{\theta}) = \mathrm{argmax}_{\boldsymbol{\theta}}\, \ell(\boldsymbol{\theta})$
    where log-likelihood $\ell(\boldsymbol{\theta}) \equiv \log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$
  - maximizing log-likelihood = minimizing negative log-likelihood

$$\begin{aligned}\mathrm{NLL}(\boldsymbol{\theta}) &\equiv -\sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) \\ &= -\sum_{i=1}^{N} \log \left[\left(\frac{1}{2\pi\sigma^2}\right)^{1/2} \exp\left(-\frac{1}{2\sigma^2}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2\right) \right] \\ &= \frac{1}{2\sigma^2}\mathrm{RSS}(\mathbf{w}) + \frac{N}{2}\log(2\pi\sigma^2)\end{aligned}$$

  - Residual sum of errors (sum of squared errors)
    $\mathrm{RSS}(\mathbf{w}) \equiv \sum_i (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$





Korea Advanced Institute of Science and Technology
한국과학기술원

Derivation of MLE

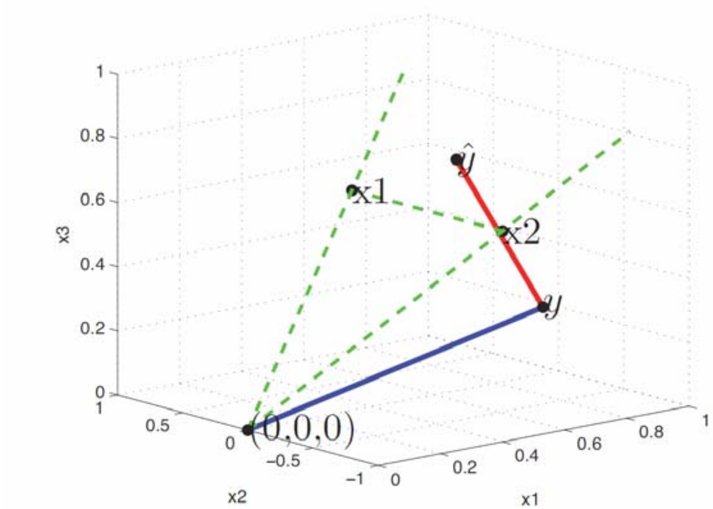
- Obtaining the least squares solution
  - $\mathrm{RSS}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$
  $$= \mathbf{w}^\top(\mathbf{X}^\top\mathbf{X})\mathbf{w} - 2\mathbf{w}^\top(\mathbf{X}^\top\mathbf{y}) + \mathbf{y}^\top\mathbf{y}$$
  - where
  $$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} \qquad \mathbf{X}^\top\mathbf{y} = \sum_{i=1}^{N} \mathbf{x}_i y_i$$
  $$\mathbf{X}^\top\mathbf{X} = \sum_{i=1}^{N} \mathbf{x}_i^\top \mathbf{x}_i = \sum_{i=1}^{N} \begin{pmatrix} x_{i,1}^2 & \cdots & x_{i,1}x_{i,D} \\ & \ddots & \\ x_{i,D}x_{i,1} & \cdots & x_{i,D}^2 \end{pmatrix}$$
  - gradient: $\mathbf{g}(\mathbf{w}) = \mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{y}$
  - extreme point: $\mathbf{X}^\top\mathbf{X}\mathbf{w} = \mathbf{X}^\top\mathbf{y}$
  $$\hat{\mathbf{w}}_{\mathrm{OLS}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

Geometric Interpretation

- Orthogonal projection
  - column vectors from the data $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} = (\tilde{\mathbf{x}}_1 \cdots \tilde{\mathbf{x}}_D)$
  - target value vector $\mathbf{y} \in \mathbb{R}^N$
  - Linear regression = find vector $\hat{\mathbf{y}} = \mathrm{argmin}_{\hat{\mathbf{y}}} \|\mathbf{y} - \hat{\mathbf{y}}\|_2$ such that $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} = w_1\tilde{\mathbf{x}}_1 + \cdots + w_D\tilde{\mathbf{x}}_D$

$$\Leftrightarrow \hat{\mathbf{y}} \in \mathrm{span}(\mathbf{X})$$

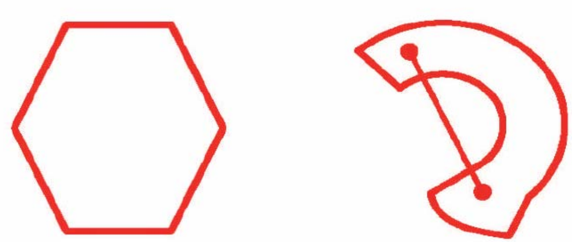
$$\Leftrightarrow \hat{\mathbf{y}} \in \mathrm{span}(\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_D\})$$



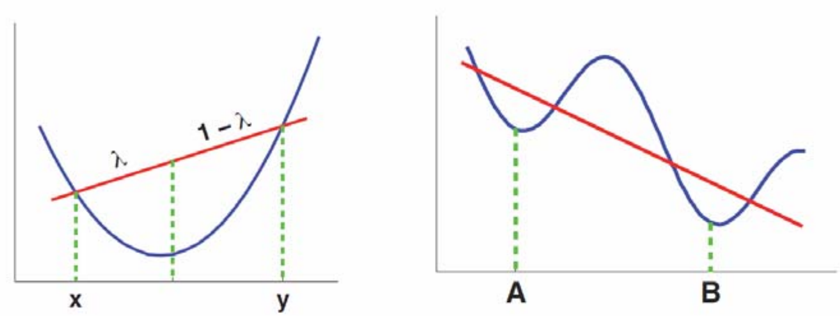
- residual vector should be orthogonal so that the norm is minimized:
  $\tilde{\mathbf{x}}_j^\top(\mathbf{y} - \hat{\mathbf{y}}) = 0 \Rightarrow \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$
  $\Rightarrow \mathbf{w} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$
- Also, $\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$

Convexity

- negative log-likelihood of Gaussian:
  - $$\mathrm{NLL}(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{1/2} \exp\left(-\frac{1}{2\sigma^2}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2\right)\right]$$
    $$= \frac{1}{2\sigma^2}\mathrm{RSS}(\mathbf{w}) + \frac{N}{2}\log(2\pi\sigma^2)$$
  - $\mathrm{RSS}(\mathbf{w}) \equiv \sum_i (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$
- Convex set $\mathcal{S} : \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{S}, \forall \lambda \in [0,1],\ \lambda\boldsymbol{\theta} + (1-\lambda)\boldsymbol{\theta}' \in \mathcal{S}$
- Convex function $f(\boldsymbol{\theta})$
  - $\boldsymbol{\theta} \in \mathcal{S}$ (defined on a convex set)
  - $f(\lambda\boldsymbol{\theta} + (1-\lambda)\boldsymbol{\theta}') \leq \lambda f(\boldsymbol{\theta}) + (1-\lambda) f(\boldsymbol{\theta}')$
- convex functions are ideal for optimization



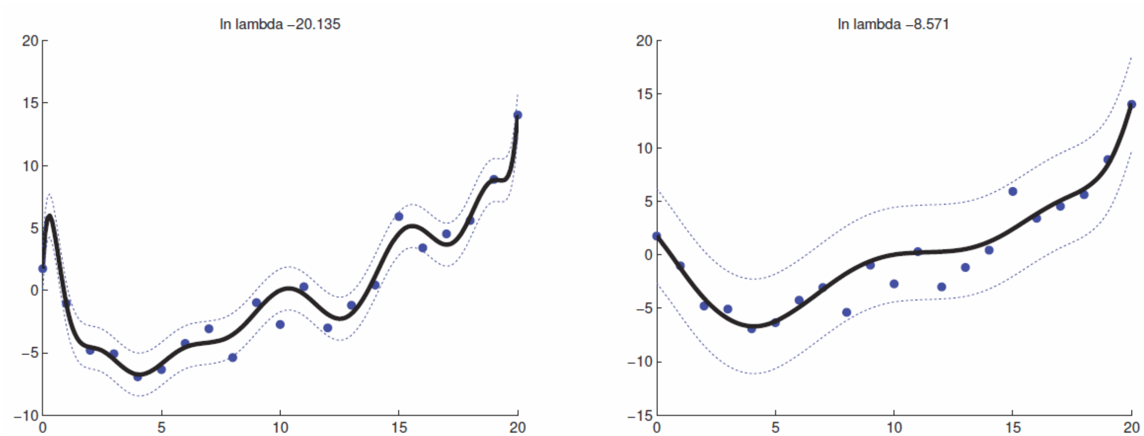
(a) (b)



(a) (b)

Ridge Regression

- Overfitting is a common problem in higher-order regression
  - Smoother curve = Smaller parameters
    $p(\mathbf{w}) = \prod_j \mathcal{N}(w_j|0, \tau^2)$
  - MAP estimation
    $\operatorname{argmax}_{\mathbf{w}} \sum_{i=1}^{N} \log \mathcal{N}(y_i|\mathbf{w}^\top \mathbf{x}_i, \sigma^2) + \sum_{j=1}^{D} \log \mathcal{N}(w_j|0, \tau^2)$
    $= \operatorname{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2$
  - Gaussian prior is equivalent to ℓ_2 regularization (i.e. weight decay)
- Ridge regression: $\hat{\mathbf{w}}_{\text{ridge}} = (\lambda \mathbf{I}_D + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$



KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Bayesian Linear Regression

- ☐ want: full posterior over $\mathbf{w}$ and σ^2
  - assume σ^2 is known and focus on posterior over $\mathbf{w}$ only
  - likelihood: $p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I}_N)$
    $\propto \exp(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}))$
  - use conjugate prior: $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_0, \mathbf{V}_0)$
  - compute posterior:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \sigma^2) \propto \mathcal{N}(\mathbf{w}|\mathbf{w}_0, \mathbf{V}_0)\mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_N, \mathbf{V}_N)$$

$$\mathbf{w}_N = \mathbf{V}_N \mathbf{V}_0^{-1}\mathbf{w}_0 + \frac{1}{\sigma^2}\mathbf{V}_N \mathbf{X}^\top \mathbf{y}$$

$$\mathbf{V}_N^{-1} = \mathbf{V}_0^{-1} + \frac{1}{\sigma^2}\mathbf{X}^\top \mathbf{X}$$

Bayesian Linear Regression

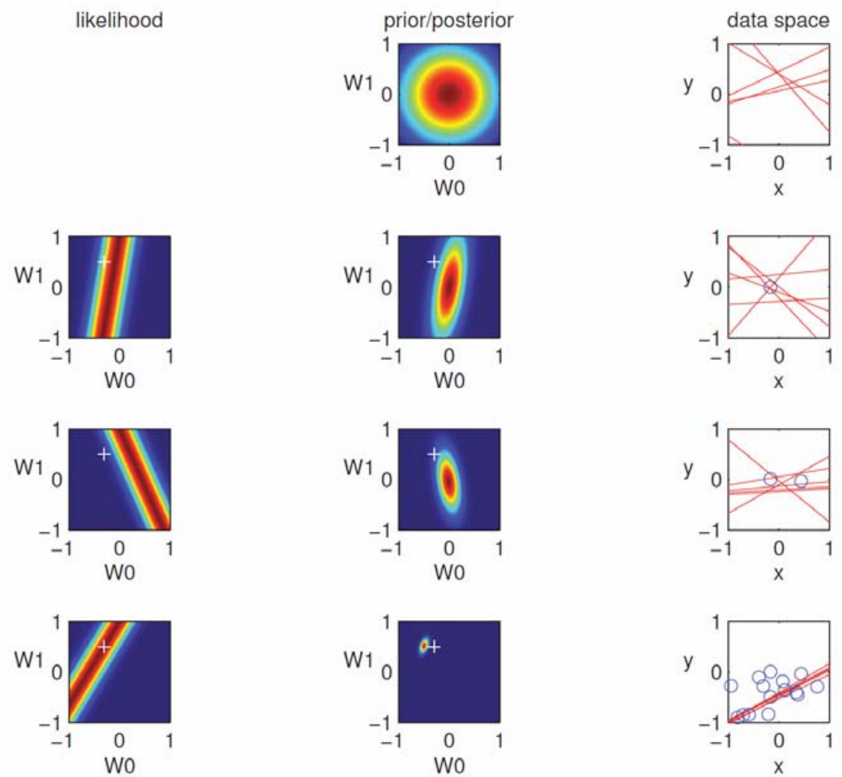
- posterior:

$$p(\mathbf{w}|\mathbf{X},\mathbf{y},\sigma^2) \propto \mathcal{N}(\mathbf{w}|\mathbf{w}_0,\mathbf{V}_0)\mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w},\sigma^2\mathbf{I}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_N,\mathbf{V}_N)$$

$$\mathbf{w}_N = \mathbf{V}_N\mathbf{V}_0^{-1}\mathbf{w}_0 + \frac{1}{\sigma^2}\mathbf{V}_N\mathbf{X}^\top\mathbf{y}$$

$$\mathbf{V}_N^{-1} = \mathbf{V}_0^{-1} + \frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}$$

  - Ridge regression is a special case: $\mathbf{w}_0 = \mathbf{0}$ and $\mathbf{V}_0 = \tau^2\mathbf{I}$



Bayesian Linear Regression

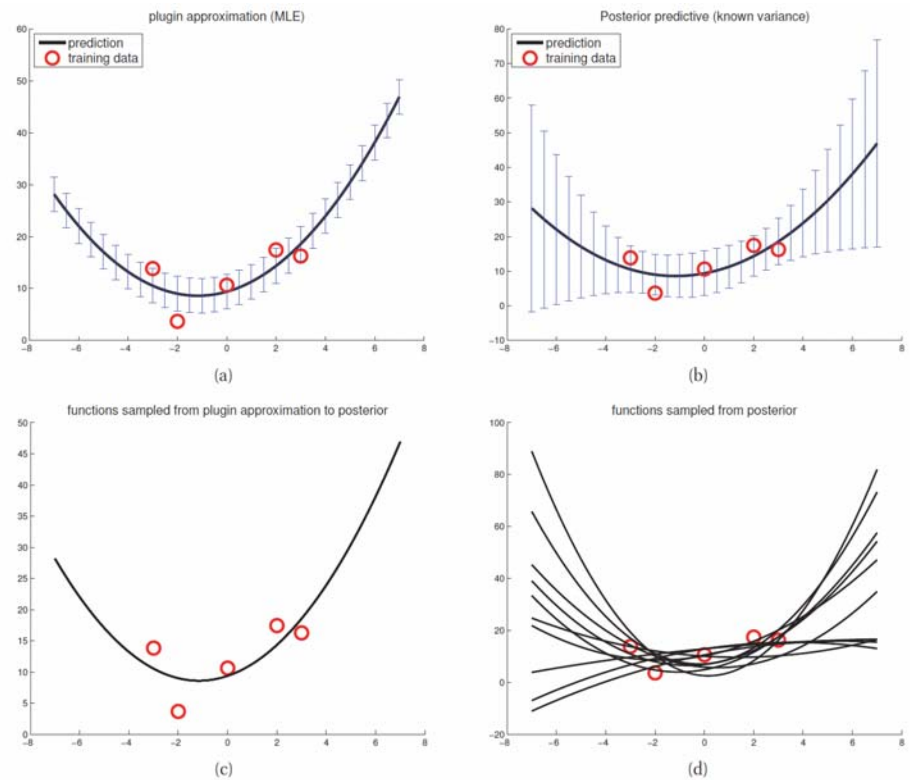
- Posterior predictive distribution:

$$
\begin{aligned}
p(y|\mathbf{x}, \mathcal{D}, \sigma^2) &= \int \mathcal{N}(y|\mathbf{x}^\top \mathbf{w}, \sigma^2)\mathcal{N}(\mathbf{w}|\mathbf{w}_N, \mathbf{V}_N)d\mathbf{w} \\
&= \mathcal{N}(y|\mathbf{w}_N^\top \mathbf{x}, \sigma_N^2(\mathbf{x}))
\end{aligned}
$$

$$
\sigma_N^2(\mathbf{x}) = \sigma^2 + \mathbf{x}^\top \mathbf{V}_N \mathbf{x}
$$

- Plug-in approximation (constant variance)

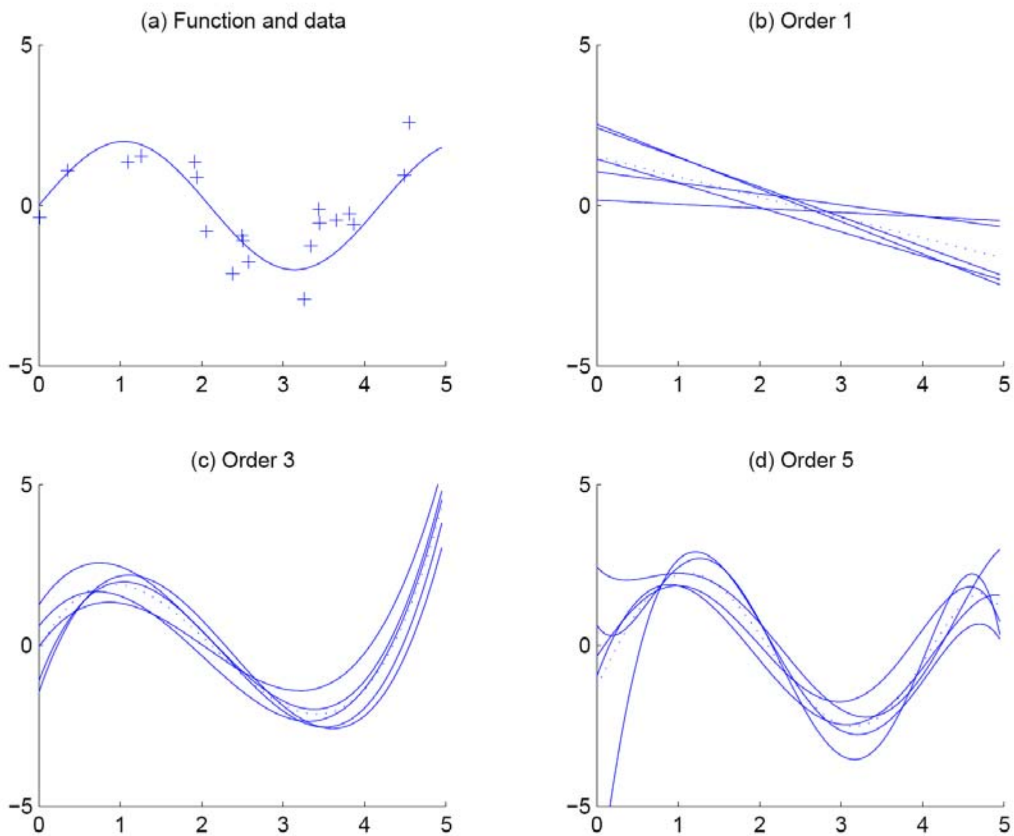
$$
\begin{aligned}
p(y|\mathbf{x}, \mathcal{D}, \sigma^2) & \\
&\approx \int \mathcal{N}(y|\mathbf{x}^\top \mathbf{w}, \sigma^2)\delta_{\hat{\mathbf{w}}}(\mathbf{w})d\mathbf{w} \\
&= \mathcal{N}(y|\mathbf{x}^\top \hat{\mathbf{w}}, \sigma^2)
\end{aligned}
$$



- Full PPD essential for active learning

Model Selection

- Linear regression extends to non-linear regression by basis expansion:
  - $p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}), \sigma^2) \quad \text{e.g. } \boldsymbol{\phi}(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2]$



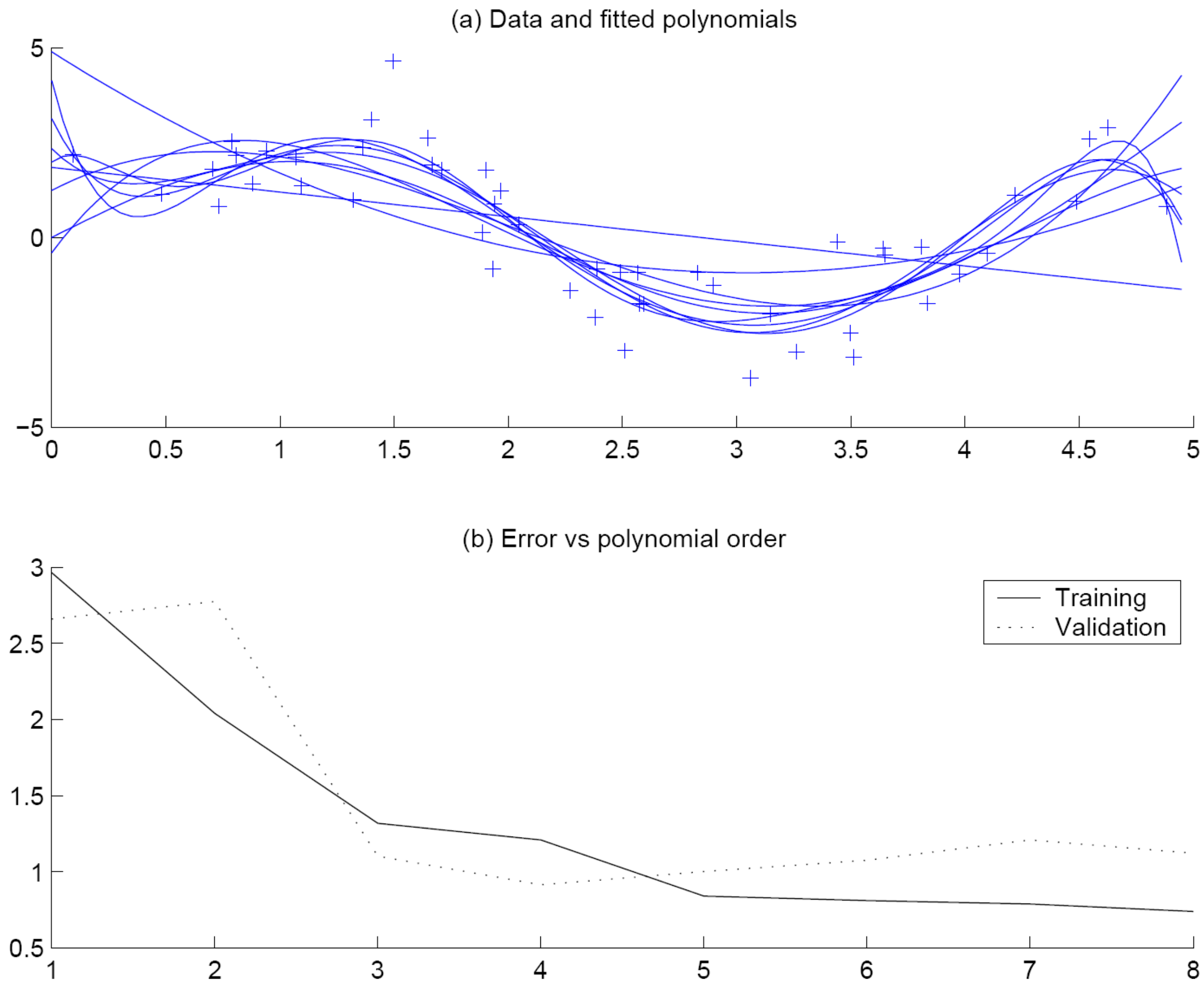
Korea Advanced Institute of Science and Technology
한국과학기술원

Model Selection Techniques

- Cross-validation
  - Measure generalization accuracy by testing on data unused during training
- Regularization
  - Penalize complex models
  - E' = error on data + λ model complexity
  - Akaike's information criterion (AIC), Bayesian information criterion (BIC), Mimimum description length (MDL)
- Structural risk minimization (SRM)
  - Foundation of support vector machines
- Bayesian model selection
  - Suppose we have prior on models, p(model)
  - P(model|data) = p(data|model) p(model) / p(data)
  - If prior favors simpler models, it is a regularization

Model Selection Procedures

- Cross-validation

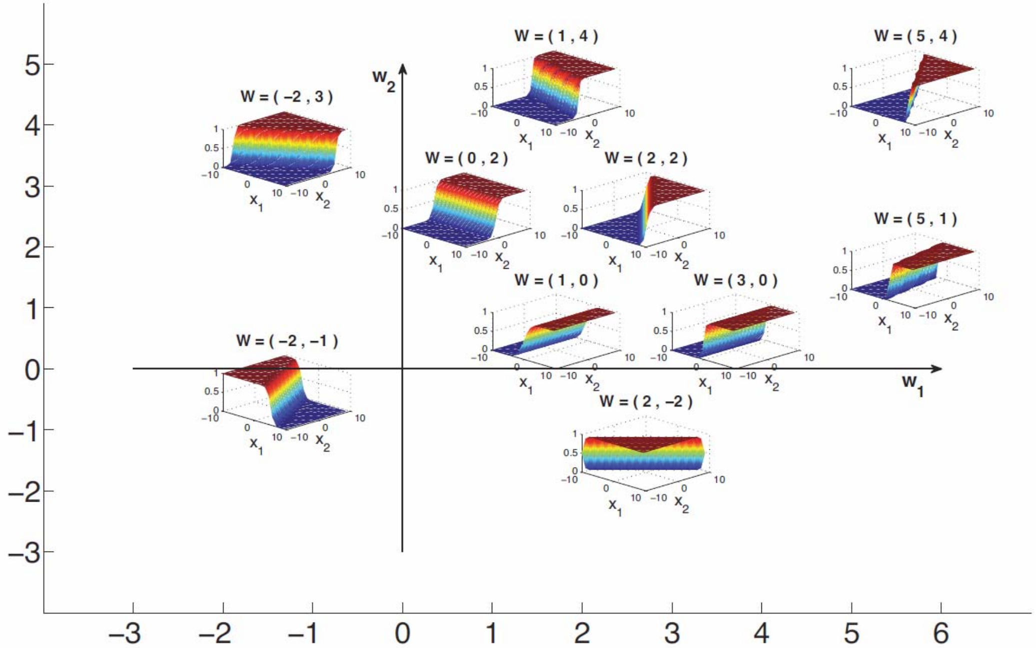


Korea Advanced Institute of Science and Technology
한국과학기술원

Logistic Regression (Classification)

- Generative vs. Discriminative approach to $p(y|\mathbf{x})$
  - Generative: estimate likelihood $p(\mathbf{x}|y)$ and use Bayes rule
  - Discriminative: fit $p(y|\mathbf{x})$ directly from data

- Logistic regression for binary classification
  - $p(y|\mathbf{x}, \mathbf{w}) = \mathrm{Ber}(y|\mathrm{sigm}(\mathbf{w}^\top \mathbf{x}))$ and estimate $\mathbf{w}$ from data



Maximum Likelihood Estimation

- Minimize negative log-likelihood (NLL)
  - $$\begin{aligned}\mathrm{NLL}(\mathbf{w}) &= -\sum_{i=1}^{N} \log[\mu_i^{\mathbb{I}(y_i=1)} \times (1-\mu_i)^{\mathbb{I}(y_i=0)}] \\ &= -\sum_i [y_i \log \mu_i + (1-y_i)\log(1-\mu_i)]\end{aligned}$$

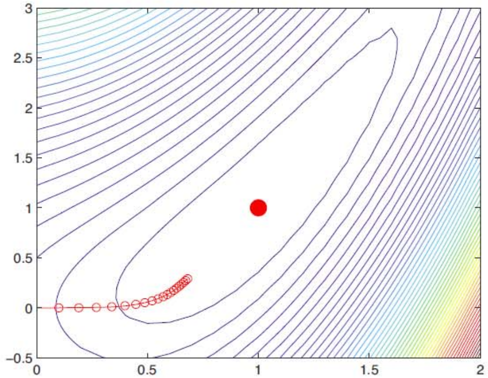
  where $\mu_i = \mathrm{sigm}(\mathbf{w}^\top \mathbf{x}_i)$

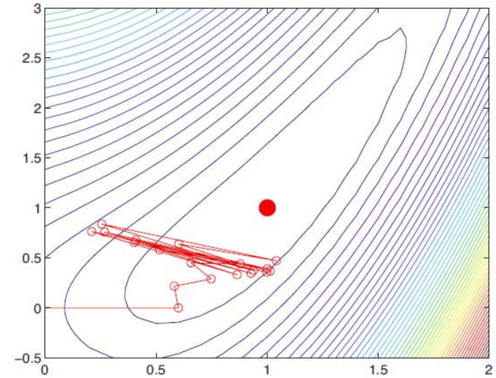
- Use an optimization algorithm to minimize NLL
  - $\mathbf{g} = \frac{d}{d\mathbf{w}}\mathrm{NLL}(\mathbf{w}) = \sum_i (\mu_i - y_i)\mathbf{x}_i = \mathbf{X}^\top(\boldsymbol{\mu} - \mathbf{y})$
  - $\mathbf{H} = \frac{d}{d\mathbf{w}}\mathbf{g}(\mathbf{w})^\top = \sum_i (\nabla_{\mathbf{w}}\mu_i)\mathbf{x}_i^\top = \sum_i \mu_i(1-\mu_i)\mathbf{x}_i\mathbf{x}_i^\top = \mathbf{X}^\top\mathbf{S}\mathbf{X}$

  where $\mathbf{S} \equiv \mathrm{diag}(\mu_i(1-\mu_i))$

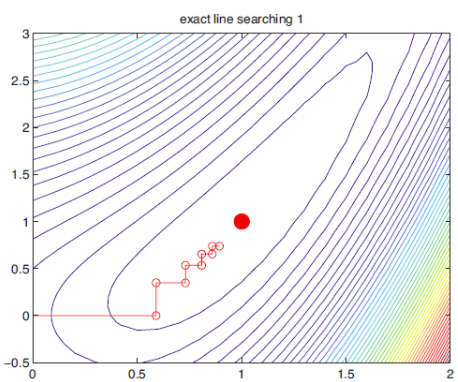
Steepest Descent

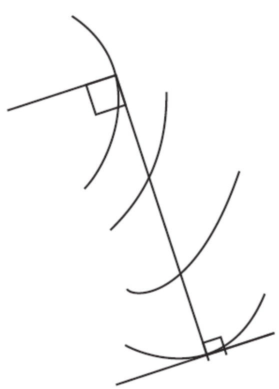
- Also known as gradient descent: $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \mathbf{g}_k$





- Line search
  - from Taylor expansion: $f(\boldsymbol{\theta} + \eta \mathbf{d}) \approx f(\boldsymbol{\theta}) + \eta \mathbf{g}^\top \mathbf{d}$
  - minimize $\phi(\eta) = f(\boldsymbol{\theta}_k + \eta \mathbf{d}_k)$ for obtaining the step size





- Momentum to reduce zig-zag

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \mathbf{g}_k + \mu_k(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})$$

(a.k.a. heavy ball method)

KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Newton's Method

☐ Second order optimization method (use hessian)

- $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \mathbf{H}_k^{-1} \mathbf{g}_k$

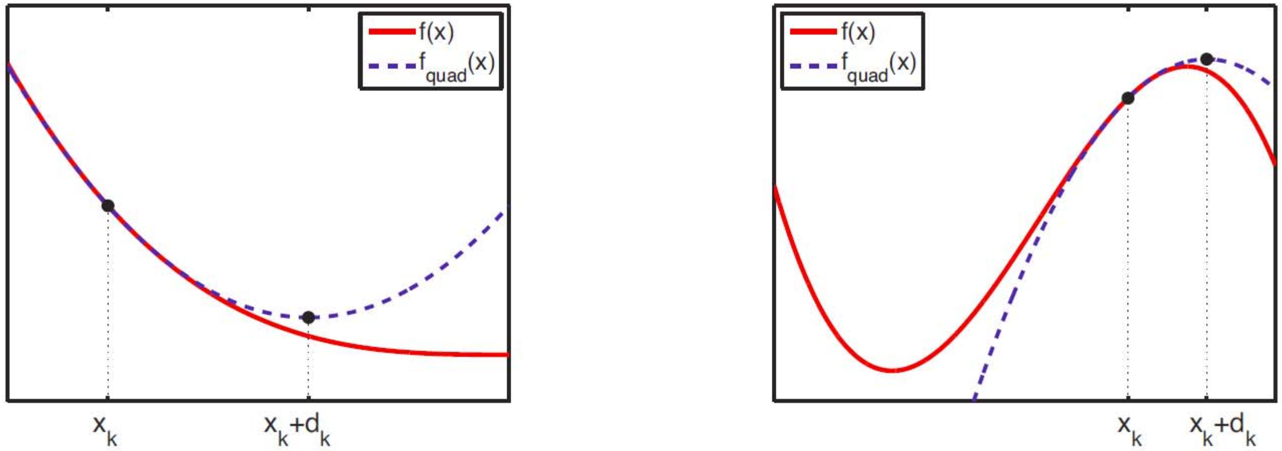
- Consider second-order Taylor expansion:

$$f(\boldsymbol{\theta}) \approx f_k + \mathbf{g}_k^\top (\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}_k)^\top \mathbf{H}_k (\boldsymbol{\theta} - \boldsymbol{\theta}_k)$$

$$= \boldsymbol{\theta}^\top \mathbf{A} \boldsymbol{\theta} + \mathbf{b}^\top \boldsymbol{\theta} + c$$

where $\mathbf{A} = \frac{1}{2}\mathbf{H}_k, \ \mathbf{b} = \mathbf{g}_k - \mathbf{H}_k \boldsymbol{\theta}_k, \ c = f_k - \mathbf{g}_k^\top \boldsymbol{\theta}_k + \frac{1}{2} \boldsymbol{\theta}_k^\top \mathbf{H}_k \boldsymbol{\theta}_k$

- Minimum: $\boldsymbol{\theta} = -\frac{1}{2}\mathbf{A}^{-1}\mathbf{b} = \boldsymbol{\theta}_k - \mathbf{H}_k^{-1} \mathbf{g}_k$



Regularization

- For linearly separable training data, MLE yields $\|\mathbf{w}\| \to \infty$
  - Linear threshold unit $\mathbb{I}(\mathbf{w}^\top \mathbf{x} \geq w_0)$ assigning maximal probability mass to the training data
  - Brittle and does not generalize well

- Regularized objective function
  - $$f'(\mathbf{w}) = \mathrm{NLL}(\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w}$$
    $$\mathbf{g}'(\mathbf{w}) = \mathbf{g}(\mathbf{w}) + \lambda \mathbf{w}$$
    $$\mathbf{H}'(\mathbf{w}) = \mathbf{H}(\mathbf{w}) + \lambda \mathbf{I}$$

Multi-Class Logistic Regression

- i.e. multinomial logistic regression, maximum entropy classifier:

$$p(y = c|\mathbf{x}, \mathbf{W}) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x})}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^\top \mathbf{x})}$$

- Optimization:
  - $\mu_{ic} \equiv p(y_i = c|\mathbf{x}_i, \mathbf{W})$
  - $y_{ic} \equiv \mathbb{I}(y_i = c)$
  - $\mathrm{NLL}(\mathbf{W}) = -\log \prod_{i=1}^{N} \prod_{c=1}^{C} \mu_{ic}^{y_{ic}} = -\sum_i \sum_c y_{ic} \log \mu_{ic}$
    $= -\sum_i \left[\sum_c y_{ic} \mathbf{w}_c^\top \mathbf{x}_i - \log \sum_{c'} \exp(\mathbf{w}_{c'}^\top \mathbf{x}_i)\right]$
  - Compute gradient and hessian:

$$\mathbf{g}_c(\mathbf{W}) = \nabla_{\mathbf{w}_c} \mathrm{NLL}(\mathbf{W}) = \sum_i (\mu_{ic} - y_{ic}) \mathbf{x}_i$$

$$\mathbf{H}_{c,c'}(\mathbf{W}) = \sum_i \mu_{ic} (\delta_{c,c'} - \mu_{i,c'}) \mathbf{x}_i \mathbf{x}_i^\top$$

KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Extending to Bayesian...

- Want $p(\mathbf{w}|\mathcal{D})$ but no conjugate prior $p(\mathbf{w})$
  - MCMC, variational inference, ...

- Laplace approximation
  - Generally, posterior can be re-written $p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{Z}\exp(-E(\boldsymbol{\theta}))$ using energy function $E(\boldsymbol{\theta}) \equiv -\log p(\boldsymbol{\theta}, \mathcal{D})$
  - Taylor expansion of the energy function around mode:

$$E(\boldsymbol{\theta}) \approx E(\boldsymbol{\theta}^*) + (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \mathbf{g} + \tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$$
$$= E(\boldsymbol{\theta}^*) + \tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$$

  - This leads to *Gaussian approximation* to the posterior:

$$\hat{p}(\boldsymbol{\theta}|\mathcal{D}) \approx \frac{1}{Z} e^{-E(\boldsymbol{\theta}^*)} \exp\left[-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)\right] = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}^*, \mathbf{H}^{-1})$$

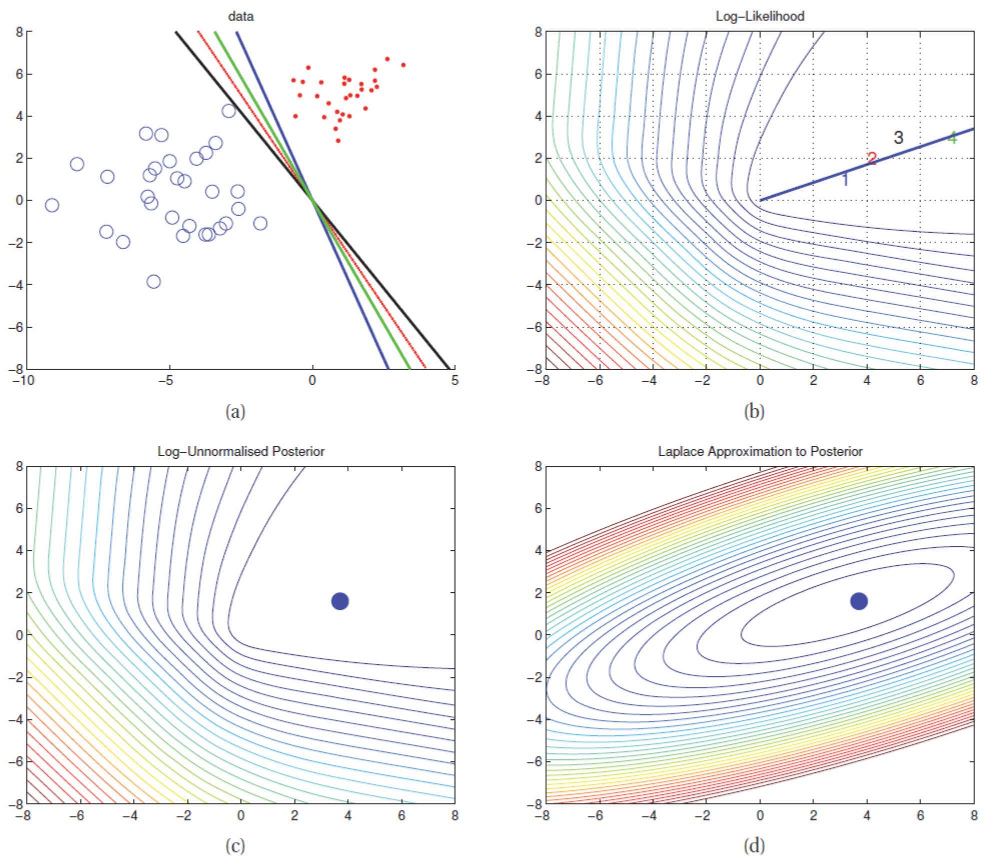
$$Z \approx p(\mathcal{D}) = e^{-E(\boldsymbol{\theta}^*)}(2\pi)^{D/2}|H|^{-\frac{1}{2}}$$

Bayesian Logistic Regression

- Laplace approximation with prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{V}_0)$

$$p(\mathbf{w}|\mathcal{D}) \approx \mathcal{N}(\mathbf{w}|\hat{\mathbf{w}}, \mathbf{H}^{-1})$$

  - $\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} E(\mathbf{w}), \;\; E(\mathbf{w}) = -(\log p(\mathcal{D}|\mathbf{w}) + \log p(\mathbf{w}))$
  - $\mathbf{H} = \nabla^2 E(\mathbf{w})|_{\hat{\mathbf{w}}}$

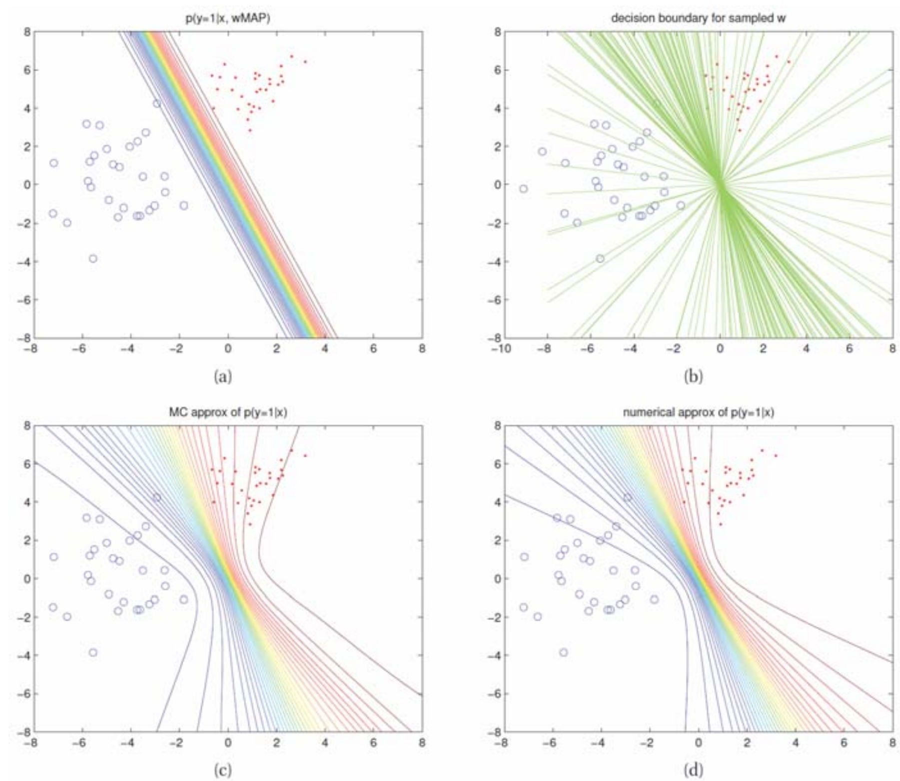


KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Posterior Predictive Distribution

- ☐ Want: $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}$

- ☐ MAP approximation: $p(y|\mathbf{x}, \mathcal{D}) \approx p(y|\mathbf{x}, \mathbf{w}_{\mathrm{MAP}})$
- ☐ Monte-Carlo approximation: using samples $\mathbf{w}^s \sim p(\mathbf{w}|\mathcal{D})$
  - $p(y=1|\mathbf{x}, \mathcal{D}) \approx \frac{1}{S}\sum_{s=1}^{S} p(y=1|\mathbf{x}, \mathbf{w}^s) = \frac{1}{S}\sum_{s=1}^{S} \mathrm{sigm}((\mathbf{w}^s)^\top \mathbf{x})$



KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Extras

* Derivation of BIC

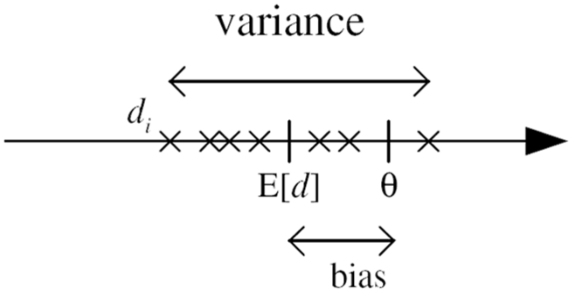
- From $Z \approx p(\mathcal{D}) = e^{-E(\boldsymbol{\theta}^*)}(2\pi)^{D/2}|H|^{-\frac{1}{2}}$
  - we can rewrite:
    $$\log p(\mathcal{D}) \approx \log p(\mathcal{D}, \boldsymbol{\theta}^*) - \frac{1}{2}\log|\mathbf{H}|$$
    $$= \log p(\mathcal{D}|\boldsymbol{\theta}^*) + \log p(\boldsymbol{\theta}^*) - \frac{1}{2}\log|\mathbf{H}|$$
  - Occam's factor: $\log p(\boldsymbol{\theta}^*) - \frac{1}{2}\log|\mathbf{H}|$
    - If we use uniform prior, then the first term can be ignored and $\boldsymbol{\theta}^* = \hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$
  - $\mathbf{H} = \sum_{i=1}^{N} \mathbf{H}_i = \sum_{i=1}^{N} \nabla\nabla \log p(\mathcal{D}_i|\boldsymbol{\theta})$
    - If $\mathbf{H}_i \approx \hat{\mathbf{H}}$, then
      $$\log|\mathbf{H}| \approx \log|N\hat{\mathbf{H}}| = \log N^d|\hat{\mathbf{H}}| = D\log N + \log|\hat{\mathbf{H}}|$$
      where $D = \dim(\boldsymbol{\theta})$

- Hence $p(\mathcal{D}) \approx \log p(\mathcal{D}|\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}) - \dfrac{D}{2}\log N$

KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Bias and Variance of Estimators

- How do we measure the goodness of our estimators?
  - View estimators as learning algorithms
  - Evaluate output of the algorithm on all possible training data
  - More formally, training data is sampled according to $p(\mathcal{D})$
  - The "desired" response for an input: $y^*(x) = E_{p(y|x)}[y]$
- Questions to be asked are:
  - **Bias**: How accurate (on average) are the model predictions?
    - We want the bias to be as small as possible
  - **Variance**: How spread out are the model predictions?
    - We also want the variance to be as small as possible (less subject to noise in the data and initial setting parameter)



KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Bias-Variance Decomposition

- Mean square error at input x: $E_{p(\mathcal{D})}[(g(x|\theta) - r^*(x))^2]$
  - Denote $r_{\mathcal{D}}(x) = g(x|\theta)$
  - $E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - r^*(x))^2]$

$$
\begin{aligned}
&= E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] + E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))^2] \\
&= E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)])^2 + (E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))^2 \\
&\qquad\qquad\qquad + 2(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)])(E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))] \\
&= E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)])^2] + E_{p(\mathcal{D})}[(E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))^2] \\
&\quad + 2E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)])(E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))] \\
&= E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)])^2] + (E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))^2 \\
&\quad + 2E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)])](E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x)) \\
&= \underbrace{E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D}))}[r_{\mathcal{D}}(x)])^2]}_{\text{Variance at x}} + \underbrace{(E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))^2}_{\text{Bias}^2\text{ at x}}
\end{aligned}
$$

Estimating Bias and Variance

- From bias-variance decomposition:
  - $$E_{p(x)}\left[E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - r^*(x))^2]\right]$$
    $$= E_{p(x)}\left[E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)])^2]\right]$$
    $$+ E_{p(x)}\left[(E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))^2\right]$$
  - Generate M additional training sets, each with size N
    - $\mathcal{D}_i = \{x_i^t, r_i^t\}, i = 1, \ldots, M$
  - Use $\mathcal{D}_i$ to fit $g_i(x|\theta)$
  - Now we have:
    - $\mathrm{Bias}^2(g) = \frac{1}{N}\sum_t [\bar{g}(x^t) - f(x^t)]^2$
    - $\mathrm{Variance}(g) = \frac{1}{NM}\sum_t \sum_i [g_i(x^t) - \bar{g}(x^t)]^2$
    - where $\bar{g}(x) = \frac{1}{M}\sum_i g_i(x)$
  - In reality, you can’t compute because you don’t know f!

KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Bias-Variance Dilemma

- From bias-variance decomposition:

$$
\begin{aligned}
&E_{p(x)}\left[E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - r^*(x))^2]\right] \\
&\qquad = E_{p(x)}\left[E_{p(\mathcal{D})}[(r_{\mathcal{D}}(x) - E_{p(\mathcal{D})}[r_{\mathcal{D}}(x)])^2]\right] \\
&\qquad\quad + E_{p(x)}\left[(E_{P(\mathcal{D})}[r_{\mathcal{D}}(x)] - r^*(x))^2\right]
\end{aligned}
$$

- Example:
  - Constant function with no parameter to learn: $g(x) = 2$
    - High bias and no variance
  - Constant function with one parameter to learn: $g(x) = \sum_t r^t / N$
    - Lower bias with some variance
- As we increase complexity of the model (e.g. const -> linear -> poly)
  - Bias decreases (a better fit to the data)
  - Variance increases (fit varies more with data)
- This is called Bias-Variance dilemma (Geman et al., 1992)

KAIST
Korea Advanced Institute of Science and Technology
한국과학기술원

Model Complexity and Bias-Variance