Artificial Intelligence

Course Review

Donghui Wang
AI Institute@ZJU

2015.4

Contents

- About final exam
- Basic concepts
- Important formulas and derivations
- Learning Algorithms

References:

1. *AI Course slides, http://10.15.62.79/cv*
2. *Christopher M. Bishop. "Pattern Recognition and Machine Learning", 2006, Springer.*
3. *Stuart J. Russell and Peter Norvig. "Artificial Intelligence: A Modern Approach (Third Edition)". 2009, Prentice Hall.*
4. *http://cs229.stanford.edu/info.html, by Prof. Andrew Ng*

Course Review

ABOUT FINAL EXAM

Final Exam

1. Fill in the blank (30 points, 2pt/per)
   - Basic concept, definition and fundamental knowledge
2. Multiple Choice (20 points, 2pt/per)
   - Same as first part, but focus on the difference among important concepts and definitions.
3. Calculus, Analysis and Proof (50 points, 10~15pt/per)
4. Algorithm Design (10 points)

Final Exam

- Information:

Date and Time	04 May, 14:00pm~16:00pm (2 hours)
Location	Teaching Building #7, Room 108 (Multimedia) Yuquan Campus
Number of Examination	30
Examination form	Closed Book
Final Q&A Time	29 April, 14:00-17:00
Final Q&A Room	CaoGuangBiao Building, Room 608

Course Review

Basic concepts

Basic concepts

1. What is AI?
   - Acting humanly: the Turing test approach
   - Thinking humanly: the cognitive modeling approach
   - Thinking rationally: the “laws of thought” approach
   - Acting rationally: the rational agent approach
2. Turing test
   - A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a computer.
3. Strong AI and Weak AI
   - Philosophers use the term weak AI for the hypothesis that machines could possibly behave intelligently, and strong AI for the hypothesis that such machines would count as having actual minds.

Basic concepts

4. Supervised learning approach (predictive approach)
   - Regression, classification (SVM)
   - Training phase (learning phase), prediction phase
   - Training set (with target vector), test set
   - Binary classification, multiclass classification

5. Unsupervised learning approach (descriptive approach)
   - Knowledge discovery
   - Clustering, density estimation, dimensionality reduction (manifold learning)
   - Training set (without target vector), test set

6. Reinforcement learning

Basic concepts

7. Error function

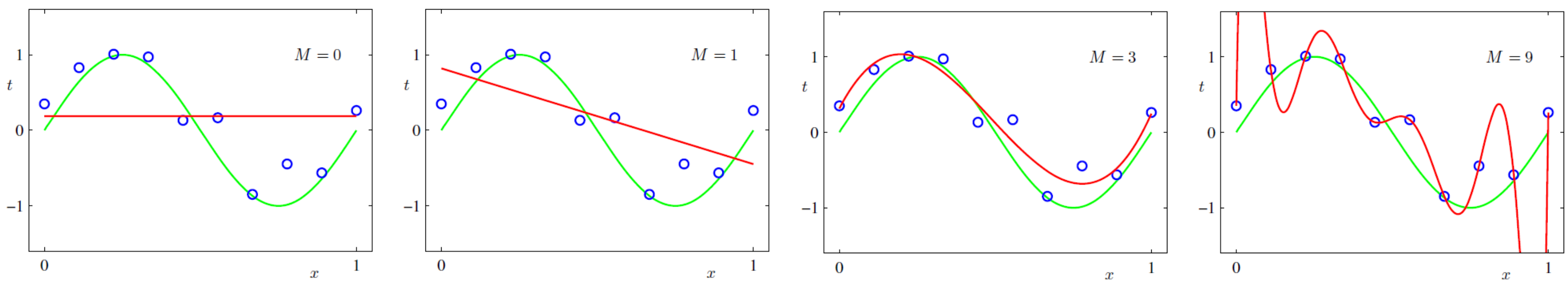
– SSE (the sum-of-squares error)

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

– RMS (the root-mean-square error)

$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$$

8. Over-fitting phenomenon



– How to control?

– *Regularization (penalty term)*, *Bayesian approach (prior)*, *CV*…

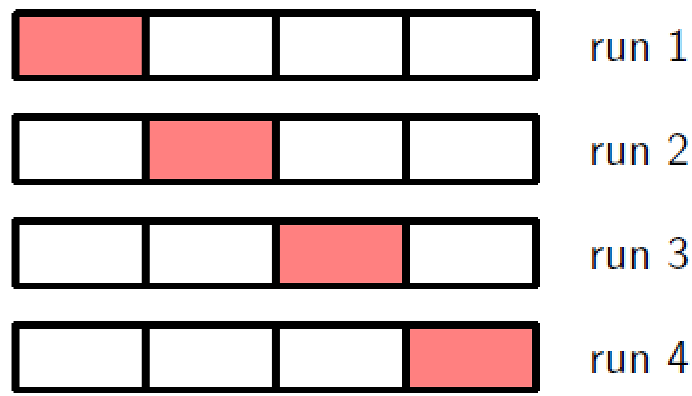
– *Shrinkage methods: ridge regression (weight decay)*

Basic concepts

9. Model comparison (model selection)

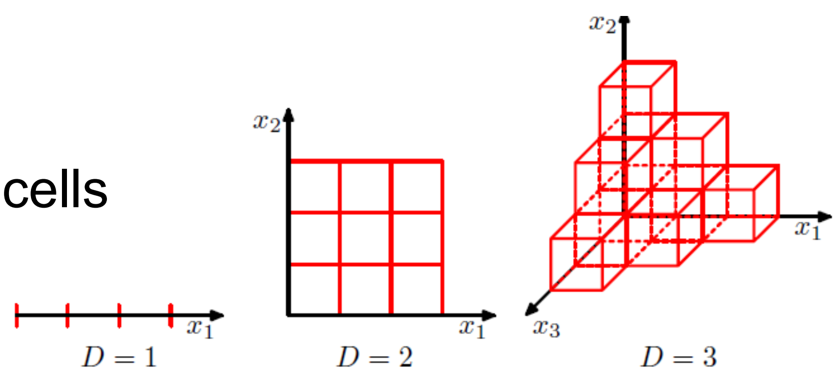
– validation set, Cross-validation (CV)

The technique of S-fold cross-validation, illustrated here for the case of $S = 4$, involves taking the available data and partitioning it into S groups (in the simplest case these are of equal size). Then $S - 1$ of the groups are used to train a set of models that are then evaluated on the remaining group. This procedure is then repeated for all S possible choices for the held-out group, indicated here by the red blocks, and the performance scores from the S runs are then averaged.



10. The curse of dimensionality

– Divide the input space into regular cells



Basic concepts

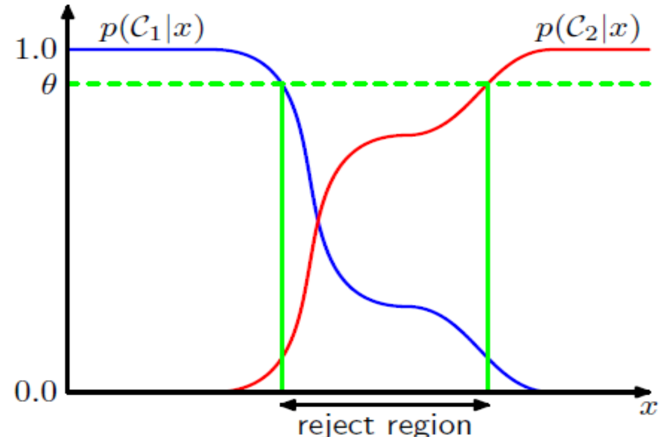
11. Frequentist statistics vs. Bayesian statistics
    - View probabilities in terms of the frequencies of random, repeatable events.
    - Probabilities provide a quantification of uncertainty and make rational coherent inference.
12. Likelihood function $p(\mathcal{D}|\mathbf{w})$ $p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$
    - It expresses how probable the observed data set is for different settings of the parameter vector w
    - The likelihood is not a probability distribution over w, and its integral with respect to w does not (necessarily) equal one.
    - Maximum Likelihood – ML (widely used frequentist estimator)
        - *Maximizing the likelihood is equivalent to minimizing the error (e.g. SSE)*.
        - i.i.d (independent and identically distributed)
    - Maximum posterior – MAP

Basic concepts

13. Decision regions, decision boundaries (decision surface)
14. Linearly separable
15. Loss function (cost function, - utility function), loss matrix
    - Minimize the average loss:
        - L_{kj}=0, for k=j. others 1.

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}.$$

16. Reject option
    - Threshold θ
    - $\theta = 1$: reject all
    - For K classes, $\theta < 1/K$: no examples rejected



Basic concepts

17. Discriminant function
18. Generative models and discriminative models
19. Relative entropy (Kullback-Leibler divergence or KL divergence)

$$\mathrm{KL}(q\|p) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})} \right\}$$

20. Conjugacy and Conjugate priors
    - Posterior distribution has the same functional form as the prior.
    - The mean of a Gaussian ↔ another Gaussian
    - Beta distribution ↔ Binomial distribution
    - Dirichlet distribution ↔ multinomial distribution

Basic concepts

21. Exponential family of distribution
22. Mahalanobis distance

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

23. Covariance matrix and precision matrix
    - The inverse of the covariance matrix
24. Parametric and non-parametric models
    - Probability distributions have specific functional forms governed by a small number of parameters.
    - There is few assumptions about the form of the distribution.
25. Jacobian matrix

Basic concepts

26. Linear models

- Linear model $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D$ where $\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$
- Linear basis function model
$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$
- Generalized linear models
  - Activation function and link function $y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right)$
  - logistic sigmoid function and logit function

27. The Fisher's criterion

- maximize the separation between the projected class means as well as the inverse of the total within-class variance.
- Generalized Rayleigh quotient, Between-class covariance matrix and Within-class covariance matrix
$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}} \qquad J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

28. The perceptron criterion: $E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}_n t_n$

Basic concepts

29. The probit regression:
    – Use CDF of N(x|0,1) to construct an activation function (Inverse probit function)
30. The Laplace Approximation
    – Find a Gaussian approximation q(z) which is centred on a mode of the distribution p(z)
31. kernel function $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{x}')$
    – symmetric function
    – How to construct kernel functions?
    – Kernel trick
    – The simplest kernel function: $k(x,y) = x^{T}y$

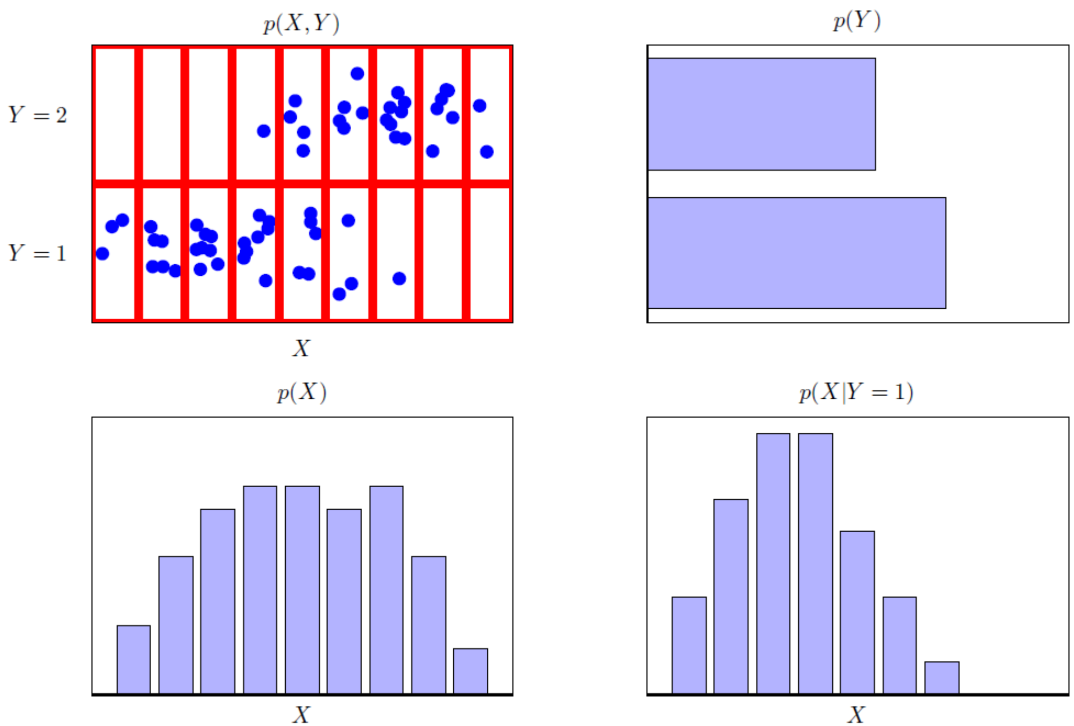
Course Review

IMPORTANT FORMULAS AND DERIVATIONS

Probability theory

- Marginal probability
$$p(X) = \sum_Y p(X,Y)$$
- Joint probability
$$p(X,Y) = p(Y|X)p(X)$$
- Conditional probability
- Bayes' theorem

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$



Multivariate Gaussian Distribution

- Definition: $$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$$\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

- *The matrix Σ can be taken to be symmetric, without loss of generality.* Σ^{-1} *is symmetric.*

- *The eigenvector equation for the covariance matrix:*

$$\boldsymbol{\Sigma}\mathbf{u}_i = \lambda_i\mathbf{u}_i \quad \text{where } i = 1,\ldots,D \qquad \mathbf{u}_i^{\mathrm{T}}\mathbf{u}_j = I_{ij} \qquad I_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \qquad \mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{I}$$

$$\Rightarrow \boldsymbol{\Sigma}^{-1} = \mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^{\mathrm{T}} \qquad \Rightarrow \Delta^2 = \sum_{i=1}^{D}\frac{y_i^2}{\lambda_i} \qquad y_i = \mathbf{u}_i^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}) \qquad \mathbf{y} = \mathbf{U}(\mathbf{x}-\boldsymbol{\mu})$$

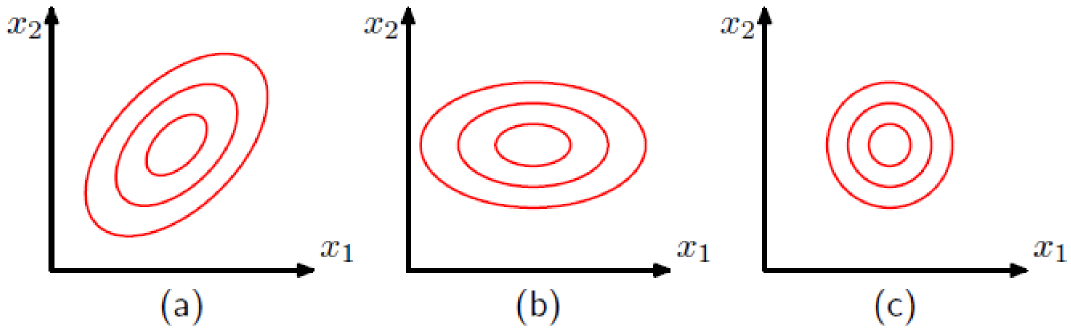
$$\Rightarrow |\boldsymbol{\Sigma}|^{1/2} = \prod_{j=1}^{D}\lambda_j^{1/2} \qquad \Rightarrow |\mathbf{J}| = 1$$

Multivariate Gaussian Distribution

- Definition: $$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \qquad \mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \boldsymbol{\Sigma} \qquad \operatorname{cov}[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x}-\mathbb{E}[\mathbf{x}])(\mathbf{x}-\mathbb{E}[\mathbf{x}])^{\mathrm{T}}\right]$$

$$\Rightarrow \operatorname{cov}[\mathbf{x}] = \boldsymbol{\Sigma}$$



- A general symmetric covariance matrix Σ will have D(D + 1)/2 independent parameters, and there are another D independent parameters in μ, giving D(D + 3)/2 parameters in total.
  - 2D independent parameters $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_i^2)$
  - isotropic covariance, D + 1 independent parameters $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$

Multivariate Gaussian Distribution

- Bayes' Theorem for Gaussian Variables:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1})$$

$$p(\mathbf{z}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \qquad \mathbf{z} = \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}$$

$$\boxed{\begin{pmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{pmatrix}^{-1} = \begin{pmatrix}\mathbf{M} & -\mathbf{M}\mathbf{B}\mathbf{D}^{-1}\\ -\mathbf{D}^{-1}\mathbf{C}\mathbf{M} & \mathbf{D}^{-1}+\mathbf{D}^{-1}\mathbf{C}\mathbf{M}\mathbf{B}\mathbf{D}^{-1}\end{pmatrix} \quad \mathbf{M} = (\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}}$$

$$\boxed{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = -\frac{1}{2}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \text{const}}$$

$$\begin{aligned}\ln p(\mathbf{z}) &= \ln p(\mathbf{x}) + \ln p(\mathbf{y}|\mathbf{x})\\ &= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Lambda}(\mathbf{x}-\boldsymbol{\mu}) - \frac{1}{2}(\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{b})^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{b}) + \text{const}\end{aligned}$$

$$\begin{aligned}&-\frac{1}{2}\mathbf{x}^{\mathrm{T}}(\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})\mathbf{x} - \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{L}\mathbf{y} + \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{L}\mathbf{A}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{y}\\ &= -\frac{1}{2}\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A} & -\mathbf{A}^{\mathrm{T}}\mathbf{L}\\ -\mathbf{L}\mathbf{A} & \mathbf{L}\end{pmatrix}\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = -\frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{R}\mathbf{z}\end{aligned}$$

$$\mathbf{R} = \begin{pmatrix}\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A} & -\mathbf{A}^{\mathrm{T}}\mathbf{L}\\ -\mathbf{L}\mathbf{A} & \mathbf{L}\end{pmatrix}$$

$$\operatorname{cov}[\mathbf{z}] = \mathbf{R}^{-1} = \begin{pmatrix}\boldsymbol{\Lambda}^{-1} & \boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}\\ \mathbf{A}\boldsymbol{\Lambda}^{-1} & \mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}\end{pmatrix}$$

$$\mathbf{x}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{\mu} - \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{b} + \mathbf{y}^{\mathrm{T}}\mathbf{L}\mathbf{b} = \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}\boldsymbol{\Lambda}\boldsymbol{\mu}-\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{b}\\ \mathbf{L}\mathbf{b}\end{pmatrix}$$

$$\mathbb{E}[\mathbf{z}] = \mathbf{R}^{-1}\begin{pmatrix}\boldsymbol{\Lambda}\boldsymbol{\mu}-\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{b}\\ \mathbf{L}\mathbf{b}\end{pmatrix} = \begin{pmatrix}\boldsymbol{\mu}\\ \mathbf{A}\boldsymbol{\mu}+\mathbf{b}\end{pmatrix}$$

Maximum likelihood for the Gaussian

- Given a data set $\mathbf{X} = (\mathrm{x}_1, \ldots, \mathrm{x}_N)^{\mathrm{T}}$ in which the observations $\{\mathrm{x}_n\}$ are assumed to be drawn independently from a multivariate Gaussian distribution:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$$\Rightarrow \quad \ln p(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n-\boldsymbol{\mu})$$

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu})(\mathbf{x}_n-\boldsymbol{\mu})^{\mathrm{T}}$$

$$\boxed{\frac{\partial \mathbf{a}^T\mathbf{X}^{-1}\mathbf{b}}{\partial \mathbf{X}} = -\mathbf{X}^{-T}\mathbf{a}\mathbf{b}^T\mathbf{X}^{-T}}$$

$$\frac{\partial}{\partial\boldsymbol{\mu}}\ln p(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{n=1}^{N}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}) \quad \Rightarrow \quad \boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

$$\boxed{\frac{\partial}{\partial \mathbf{A}}\ln|\mathbf{A}| = \left(\mathbf{A}^{-1}\right)^{\mathrm{T}}}$$

$$\frac{\partial}{\partial\boldsymbol{\Sigma}}\ln p(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = -\frac{N}{2}\frac{\partial}{\partial\boldsymbol{\Sigma}}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\frac{\partial}{\partial\boldsymbol{\Sigma}}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}) \quad \Rightarrow \quad \boldsymbol{\Sigma} = \mathbf{S}$$

$$\Rightarrow \quad \boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n-\boldsymbol{\mu}_{\mathrm{ML}})^{\mathrm{T}}$$

$$\boxed{\begin{aligned}\mathbb{E}[\boldsymbol{\mu}_{\mathrm{ML}}] &= \boldsymbol{\mu}\\ \mathbb{E}[\boldsymbol{\Sigma}_{\mathrm{ML}}] &= \frac{N-1}{N}\boldsymbol{\Sigma}\end{aligned}}$$

Bayesian Inference for the Gaussian

1. Known the variance, to infer the mean:

Likelihood: $$p(\mathbf{X}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 \right\}$$

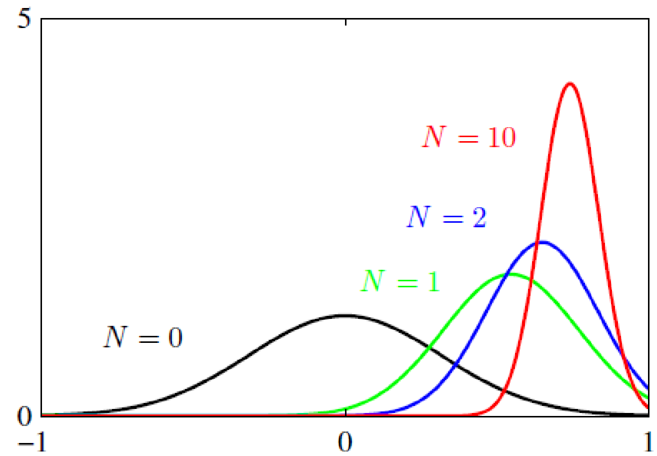
Prior: $$p(\mu) = \mathcal{N}\left(\mu|\mu_0, \sigma_0^2\right)$$

Posterior: $$p(\mu|\mathbf{X}) \propto p(\mathbf{X}|\mu)p(\mu)$$

$$p(\mu|\mathbf{X}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$$

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{\mathrm{ML}}$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}$$



Likelihood: $$p(\mathbf{X}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right\}$$

Prior: $$p(\mu) = \mathcal{N}\left(\mu|\mu_0, \sigma_0^2\right)$$

Posterior: $$p(\mu|\mathbf{X}) \propto p(\mathbf{X}|\mu)p(\mu) \qquad p(\mu|\mathbf{X}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$$

$$\boxed{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = -\frac{1}{2}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \text{const}}$$

$$\begin{aligned}
&-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2 - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 \\
&\qquad = -\frac{\mu^2}{2}\left(\frac{1}{\sigma_0^2}+\frac{N}{\sigma^2}\right) + \mu\left(\frac{\mu_0}{\sigma_0^2}+\frac{1}{\sigma^2}\sum_{n=1}^{N}x_n\right) + \text{const}
\end{aligned}$$

$$\frac{1}{\sigma_N^2} = \frac{N}{\sigma^2} + \frac{1}{\sigma_0^2} \qquad \mu_N = \left(\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}\right)^{-1}\left(\frac{\mu_0}{\sigma_0^2}+\frac{1}{\sigma^2}\sum_{n=1}^{N}x_n\right)$$

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}x_n \qquad\qquad = \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\mu_{\mathrm{ML}}.$$

Bayesian Inference for the Gaussian

2. Known the mean, to infer the variance: $\lambda \equiv 1/\sigma^2$

Likelihood: $$p(\mathbf{X}|\lambda) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu, \lambda^{-1}) \propto \lambda^{N/2} \exp\left\{-\frac{\lambda}{2}\sum_{n=1}^{N}(x_n-\mu)^2\right\}$$

Prior: $\mathrm{Gam}(\lambda|a_0, b_0)$ *gamma distribution*

Posterior: $p(\lambda|\mathbf{X}) \propto p(\mathbf{X}|\lambda)\,\mathrm{Gam}(\lambda|a_0, b_0)$

$$\boxed{\mathrm{Gam}(\lambda|a,b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda)}$$

$$p(\lambda|\mathbf{X}) \propto \lambda^{a_0-1}\lambda^{N/2} \exp\left\{-b_0\lambda - \frac{\lambda}{2}\sum_{n=1}^{N}(x_n-\mu)^2\right\} \Rightarrow \mathrm{Gam}(\lambda|a_N, b_N)$$

$$a_N = a_0 + \frac{N}{2}$$

$$b_N = b_0 + \frac{1}{2}\sum_{n=1}^{N}(x_n-\mu)^2 = b_0 + \frac{N}{2}\sigma^2_{\mathrm{ML}}$$

Maximum likelihood and least squares

- Assume: $$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$
- Thus: $$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \longrightarrow \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x}) \, \mathrm{d}t = y(\mathbf{x}, \mathbf{w})$$
- For data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and target vector $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$, the likelihood function:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1})$$

$$\ln p(\mathbf{t}|\mathbf{w}, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\mathbf{w})$$

SSE: sum-of-squares error function

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

Maximum likelihood and least squares

$$\ln p(\mathbf{t}|\mathbf{w},\beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n),\beta^{-1}) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w})$$

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

- Solving w by ML:

$$\nabla \ln p(\mathbf{t}|\mathbf{w},\beta) = \sum_{n=1}^{N}\left\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}$$

$$0 = \sum_{n=1}^{N} t_n \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} - \mathbf{w}^{\mathrm{T}}\left(\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}\right)$$

$$\Rightarrow \quad \mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

N×M design matrix

$$\boldsymbol{\Phi}^{\dagger} \equiv \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}$$

Moore-Penrose pseudo-inverse

Two classes

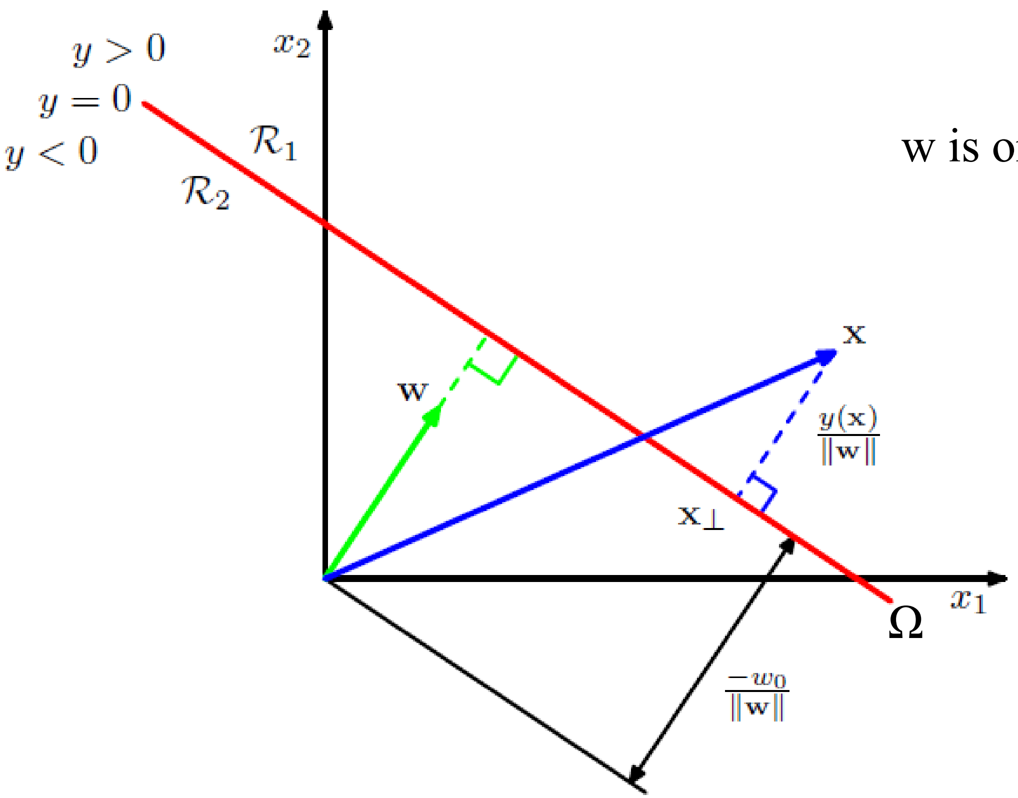
- Linear discriminant function: $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$
  - if $y(\mathbf{x}) \geq 0$, assign $\mathbf{x}$ to class C_1, else class C_2
  - decision surface Ω: $y(\mathbf{x}) = 0$
  - the normal distance from the origin to the decision surface: $\frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}$

x_A and x_B lie on the decision surface: $y(\mathbf{x}_A) = y(\mathbf{x}_B) = 0$

$$\mathbf{w}^{\mathrm{T}}(\mathbf{x}_A - \mathbf{x}_B) = 0$$

w is orthogonal to every vector lying within Ω

$\frac{\mathbf{w}}{\|\mathbf{w}\|}$ is the normal vector of Ω



$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|} \Rightarrow r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$$

$$\widetilde{\mathbf{w}} = (w_0, \mathbf{w}) \qquad \widetilde{\mathbf{x}} = (x_0, \mathbf{x})$$

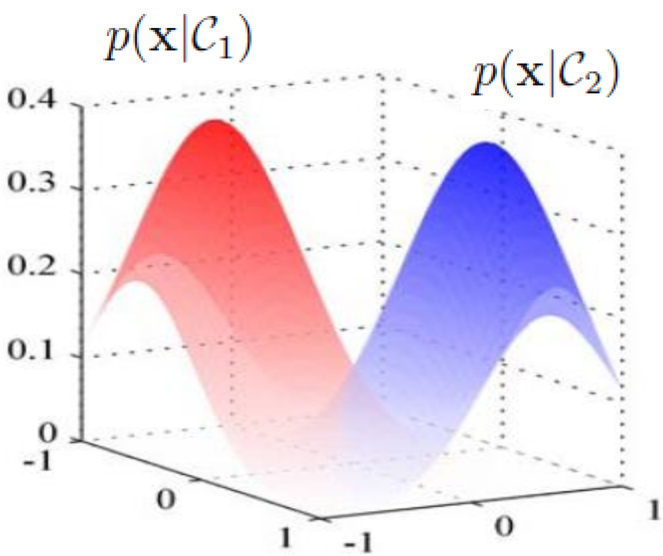
$$y(\mathbf{x}) = \widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

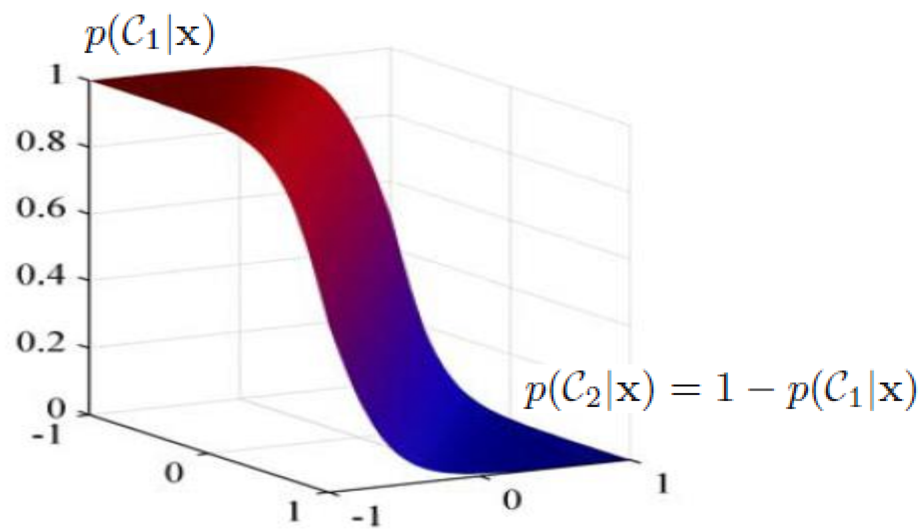
Probabilistic Generative Models: Continuous inputs

- *Assume:* $$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

- *2 classes:* $$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$

$$\Rightarrow \quad \mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \qquad w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$





Logistic regression

- Logistic regression model:
  - Only M parameters need to be estimated.

logistic sigmoid function

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma\left(\mathbf{w}^{\mathrm{T}}\phi\right) \qquad p(\mathcal{C}_2|\phi) = 1 - p(\mathcal{C}_1|\phi) \qquad \sigma(a) = \frac{1}{1+\exp(-a)}$$

  - For a data set $\{\phi_n, t_n\}$, where $t_n \in \{0,1\}$ and $\phi_n = \phi(\mathbf{x}_n)$, $n = 1, \ldots, N$, the likelihood function can be written

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}\left\{1-y_n\right\}^{1-t_n}$$ where $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ and $y_n = p(\mathcal{C}_1|\phi_n)$.

  - Cross-entropy error function:

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N}\left\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\right\}$$

$$y_n = \sigma(a_n)$$

$$a_n = \mathbf{w}^{\mathrm{T}}\phi_n$$

$$\frac{d\sigma}{da} = \sigma(1-\sigma)$$

$$\left\{\begin{aligned} \frac{\partial E}{\partial y_n} &= \frac{1-t_n}{1-y_n} - \frac{t_n}{y_n} = \frac{y_n(1-t_n) - t_n(1-y_n)}{y_n(1-y_n)} = \frac{y_n - t_n}{y_n(1-y_n)} \\ \frac{\partial y_n}{\partial a_n} &= \frac{\partial \sigma(a_n)}{\partial a_n} = \sigma(a_n)\left(1-\sigma(a_n)\right) = y_n(1-y_n) \end{aligned}\right.$$

No closed-form solution

$$\boxed{\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \frac{\partial E}{\partial y_n}\frac{\partial y_n}{\partial a_n}\nabla a_n = \sum_{n=1}^{N}(y_n - t_n)\phi_n = \sum_{n=1}^{N}\left(\sigma(\mathbf{w}^{\mathrm{T}}\phi_n) - t_n\right)\phi_n}$$

Course Review

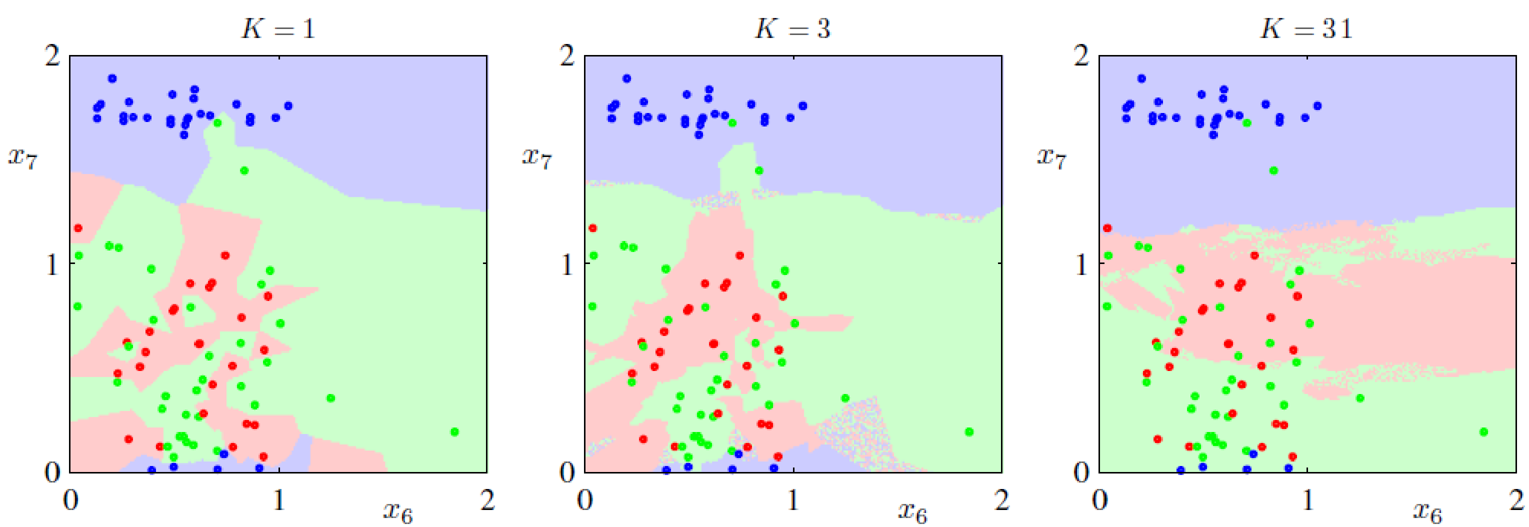
LEARNING ALGORITHMS

Nearest-neighbour methods

- KNN density estimation
  - K govern the radius of the sphere
- KNN classifier

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{K_k}{N_k V} \qquad p(\mathbf{x}) = \frac{K}{NV} \qquad p(\mathcal{C}_k) = \frac{N_k}{N}$$

$$\Rightarrow \quad p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} = \frac{K_k}{K}$$



Sequential learning

- Stochastic gradient descent (sequential gradient descent)

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \qquad n = 1, 2, \ldots, N$$

Learning rate

Error function

  - least-mean-squares or the LMS algorithm

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2 \Rightarrow E_n(\mathbf{w}) = \frac{1}{2}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2$$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \Rightarrow \mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\phi_n)\phi_n$$

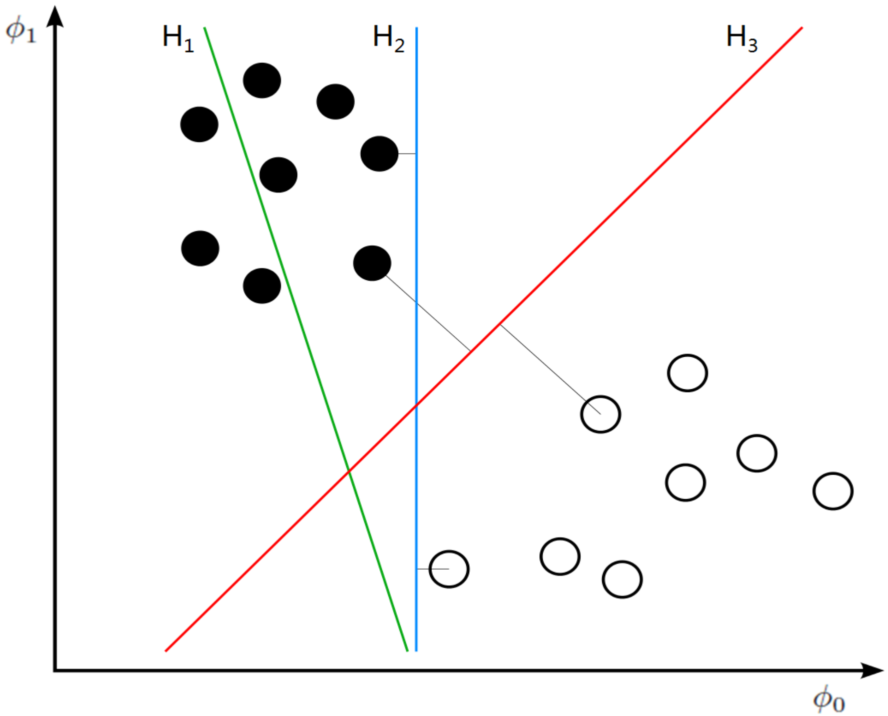
Maximum margin classifiers

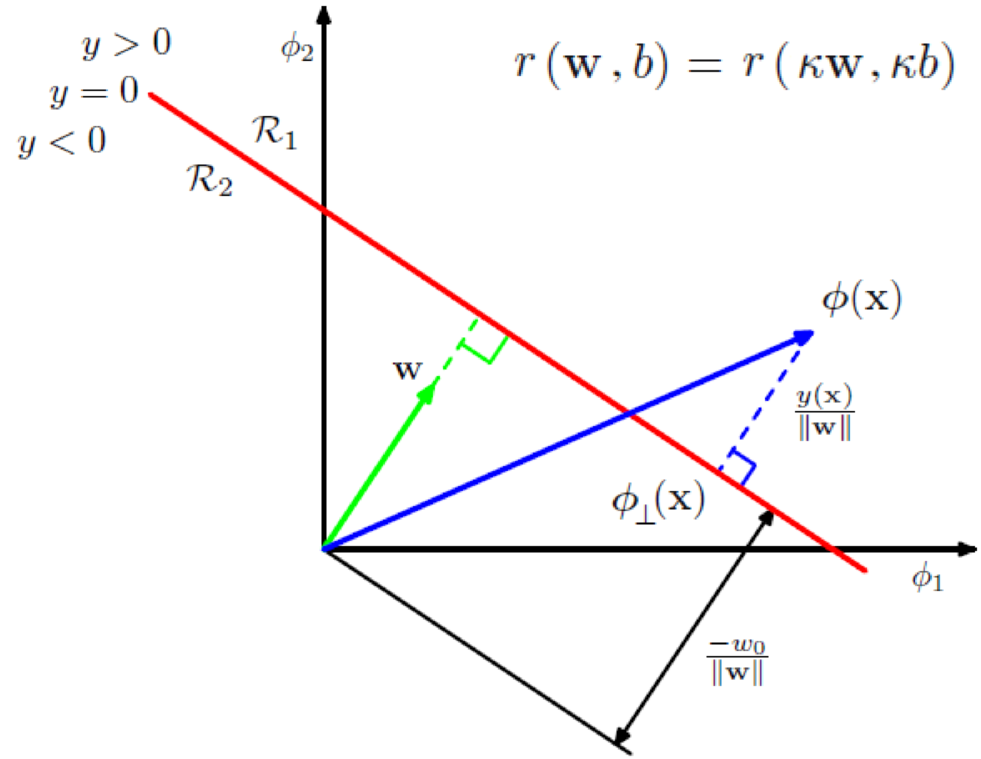
- *Support Vector Machines (SVM) learning algorithm*:
  1. *Choose a kernel function, e.g. Gaussian kernel function.*
  2. *Use SMO algorithm to solve* $\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$
  3. *Select support vectors with* $a_n > 0$
  4. *Compute the threshold parameter b by using SV set:* $b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$
  5. *Use SV set to classify new data point:* $y(\mathbf{x}) = \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b$

Maximum margin classifiers

- For the two-class classification problem using linear models: $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b$
  - Assume the training data set is linearly separable in feature space;
  - And all data points are correctly classified, so that: $t_n y(\mathbf{x}_n) > 0$ $\quad$ $t_n \in \{-1, 1\}$
- The definition of the *Margin* (rescaling w and b doesn't change *r*)

$$r(\mathbf{w}, b) = \min_n \left\{ \frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} \right\} = \min_n \left\{ \frac{t_n(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|} \right\}$$





Maximum margin classifiers

- Optimization problem: *find the solution of the maximum margin*

$$\arg\max_{\mathbf{w},b} \; r(\mathbf{w}, b) \iff \arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n \left(\mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}_n) + b \right) \right] \right\}$$

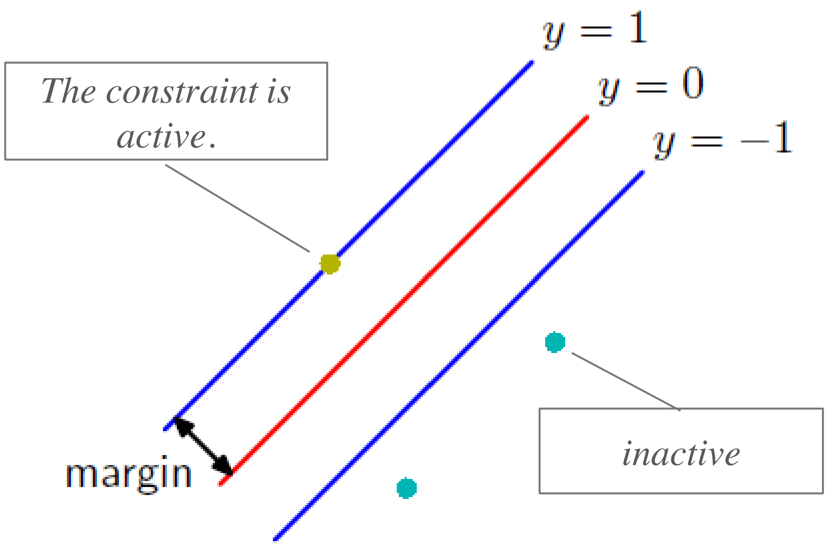
  – Because $r(\mathbf{w}, b) = r(\kappa\mathbf{w}, \kappa b)$, so we can set $t_n \left(\mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}_n) + b \right) = 1$ for the point that is closest to the decision surface. Then, all data points will satisfy the constraints: $t_n y(\mathbf{x}_n) \geqslant 1$

- Equivalent constrained optimization problem:

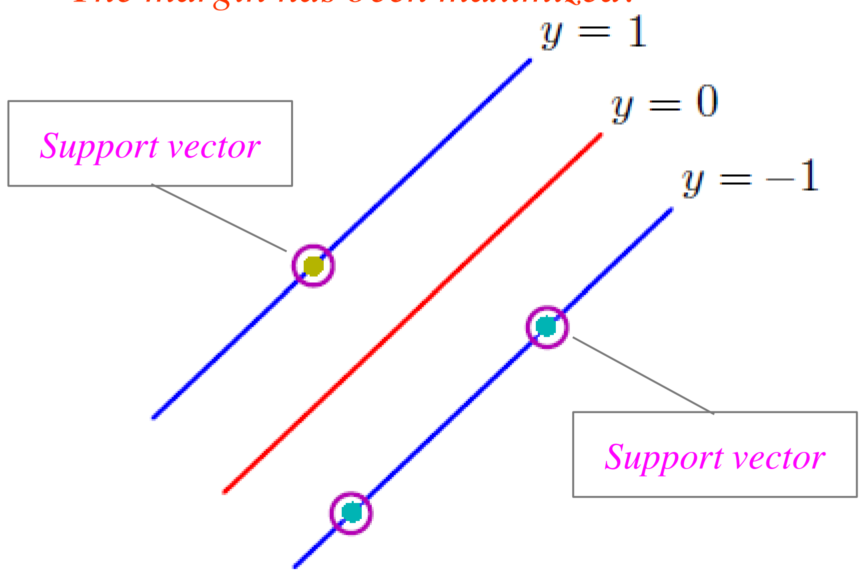
$$\arg\max_{\mathbf{w},b} \; r(\mathbf{w}, b) \iff \arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \right\} \iff \arg\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2$$

subject to $t_n y(\mathbf{x}_n) \geqslant 1, \; n = 1, \ldots, N.$

The margin has not been maximized.



The margin has been maximized.



K-means clustering

- Distortion measure (responsibilities):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

Responsibilities

Data

Prototypes (expected value)

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = 2 \sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

$$r_{n,k} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Example: 5 data points and 3 clusters

K-means algorithm (batch version):

1. Pick number of clusters *k*
2. Randomly scatter *k* “cluster centers” in data space
3. Repeat:
   a. Assign each data point to its closest cluster center
   b. Move each cluster center to the mean of the points assigned to it

K-means clustering

- Online k-means algorithm (sequential k-means):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

The nearest prototype to $\mathbf{x}_n$

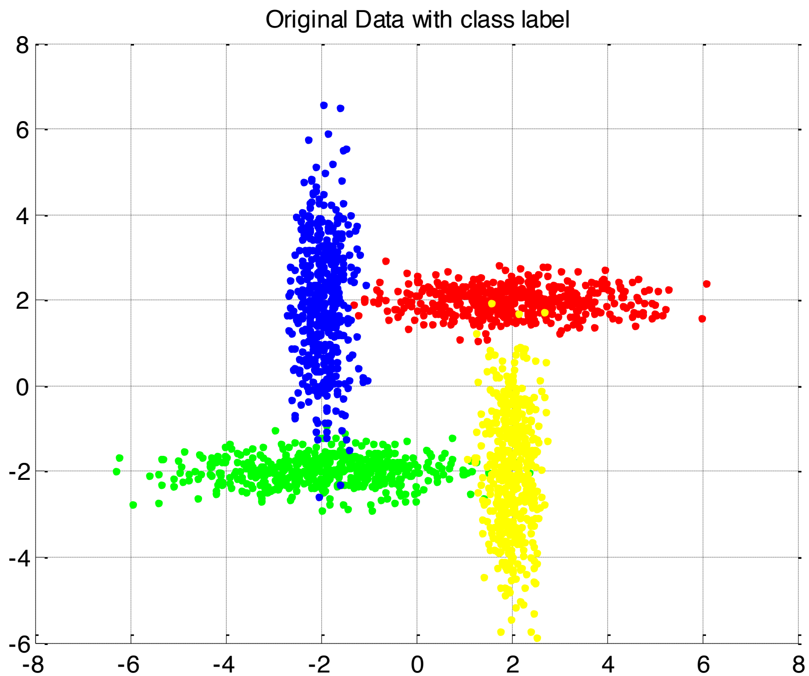
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}} \Rightarrow \boldsymbol{\mu}_k^{\text{new}} = \boldsymbol{\mu}_k^{\text{old}} + \eta_n(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{old}})$$

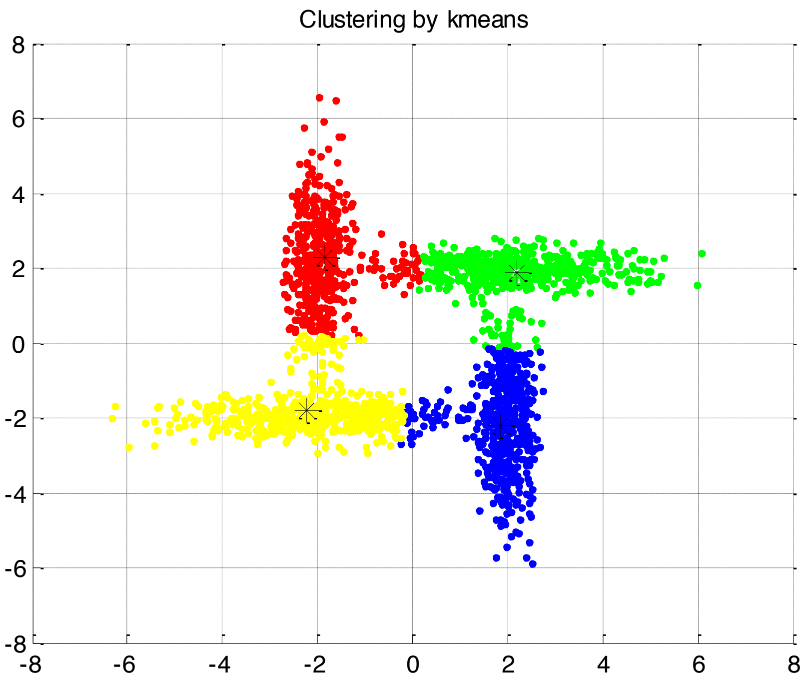
- K-medoids algorithm:
  - Chooses input data points as centers;
  - Works with an arbitrary matrix of distances between data points instead of Euclidean distance.
    - E.g. Manhattan distance or Minkowski distance

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \Rightarrow \widetilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$$

The limitation of K-means clustering

- The K-means algorithm adopts the hard assignment and doesn’t consider the data density and probabilistic distribution.





Expectation-Maximization algorithm for GMM

EM for Gaussian Mixtures

1. Initialize the means $\boldsymbol{\mu}_k$, covariances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood.

2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j)}$$

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \quad \text{where} \quad N_k = \sum_{n=1}^{N}\gamma(z_{nk}).$$

4. Evaluate the log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)\right\}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

The general EM algorithm

The General EM Algorithm

Given a joint distribution $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ over observed variables $\mathbf{X}$ and latent variables $\mathbf{Z}$, governed by parameters $\boldsymbol{\theta}$, the goal is to maximize the likelihood function $p(\mathbf{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

1. Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}}$.
2. **E step** Evaluate $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$.
3. **M step** Evaluate $\boldsymbol{\theta}^{\text{new}}$ given by $\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$

   where $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$. $\Rightarrow$ $\boxed{\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \ln p(\boldsymbol{\theta})}$

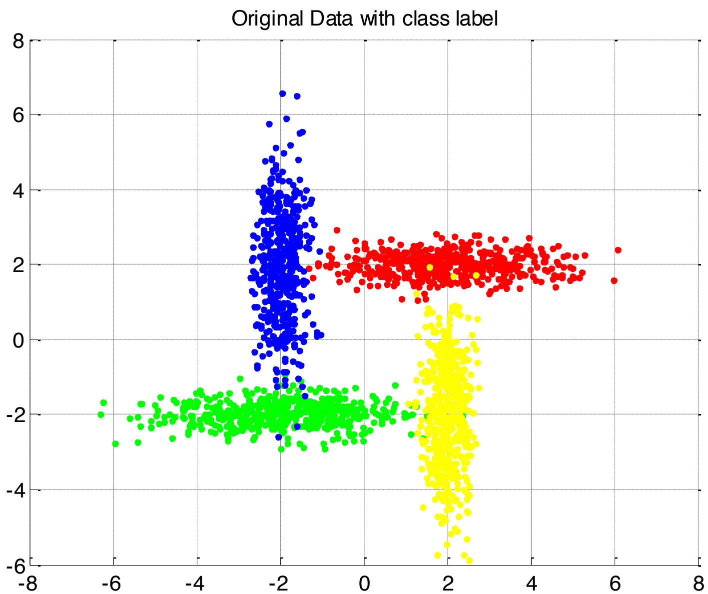
   EM algorithm can be used to find MAP solution

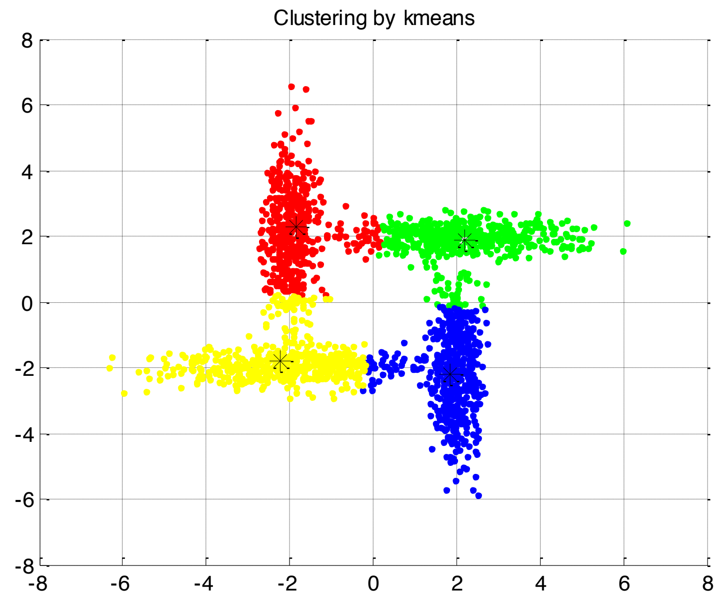
4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

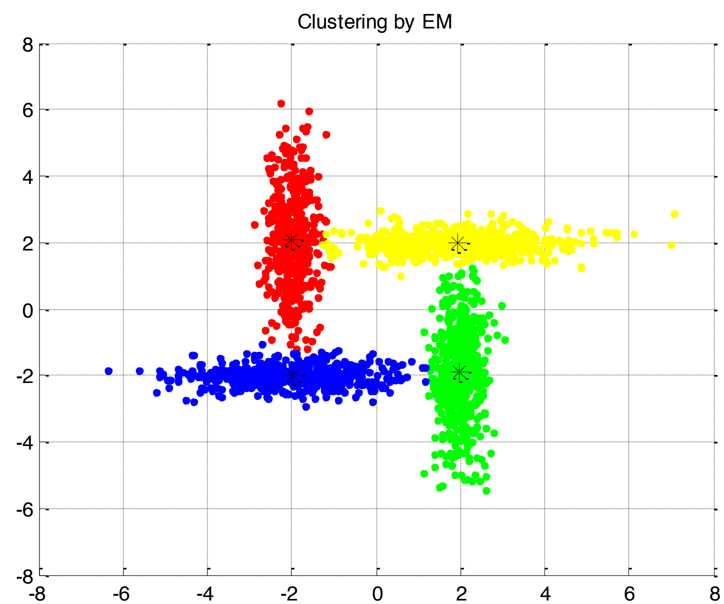
$$\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$$

   and return to step 2.

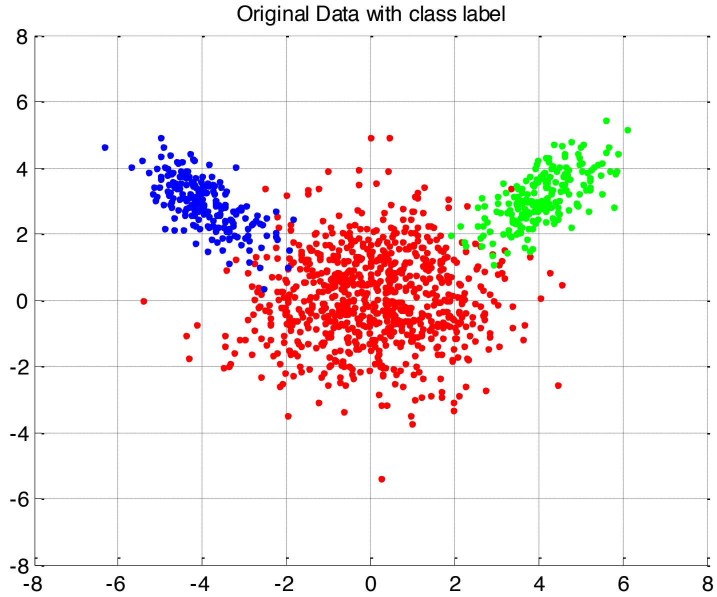
EM for GMM vs. K-means

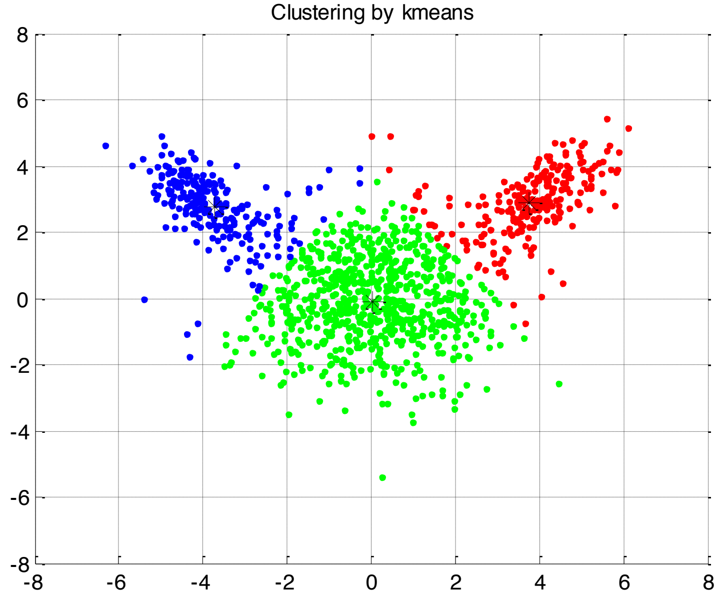


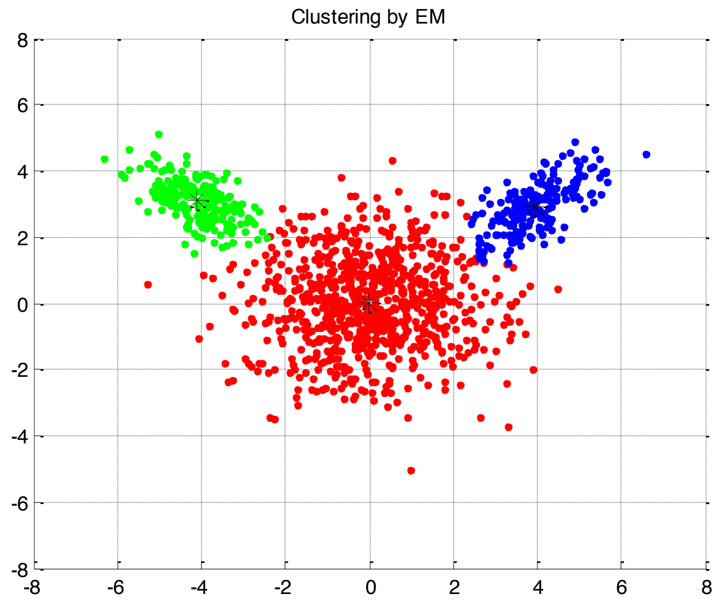




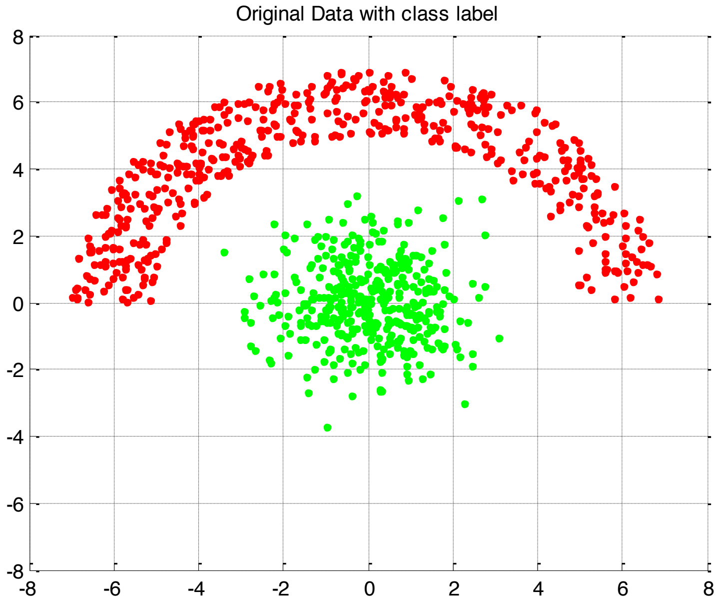
EM for GMM vs. K-means

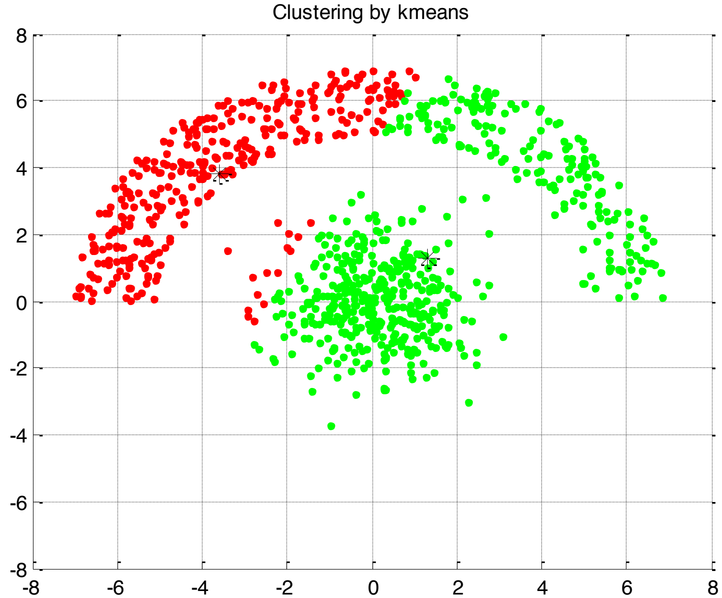


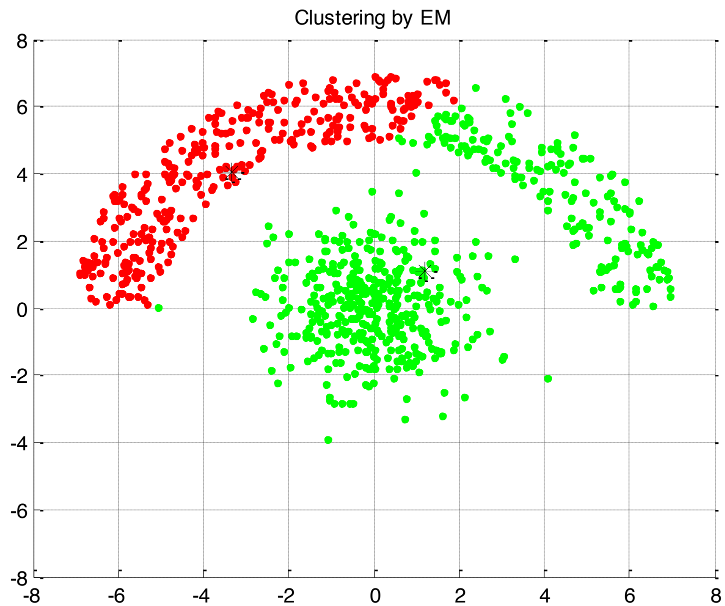




EM for GMM vs. K-means







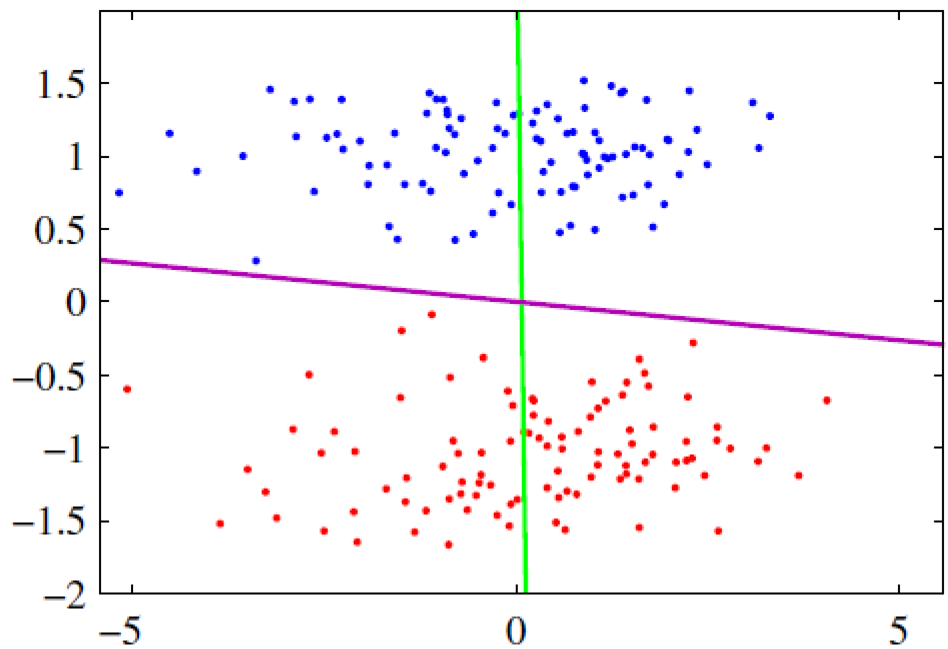
Applications of PCA

- Dimensionality reduction
  - Avoid the curse of dimensionality
- Lossy data compression
- Feature extraction
- Data visualization
  - How to visualize high-dimensional data?
- …

Applications of PCA

- PCA vs. Fisher's LDA:

A comparison of principal component analysis with Fisher's linear discriminant for linear dimensionality reduction. Here the data in two dimensions, belonging to two classes shown in red and blue, is to be projected onto a single dimension. PCA chooses the direction of maximum variance, shown by the magenta curve, which leads to strong class overlap, whereas the Fisher linear discriminant takes account of the class labels and leads to a projection onto the green curve giving much better class separation.



http://www.face-rec.org/algorithms/PCA/jcn.pdf
http://www.cs.jhu.edu/~hager/Public/teaching/cs461/pami97-eigenfaces.pdf

Modeling nonlinear manifolds

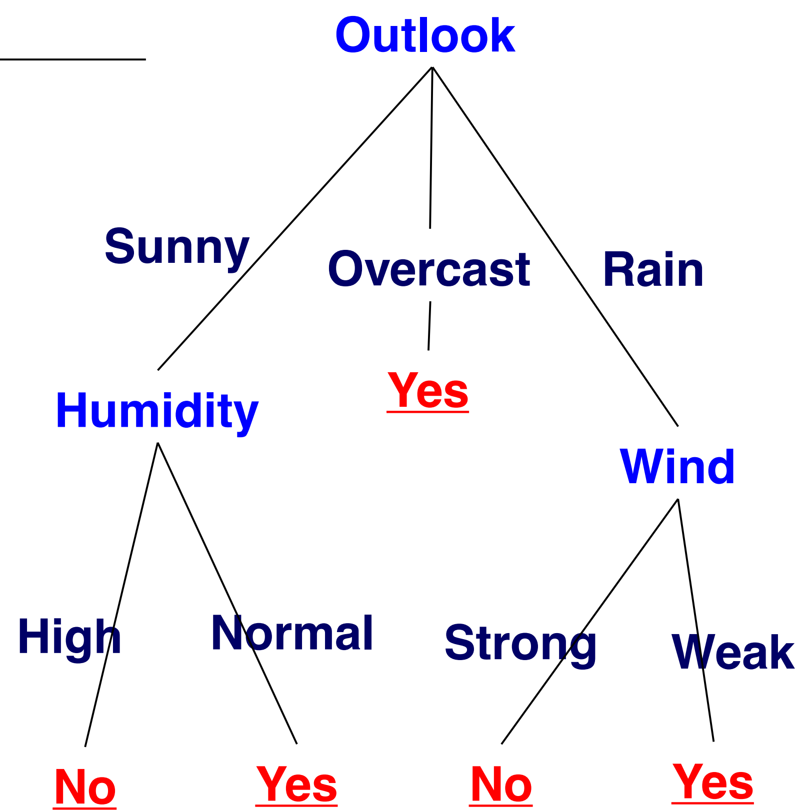
- Two nonprobabilistic methods for dimensionality reduction and data visualization:
  - *Isometric feature mapping (ISOMAP): global method*
    - project the data to a lower-dimensional space using MDS, but where the dissimilarities are defined in terms of the *geodesic distances* measured along the manifold.
  - *Locally linear embedding (LLE):local method*
    - Map the high-dimensional data points down to a lower dimensional space while preserving coefficients.

1. *Tenenbaum J.B., Silva V. De , Langford J. C., A global geometric framework for nonlinear dimensionality reduction, Science, 2000, 290 (5500): 2219-2323*
2. *Sam Roweis,Lawrence Saul,Nonlinear dimensionality reduction by locally linear embedding, Science, 2000,290(5500):2323-2326*

Basic Decision Trees Learning Algorithm

- Data is processed in Batch (i.e., all the data is available).
- Recursively build a decision tree top-down.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



Information Gain

- The information gain of an attribute *a* is the expected reduction in entropy caused by partitioning on this attribute.

$$Gain(S,a) = Entropy(S) - \sum_{v \in values(s)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Where S_v is the subset of S for which attribute *a* has value *v*

and the entropy of partitioning the data is calculated by weighing the entropy of each partition by its size relative to the original set

Partitions of low entropy lead to high gain

Course grade:

40% on homework + 10% on attendance + 50% on final exam

GOOD LUCK!