Linear Classification

Lili MOU

moull12@sei.pku.edu.cn
http://sei.pku.edu.cn/~moull12

23 April 2015

Outline

Outline

Prologue

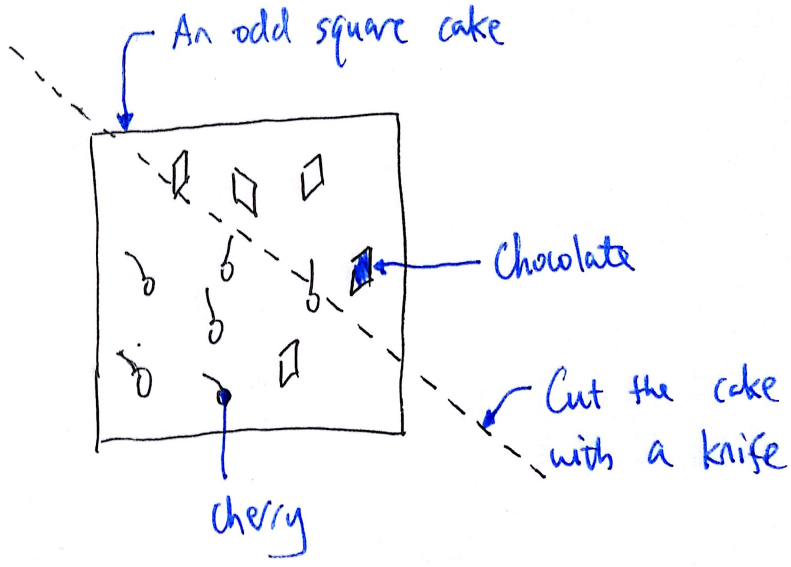
Linear classification is easy—my good old friend, logistic regression, always serves as a baseline method in various applications. Through a systematic study, however, we can grasp the main idea behind a range of machine learning techniques. This seminar also precedes our future discussion on GP classification.

References:

The materials basically follow Chapter 4 in *Pattern Recognition and Machine Learning*. Figures are taken (w/ or w/o modification) without further acknowledgment.

See also A. Webb, et al., *Statistical Pattern Recognition*.

The Linear Classification Problem



The Road Map

- Discriminant functions
- Probabilistic generative models
- Probabilistic discriminative models
- Bayesian logistic regression

On Non-Linearity

Generalized linearity: non-linear transformation by basis functions

- Feature engineering/selection

Kernels: transforming to a Hilbert space, fully characterized by a predefined "inner-product" operation

- SVM (a discriminant function)
- Gaussian processes (non-parametric Bayes)
- k-NN (slacked similarity measure, non-parametric discriminant)

Composition of non-linear activation functions

- Neural networks (finite feature learning, equivalent to learning a sophisticated kernel mapping input data to another Euclidean space)

My questions

- Can neural networks composite infinite dimensional features (with kernels)?
- Sparse kernel network?
- Or is it necessary? Or indeed, neural networks ARE non-parametric!

Outline

The Decision Boundary

Let $\boldsymbol{x} = (x_1, \cdots, x_n)^T$ be a data sample in $\mathbb{R}^n$

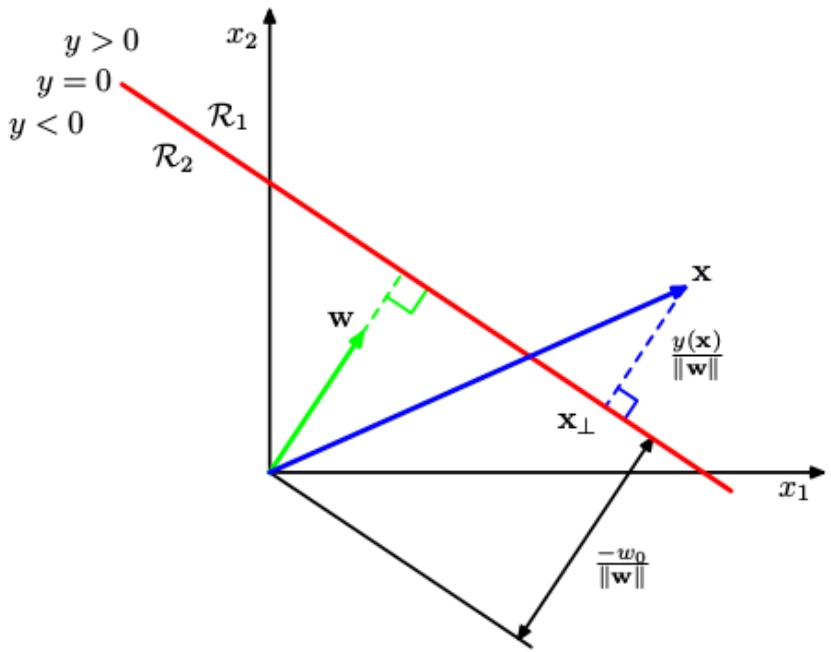
Let $y \in \{0, 1\}$ be the label of $\boldsymbol{x}$ (a two-class problem)

A linear classifier always takes the form

$$y = \begin{cases} 1, & \text{if } \boldsymbol{w}^T\boldsymbol{x} + w_0 > 0 \\ 0, & \text{otherwise} \end{cases}$$

The *decision boundary* is a hyperplane in $\mathbb{R}^{n-1}$

$$\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$$

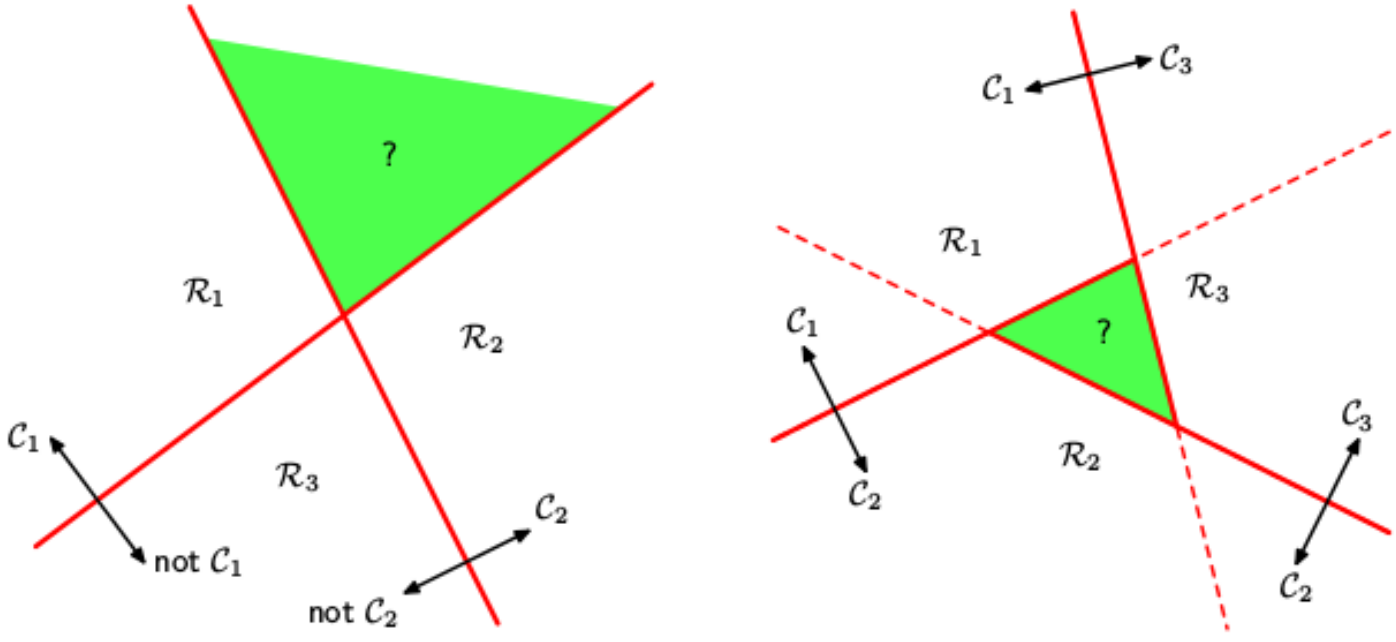


Multiclass problem

Let K be the number of classes

- *one-versus-the-rest* ($K-1$ classifiers)
- *one-versus-one* + voting ($K(K-1)/2$ classifiers)
- Scoring

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$$



Linear Regression as Classiciation

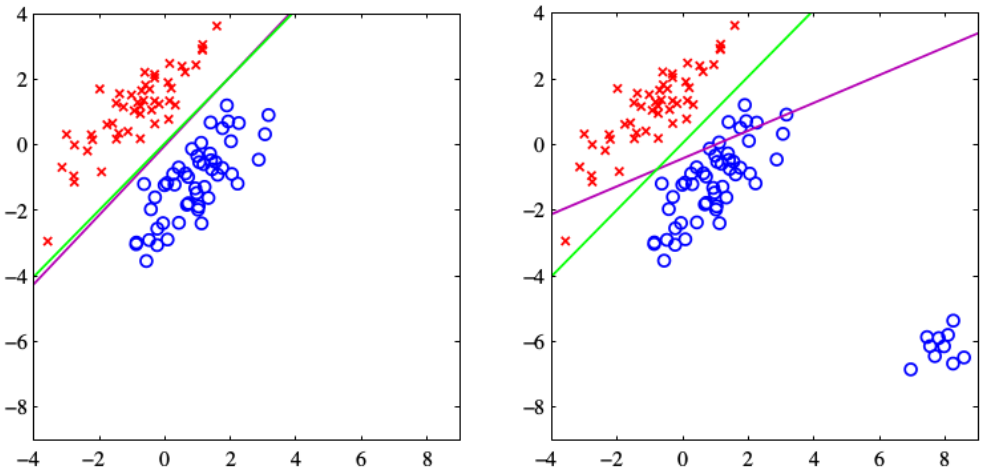
Let the target value be 1-of-K coded. (K: # of classes)
The scoring function is

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$$

The least square objective function

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} \left(y_k(\boldsymbol{x}^{(i)}) - t^{(i)} \right)^2$$

The problem of being "too correct"—the target value is too far away from a Gaussian distribution.



Perception Learning Algorithm

Non-linear hard squashing function

$$y = \begin{cases} 1, & \text{if } \boldsymbol{w}^T \boldsymbol{x} + w_0 > 0 \\ -1, & \text{otherwise} \end{cases}$$

Perception learning algorithm

For each misclassified sample $x^{(i)}$:

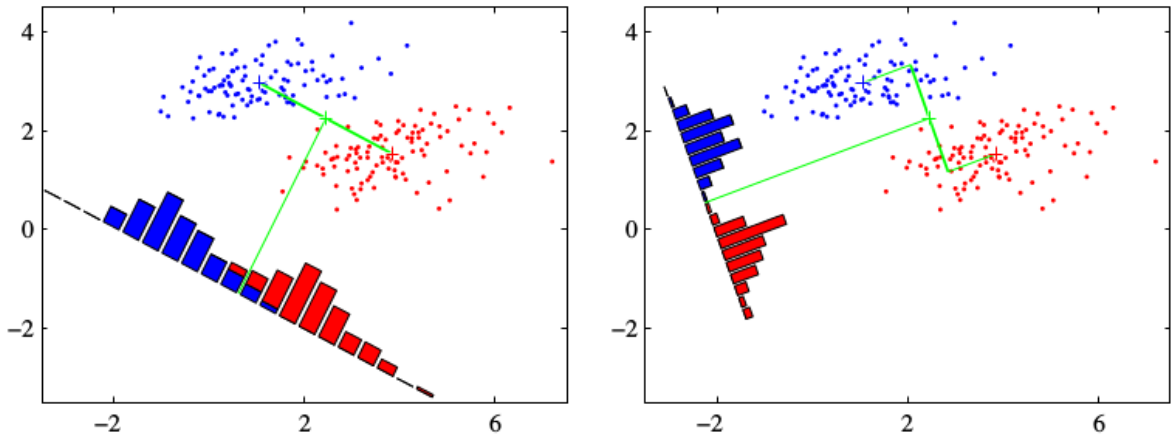
$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} + \eta \boldsymbol{x}^{(i)} t^{(i)}$$

For its convergence theorem, please refer to *Neural Networks and Machine Learning*.

Pros and Cons

- \+ Cares about mis-classified samples only
- – Does not work with linearly inseparable data
- – Does not generalize to multi-class problems
- \+ Introduces non-linear activation function

Fisher's Linear Discriminant



Heuristics:

- Attempt #1: Maximize the class separation
- Attempt #2 (the Fisher criterion):
  Maximize the ratio of the between-class variance to the within-class variance (after projecting to a one-dimensional space, perpendicular to the decision boundary)

Formalization of Fisher's Linear Discriminant

Between-class variance

$$\boldsymbol{m}_1 = \frac{1}{N_1} \sum_{i \in \mathcal{C}_1} \boldsymbol{x}^{(i)}, \qquad \boldsymbol{m}_2 = \frac{1}{N_2} \sum_{i \in \mathcal{C}_2} \boldsymbol{x}^{(i)}$$

Within-class variance

$$s_k^2 = \sum_{i \in \mathcal{C}_k} \left(y^{(i)} - m_k \right)^2$$

where

$$y_n = \boldsymbol{w}^T \boldsymbol{x}_n$$

The cost function

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T (\boldsymbol{m}_2 - \boldsymbol{m}_1)}{s_1^2 + s_2^2}$$

Solution to Fisher's Linear Discriminant

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)}{s_1^2 + s_2^2} = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}}$$

where

$$\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T$$
$$\boldsymbol{S}_W = \sum_{i \in \mathcal{C}_1} (\boldsymbol{x}_i - \boldsymbol{m}_1)(\boldsymbol{x}_i - \boldsymbol{m}_1)^T + \sum_{i \in \mathcal{C}_2} (\boldsymbol{x}_i - \boldsymbol{m}_2)(\boldsymbol{x}_i - \boldsymbol{m}_2)^T$$

Differentiating the cost function and setting it to 0, we obtain

$$(\boldsymbol{w}\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w} = (\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})\boldsymbol{S}_B\boldsymbol{w}$$

Thus

$$\boldsymbol{w} \propto \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

See *Pattern Recognition and Machine Learning* for multi-class scenarios.

Outline

A Probabilistic Model

A data $(\boldsymbol{x}, y)$ is generated according to the following story

Step 1: Choose a box $\mathcal{C}_k$, i.e., $t = k$, with probability $p(\mathcal{C}_k)$
Step 2: Draw a ticket $\boldsymbol{x}$ from box k with probability $p(x|\mathcal{C}_k)$

The Bayesian network

$$(t) \longrightarrow (\boldsymbol{x})$$

Posterior probability

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1+\exp(-a)} \stackrel{\Delta}{=} \sigma(a) \end{aligned}$$

where

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \ln \frac{p(\mathcal{C}_1|\mathbf{x})}{p(\mathcal{C}_2|\mathbf{x})}$$

Multi-class Scenarios

Softmax

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)}$$

Predicting the class label

$$\hat{t} = \underset{k}{\operatorname{argmax}}\, p(\mathcal{C}_k|\mathbf{x})$$

Generative Models v.s. Discriminative Models

Training objectives

$$\underset{\Theta}{\text{maximize}}\, p(\boldsymbol{x}, \boldsymbol{t}; \Theta)$$

i.e.,

$$\underset{\Theta}{\text{maximize}}\, p(\boldsymbol{x}|\boldsymbol{t}; \Theta) p(\boldsymbol{t}; \Theta)$$

v.s.

$$\underset{\Theta}{\text{maximize}}\, p(\boldsymbol{t}|\boldsymbol{x}; \Theta)$$

Gaussian Inputs

Assumption

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

Consider a binary classification problem

$$p(\mathcal{C}_1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

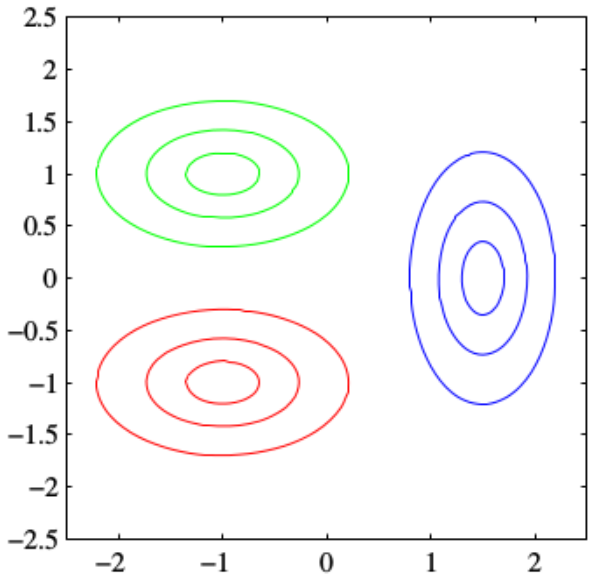
where

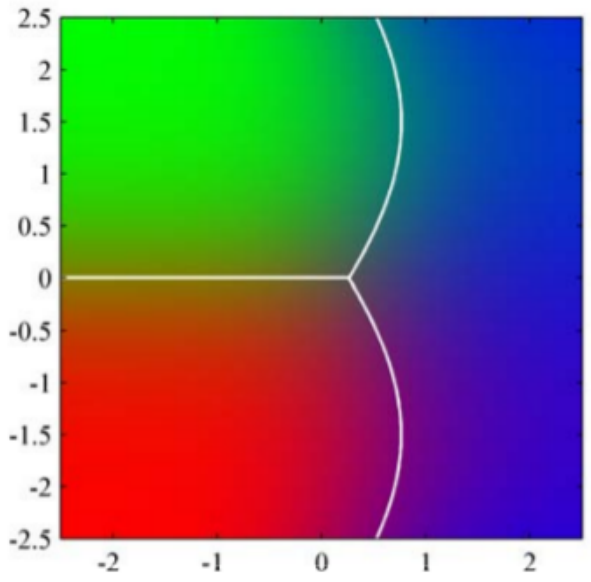
$$\begin{aligned}\boldsymbol{w} &= \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}\end{aligned}$$

Parameter estimation: Maximum likelihood estimation

See also Ch 2.3, 2.4 in *Statistical Pattern Recognition*.

Gaussian Classifiers





Shared covariance matrix $\mathbf{\Sigma} \Rightarrow$ Linear decision boundary

Different $\mathbf{\Sigma}_k \Rightarrow$ Quadratic decision boundary

Warning: Don't use Gaussian Classifiers

Discrete Features

Consider binary features $\boldsymbol{x} = (x_1, \cdots, x_n)^T$, where $x_i \in \{0, 1\}$.

$p(\boldsymbol{x}|\mathcal{C}_k)$ has $2^n - 1$ independent free parameters

Naïve Bayes assumption: x_i independent

$$p(\boldsymbol{x}|\mathcal{C}_k) = \prod_{i=1}^{n} p(x_i|\mathcal{C}_k) = \prod_{i=1}^{n} \mu_{ki}^{x_i}(1 - \mu_{ki})^{1-x_i}$$

Posterior class distributions

$$p(\mathcal{C}_k|\boldsymbol{x}) = \sigma(a_k(\boldsymbol{x}))$$

$$a_k(\boldsymbol{x}) = \sum_{i=1}^{n} \{x_i \ln \mu_{ki} - (1 - x_i) \ln(1 - \mu_{ki})\} + \ln p(\mathcal{C}_k)$$

Linear in features $\boldsymbol{x}$!

Exponential Family

Let class-conditional distributions take the form

$$p(\boldsymbol{x}|\lambda_k) = h(\boldsymbol{x})g(\boldsymbol{\lambda}_k)\exp\left\{\boldsymbol{\lambda}_k^T\boldsymbol{u}(\boldsymbol{x})\right\}$$

Binary classification

$$a(\boldsymbol{x}) = (\boldsymbol{\lambda_1} - \boldsymbol{\lambda_2})^T\boldsymbol{x} + \ln g(\boldsymbol{\lambda}_1) - \ln g(\boldsymbol{\lambda}_2) + \ln p(\mathcal{C}_1) - \ln p(\mathcal{C}_2)$$

Multi-class problem

$$a_k(\boldsymbol{x}) = \boldsymbol{\lambda}_k^T\boldsymbol{x} + \ln g(\boldsymbol{\lambda}_k) + \ln p(\mathcal{C}_k)$$

Examples: Normal, exponential, log-normal, gamma, chi-squared, beta, Dirichlet, Bernoulli, categorical, Poisson, geometric, inverse Gaussian, von Mises, and von Mises-Fisher. Provided some parameters are fixed, Pareto, binomial, multinomial, and negative binomial are also in the exponential family.

Outline

Logistic Regression

Under rather general assumptions, the posterior class distribution takes the form

$$y(\boldsymbol{x}) \stackrel{\Delta}{=} p(\mathcal{C}_1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

The idea is to optimize directly the above posterior distribution.
$\Rightarrow$ Fewer parameters, better performance.

Consider a binary classification problem with n-dimensional feature space

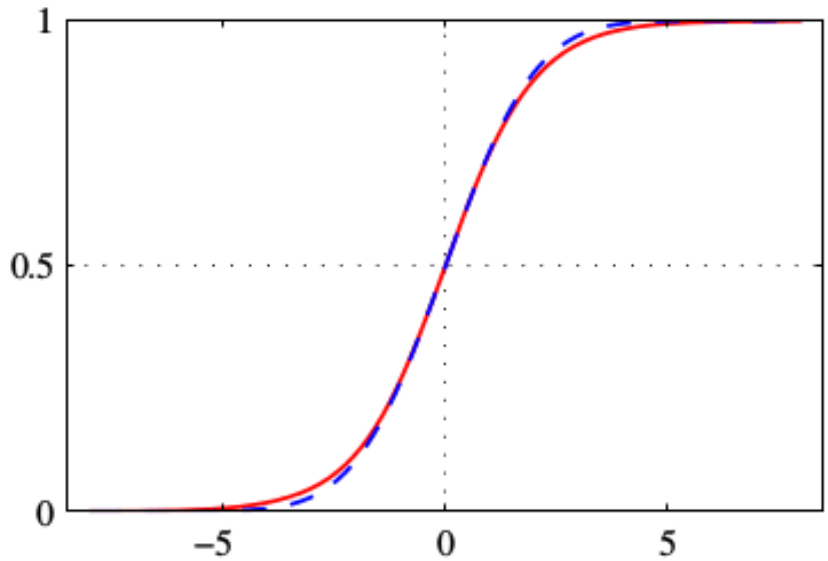
- Logistic regression: n parameters
- A Gaussian classifier:
  - $+$ Means: $2n$
  - $+$ Covariance matrices (shared): $n(n+1)/2$
  - $+$ Class prior: 1

Probit Regression

What if we substitute σ with other squashing functions?

Probit function

$$\Phi(a) = \int_{-\infty}^{a} \mathcal{N}(\theta|0,1)\,\mathrm{d}\theta$$



- Similar performance compared with logistic regression
- More prone to outliers (Why? Consider the decay rate)
- Useful later

Outline

Bayesian Learning

Let $\boldsymbol{y} \in \{0,1\}^m$ denote the labels of training data $\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_m$

Prior $p(\boldsymbol{w})$, which is a \$64,000,000 question

Likelihood

$$p(\boldsymbol{y}|\boldsymbol{w}) = \prod_{i=1}^{m} p(y^{(i)}|\boldsymbol{w}) = \prod_{i=1}^{m} \sigma\left(\boldsymbol{w}^T \boldsymbol{\phi}^{(i)}\right)^{t^{(i)}} \left(1 - \sigma\left(\boldsymbol{w}^T \boldsymbol{\phi}^{(i)}\right)\right)^{1-t^{(i)}}$$

Posterior

$$p(\boldsymbol{w}|\boldsymbol{y}) = \frac{p(\boldsymbol{w})p(\boldsymbol{y}|\boldsymbol{w})}{p(\boldsymbol{y})} \underset{\boldsymbol{w}}{\propto} p(\boldsymbol{w})p(\boldsymbol{y}|\boldsymbol{w}) = p(\boldsymbol{w}) \prod_{i=1}^{m} \sigma(\cdot)^{t^{(i)}} (1-\sigma(\cdot))^{1-t^{(i)}}$$

Predictive density

$$p(y_*|\boldsymbol{y}) = \int \mathrm{d}\boldsymbol{w}\ p(y_*|\boldsymbol{w}) \cdot p(\boldsymbol{w}|\boldsymbol{y})$$

$$\underset{y_*}{\propto} \int \mathrm{d}\boldsymbol{w}\ \sigma\left(\boldsymbol{w}^T \boldsymbol{\phi}_*\right) \cdot p(\boldsymbol{w}) \prod_{i=1}^{m} \sigma(\cdot)^{t^{(i)}} (1 - \sigma(\cdot))^{1-t^{(i)}}$$

Intractability

- Posterior is intractable due to the normalizing factor.
- Predictive density is intractable due to the integral.

We have to resort to approximations

- Sampling methods

  Stochastic, usually asymptotically correct, hard to scale
- Deterministic methods

  "Do things wrongly and hope they work"

Laplace Approximation

- Fit a Gaussian at a mode
- The standard deviation is chosen such that ...
  the second-order derivative of the log probability matches
- ☺ The first-order derivative is always 0 at a mode
- ☺ Scale free in representing the unnormalized measure
- ☹ Real variables only
- ☹ Only local properties captured, multi-mode distributions?

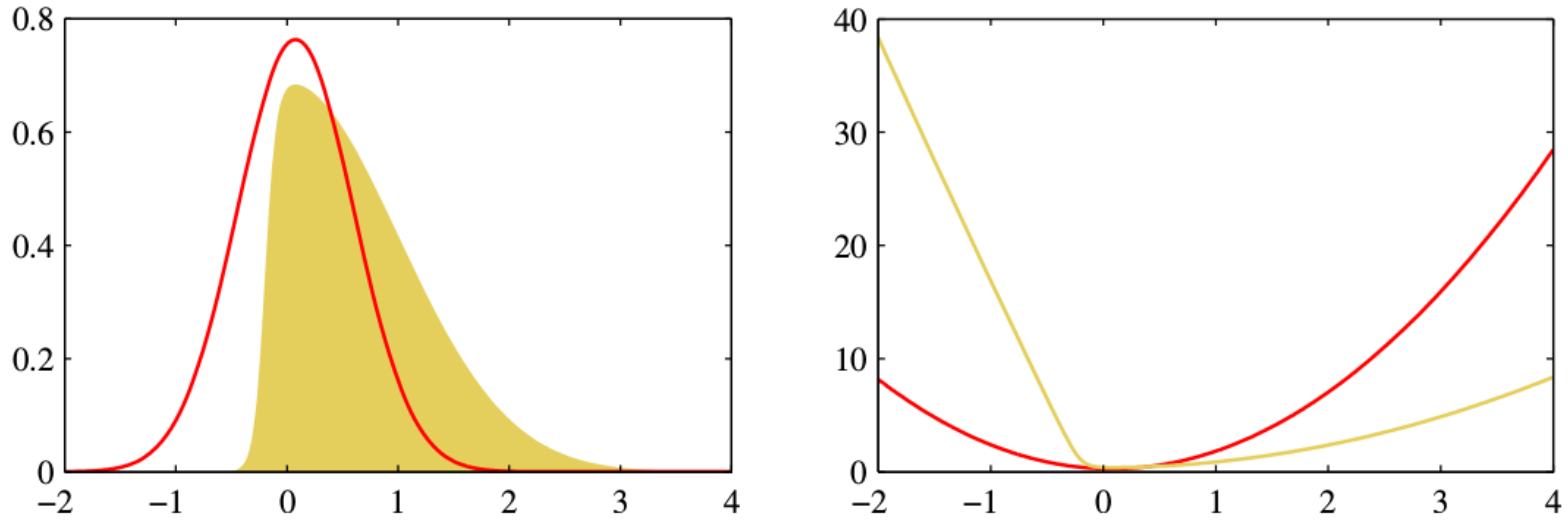


Figure 4.14 Illustration of the Laplace approximation applied to the distribution $p(z) \propto \exp(-z^2/2)\sigma(20z + 4)$ where $\sigma(z)$ is the logistic sigmoid function defined by $\sigma(z) = (1 + e^{-z})^{-1}$. The left plot shows the normalized distribution $p(z)$ in yellow, together with the Laplace approximation centred on the mode z_0 of $p(z)$ in red. The right plot shows the negative logarithms of the corresponding curves.

Fitting a Gaussian

Let $p(z) = \frac{1}{Z} f(z)$ be a true distribution, where $Z = \int f(z)\,\mathrm{d}z$

Step 1: Find a mode z_0 of $p(z)$, by gradient methods, say, satisfying

$$\left.\frac{\mathrm{d}f(z)}{\mathrm{d}z}\right|_{z=z_0} = 0$$

Step 2: Consider a Taylor expansion of $\ln f(z)$ at z_0

$$\ln f(z) \simeq \ln f(z_0) - \frac{1}{2} A (z - z_0)^2$$

where A is given by

$$A = -\left.\frac{\mathrm{d}^2}{\mathrm{d}z^2} \ln f(z)\right|_{z=z_0}$$

Taking the exponential,

$$f(z) \simeq f(z_0) \exp\left\{-\frac{A}{2}(z - z_0)^2\right\}$$

Step 3: Normalize to a Gaussian distribution

$$q(z) = \left(\frac{A}{2\pi}\right)^{1/2} \exp\left\{-\frac{A}{2}(z - z_0)^2\right\}$$

Laplace Approximation for Multivariate Distributions

To approximate $p(\boldsymbol{z}) = \frac{1}{Z} f(\boldsymbol{z})$, where $\boldsymbol{z} \in \mathbb{R}^m$

We expand at mode $\boldsymbol{z}_0$

$$\ln f(\boldsymbol{z}) \simeq \ln f(\boldsymbol{z}_0) - \frac{1}{2}(\boldsymbol{z} - z_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0)$$

where $\boldsymbol{A} = -\nabla\nabla \ln f(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{z}_0}$, Hessian of $\ln f(\boldsymbol{z})$, serving as the *precision matrix* in a Gaussian distribution

Taking the exponential, we obtain

$$f(\boldsymbol{z}) \simeq f(\boldsymbol{z}_0) \exp\left\{-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0)\right\}$$

Normalize it as a distribution, and then we have

$$q(\boldsymbol{z}) = \frac{|\boldsymbol{A}|^{1/2}}{(2\pi)^{m/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0)\right\} = \mathcal{N}(\boldsymbol{z}|\boldsymbol{z}_0, \boldsymbol{A}^{-1})$$

Bayesian Logistic Regression: A Revisit in Earnest

Prior: Gaussian, which is natural[1]

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_0, \boldsymbol{S}_0)$$

Posterior:
$$p(\boldsymbol{w}|\boldsymbol{t}) \propto p(\boldsymbol{w})p(\boldsymbol{t}|\boldsymbol{w})$$

$$\ln p(\boldsymbol{w}|\boldsymbol{t}) = -\frac{1}{2}(\boldsymbol{w}-\boldsymbol{m}_0)^T\boldsymbol{S}_0^{-1}(\boldsymbol{w}-\boldsymbol{m}_0) \quad \text{[prior]}$$

$$+\sum_{i=1}^{m}\left\{t^{(i)}\ln y^{(i)} + \left(1-t^{(i)}\right)\ln\left(1-y^{(i)}\right)\right\} \quad \text{[likelihood]}$$

$$+ \text{const}$$

$$\boldsymbol{S}_N = -\nabla\nabla \ln p(\boldsymbol{w}|\boldsymbol{t}) = \boldsymbol{S}_0^{-1} + \sum_{i=1}^{m} y^{(i)}(1-y^{(i)})\phi_n\phi_n^T$$

Hence, the Laplace approximation to the posterior is

$$q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{\text{MAP}}, \boldsymbol{S}_N)$$

---

[1] Mathematicians always choose priors for the sake of convenience rather than approaching God.

Predictive Density

$$p(\mathcal{C}_1|\boldsymbol{\phi}_*, \boldsymbol{t}) = \int p(\mathcal{C}_1|\boldsymbol{\phi}_*, \boldsymbol{w}) p(\boldsymbol{w}|\boldsymbol{t}) \, \mathrm{d}\boldsymbol{w} \simeq \int \sigma(\boldsymbol{w}^T \boldsymbol{\phi}_*) q(\boldsymbol{w}) \, \mathrm{d}\boldsymbol{w}$$

$$p(\mathcal{C}_2|\boldsymbol{\phi}_*, \boldsymbol{t}) = 1 - p(\mathcal{C}_1|\boldsymbol{\phi}_*, \boldsymbol{t})$$

Plan

- Change it to a univariate integral
- Substitute sigmoid with a probit function, which is then convolved with a normal

$$\int \sigma(\cdot) \mathcal{N}(\cdot) \, \mathrm{d}\cdot \simeq \int \Phi(\cdot) \mathcal{N}(\cdot) \, \mathrm{d}\cdot = \Phi(\cdot) \simeq \sigma(\cdot)$$

The Dirac Delta Fucntion

Let δ be the *Dirac delta* function, loosely thought of a function such that

- Gaussian distribution peaked at 0 with standard deviation $\to 0$
- $\delta(x) = \begin{cases} +\infty, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases}$

δ function satisfies

$$\int \delta(a - x) f(a) \, \mathrm{d}a = f(x)$$

and specifically

$$\int \delta(a - x) \, \mathrm{d}a = 1$$

Deriving the Predictive Density

$$p(\mathcal{C}_1|\boldsymbol{\phi}_*) \simeq \int \sigma(\boldsymbol{w}^T\boldsymbol{\phi}_*)q(\boldsymbol{w})\,\mathrm{d}\boldsymbol{w} \qquad \text{[Laplace approx.]}$$

$$\sigma(\boldsymbol{w}^T\boldsymbol{\phi}) = \int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi}_*)\sigma(a)\,\mathrm{d}a \qquad \text{[Def. of } \delta\text{]}$$

$$\begin{aligned} p(\mathcal{C}_1|\boldsymbol{\phi}_*) &= \int\int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi})\sigma(a)\,\mathrm{d}a\, q(\boldsymbol{w})\,\mathrm{d}\boldsymbol{w} \\ &= \int\int \sigma(a)\delta(a - \boldsymbol{w}^T\boldsymbol{\phi}_*)q(\boldsymbol{w})\,\mathrm{d}\boldsymbol{w}\,\mathrm{d}a \\ &\triangleq \int \sigma(a)p(a)\,\mathrm{d}a \end{aligned}$$

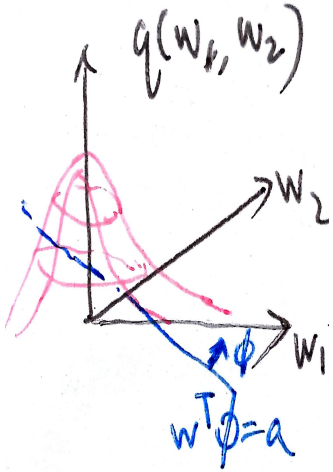
where

$$p(a) \triangleq \int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi}_*)q(\boldsymbol{w})\,\mathrm{d}\boldsymbol{w}$$

We now argue that $p(a)$ is Gaussian

Deriving the Predictive Density (2)

$$p(a) = \int\int\cdots\int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi}_*)q(\boldsymbol{w})\,\mathrm{d}w_1\,\mathrm{d}w_2\cdots\mathrm{d}w_n$$
$$= \int\int\cdots\int q(\tilde{\boldsymbol{w}})\,\mathrm{d}w_2\cdots\mathrm{d}w_n$$



$\tilde{\boldsymbol{w}}$ is such that $\tilde{\boldsymbol{w}}^T\boldsymbol{\phi}_* = a$

We can also verify that

$$\int p(a)\,\mathrm{d}a = \int\int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi}_*)q(\boldsymbol{w})\,\mathrm{d}\boldsymbol{w}\,\mathrm{d}a$$
$$= \int\int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi}_*)q(\boldsymbol{w})\,\mathrm{d}a\,\mathrm{d}\boldsymbol{w}$$
$$= \int q(\boldsymbol{w})\,\mathrm{d}\,\boldsymbol{w}\int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi})\,\mathrm{d}a$$
$$= 1$$

Deriving the Predictive Density (3)

$$\mu_a = \mathbb{E}[a] = \int p(a)a \, \mathrm{d}a = \int q(\boldsymbol{w})\boldsymbol{w}^T\boldsymbol{\phi}_* \, \mathrm{d}\boldsymbol{w} = \boldsymbol{w}_{\mathsf{MAP}}^T\boldsymbol{\phi}_*$$

$$\begin{aligned}\sigma_a^2 = \mathsf{var}[a] &= \int p(a)\left\{a^2 - \mathbb{E}[a]^2\right\} \mathrm{d}a \\ &= \int q(\boldsymbol{w})\left\{(\boldsymbol{w}^T\boldsymbol{\phi})^2 - (\boldsymbol{m}_N^2\boldsymbol{\phi})^2\right\} \mathrm{d}\boldsymbol{w} \\ &= \phi^T\boldsymbol{S}_N\boldsymbol{\phi}\end{aligned}$$

Thus

$$\begin{aligned}p(\mathcal{C}_1|\boldsymbol{t}) &\simeq \int \sigma(a)p(a) \, \mathrm{d}a = \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2) \, \mathrm{d}a \\ &\simeq \int \Phi(\lambda a)\mathcal{N}(a|\mu_a, \sigma_a^2) \, \mathrm{d}a = \Phi\left(\frac{\mu_a}{(\lambda^{-2} + \sigma_a^2)^{1/2}}\right) \simeq \sigma\left(\kappa(\sigma_a^2)\mu_a\right)\end{aligned}$$

$\lambda = \sqrt{\pi/8}$, $\kappa(\sigma_a^2) = (1 + \pi\sigma_a^2/8)^{-1/2}$, chosen such that the rescaled probit function has the same slope as sigmoid at the origin.

$\Phi(\cdot) * \phi(\cdot) = \Phi(\cdot)$

Let $X \sim \mathcal{N}(a, b^2)$ and $Y \sim \mathcal{N}(c, d^2)$

$$\begin{aligned}
\Pr\{X \leq Y\} &= \int_{-\infty}^{\infty} \Pr\{X \leq Y | Y = w\} \, \phi\left(\frac{w-c}{d}\right) \mathrm{d}w \\
&= \int_{\infty}^{\infty} \Pr\{X \leq w\} \, \phi\left(\frac{w-c}{d}\right) \mathrm{d}w \\
&= \int_{\infty}^{\infty} \Phi\left(\frac{w-a}{c}\right) \phi\left(\frac{w-c}{d}\right) \mathrm{d}w
\end{aligned}$$

By noticing that

$$X - Y \sim \mathcal{N}(a - c, b^2 + d^2)$$

We have

$$\begin{aligned}
\Pr\{X \leq Y\} &= \Pr\{X - Y \leq 0\} \\
&= \Phi\left(\frac{-a+c}{\sqrt{b^2+d^2}}\right)
\end{aligned}$$

Take-Home Messages

- Discriminant functions
- Probabilistic generative models (don't use it)
- Probabilistic discriminative models
- Bayesian logistic regression
  - ✂ Sampling methods
  - ☞ Deterministic approximations
    - Laplace approximation (fitting a Gaussian at a mode)
    - Substitute the sigmoid function with a probit function
    - Analytical solutions
    - The decision boundary is the same with equal prior probabilities

Thanks for Listening!