Learning as Inference

DD2421

Giampiero Salvi

HT2018

Outline

Probabilistic Classification and Regression

- In both cases estimate posterior

$$P(y \mid x) = \frac{P(x \mid y)P(y)}{P(x)}$$

- Classification: y is discrete
- Regression: y is continuous

Until now we assumed we knew:

- $P(y) \leftarrow$ *Prior*
- $P(x \mid y) \leftarrow$ *Likelihood*
- $P(x) \leftarrow$ *Evidence*

How can we obtain this information from observations (data)?

Learning as Inference

Given:

- the training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$
- a new observation $\mathbf{x}$

Estimate the posterior probability of the answer y:

$$P(y|\mathbf{x}, \mathcal{D})$$

Discriminative vs Generative Models

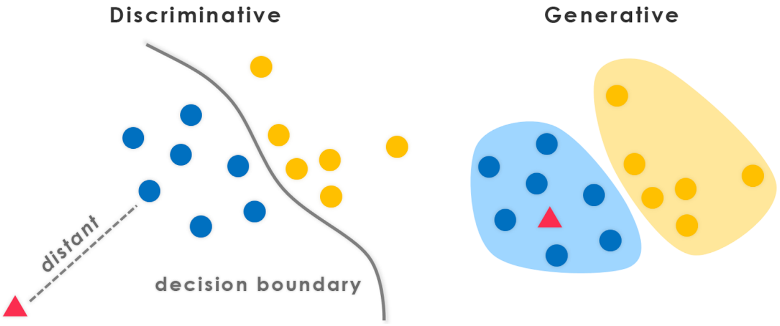


Figure from Nguyen *et al.* 2015. `http://www.evolvingai.org/fooling`

Discriminative vs Generative Models

Discriminative:

- learn the posterior $P(y|\mathbf{x}, \mathcal{D})$ directly
- examples: linear regression, logistic regression

Generative:

- learn a model of data generation: priors $P(y|\mathcal{D})$ and likelihoods $P(\mathbf{x}|y, \mathcal{D})$
- use Bayes rule to obtain posterior $P(y|\mathbf{x}, \mathcal{D})$
- example: classification

Parametric vs Non-parametric Inference

Parametric:

- First make the model parameters explicit: $P(y|\mathbf{x}) = P(y|\mathbf{x}, \theta)$
- estimate the optimal parameters $\hat{\theta}$ using the data (point estimate)
- compute the posterior $P(y|\mathbf{x}, \hat{\theta})$

Learning corresponds to finding $\hat{\theta}$

Non-Parametric:

- Use a parametric model as before: $P(y|\mathbf{x}) = P(y|\mathbf{x}, \theta)$
- but estimate the posterior of the parameters given the data: $P(\theta|\mathcal{D})$
- Compute the posterior $P(y|\mathbf{x}, \mathcal{D})$ by marginalizing out the parameters θ

The number of parameters can grow with the data!

Three Approaches

Parametric:

- Maximum Likelihood (ML)
- Maximum A Posteriori (MAP)

Non-parametric:

- Bayesian methods

Fundamental Assumption: i.i.d.

Samples from each class are independent and identically distributed:

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

The likelihood of the whole data set can be factorized:

$$P(\mathcal{D}) = P(\mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{i=1}^{N} P(\mathbf{x}_i)$$

And the log-likelihood becomes:

$$\log P(\mathcal{D}) = \sum_{i=1}^{N} \log P(\mathbf{x}_i)$$

Maximum Likelihood Estimate

- define parametric form for the distributions:

$$P(\mathbf{x}|y) \equiv P(\mathbf{x}|y, \theta) \quad \text{or} \quad P(y|\mathbf{x}) \equiv P(y|\mathbf{x}, \theta)$$

- find optimal value for the parameter θ_{ML} by maximizing the likelihood of the data:

$$\theta_{\mathsf{ML}} = \arg\max_{\theta} P(\mathcal{D}|\theta)$$

- approximate the distribution given the data with this distribution:

$$P(\mathbf{x}|y, \mathcal{D}) \approx P(\mathbf{x}|y, \theta_{\mathsf{ML}}) \quad \text{or} \quad P(y|\mathbf{x}, \mathcal{D}) \approx P(y|\mathbf{x}, \theta_{\mathsf{ML}})$$

Probabilistic Linear Regression

Model (deterministic):

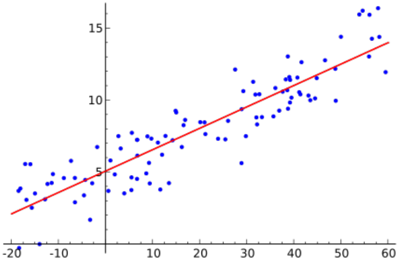
$$y = \mathbf{w}^T \mathbf{x} + \epsilon$$

But now:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Therefore:

$$\begin{aligned} y &\sim \mathcal{N}(\mu_Y(\mathbf{x}), \sigma_Y^2(\mathbf{x})) \\ &= \mathcal{N}(\mathbf{w}^T \mathbf{x}, \sigma^2) \end{aligned}$$



Learning: find $\mathbf{w}$ that maximizes $P(Y|X, \mathbf{w}, \sigma^2)$

Maximize the posterior directly $\implies$ discriminative method

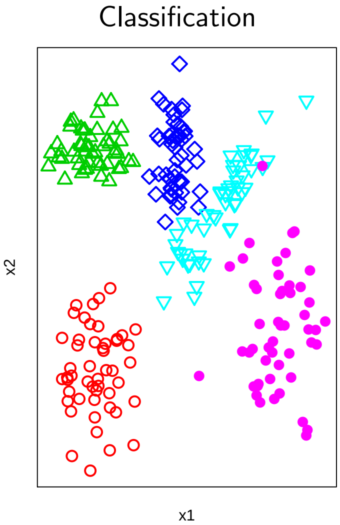
MLE for Probabilistic Linear Regression

$$\begin{aligned}
\log P(Y|X, \mathbf{w}, \sigma^2) &= \log \prod_i P(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log P(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_i \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \mathbf{w}^T\mathbf{x}_i)^2}{2\sigma^2}} \right] \\
&= \sum_i \left[-\frac{1}{2} \log(2\pi\sigma^2) - \frac{(y_i - \mathbf{w}^T\mathbf{x}_i)^2}{2\sigma^2} \right]
\end{aligned}$$

$$\arg\max_{\mathbf{w}} \left[P(Y|X, \mathbf{w}, \sigma^2) \right] = \arg\min_{\mathbf{w}} \sum_i (y_i - \mathbf{w}^T\mathbf{x}_i)^2$$

Maximizing $P(Y|X, \mathbf{w}, \sigma^2)$ equivalent to minimizing sum of squares!

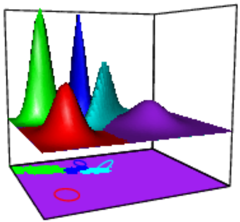
MLE for Classification



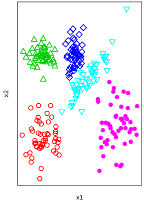
features: $\mathbf{x} \in \mathbb{R}^d$

class: $y \in \{y_1, \dots, y_K\}$

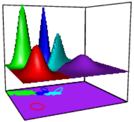
$$k_{\text{MAP}} = \arg\max_k P(y_k|\mathbf{x})$$
$$= \arg\max_k P(y_k)P(\mathbf{x}|y_k)$$



Assumption: Class Independence

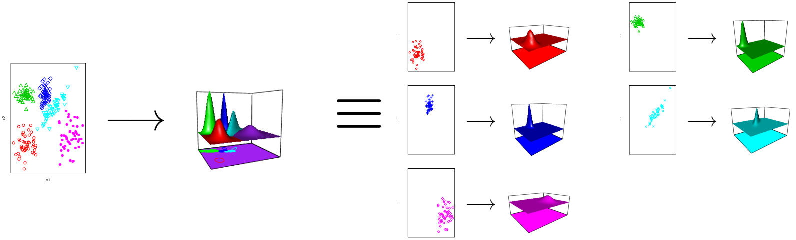


$\longrightarrow$



samples from class i do not influence estimate for class j, $i \neq j$

Assumption: Class Independence



- each distribution for class y_j is a likelihood in the form $P(\mathbf{x}|\theta_i)$
- in the following we drop the class index i and write $P(\mathbf{x}|\theta)$
- also we call $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ the set of data point belonging to a single class y_i

ML estimation of Gaussian mean

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \text{ with } \theta = \{\mu, \sigma^2\}$$

Log-likelihood of data (i.i.d. samples):

$$\log P(\mathcal{D}|\theta) = \sum_{i=1}^{N} \log \mathcal{N}(x_i|\mu,\sigma^2) = -N \log\left(\sqrt{2\pi\sigma^2}\right) - \sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma^2}$$

$$0 = \frac{d \log P(\mathcal{D}|\theta)}{d\mu} = \sum_{i=1}^{N} \frac{(x_i-\mu)}{\sigma^2} = \frac{\sum_{i=1}^{N} x_i - N\mu}{\sigma^2} \iff$$

$$\mu_{\text{ML}} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

ML estimation of Gaussian parameters

$$\mu_{\mathsf{ML}} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma^2_{\mathsf{ML}} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu_{\mathsf{ML}})^2$$

- same result by minimizing the sum of square errors!
- but we make assumptions explicit

MLE with Discrete Variables

Will I play tennis dependent on the weather?

$$x \in \{\text{sunny}, \text{overcast}, \text{rainy}\}$$
$$y \in \{\text{yes}, \text{no}\}$$

$$x \sim \text{Cat}(\lambda_1, \dots, \lambda_k)$$
$$y \sim \text{Bernoulli}(\alpha)$$
$$x|y \sim \text{Cat}(\lambda'_1, \dots, \lambda'_k)$$
$$y|x \sim \text{Bernoulli}(\alpha')$$

Training data

i example	x_i outlook	y_i play	i example	x_i outlook	y_i play
1	sunny	no	8	sunny	no
2	sunny	no	9	sunny	yes
3	overcast	yes	10	rainy	yes
4	rainy	yes	11	sunny	yes
5	rainy	yes	12	overcast	yes
6	rainy	no	13	overcast	yes
7	overcast	yes	14	rainy	no

MLE: Bernoulli

$$P(y) = \begin{cases} \alpha & \text{if } y = \text{yes} \\ 1 - \alpha & \text{if } y = \text{no} \end{cases}$$

1. compute (log) likelihood of the data $P(\mathcal{D}|\alpha)$
2. find α_{ML} that optimizes $P(\mathcal{D}|\alpha)$

i example	x_i outlook	y_i play	i example	x_i outlook	y_i play
1	sunny	no	8	sunny	no
2	sunny	no	9	sunny	yes
3	overcast	yes	10	rainy	yes
4	rainy	yes	11	sunny	yes
5	rainy	yes	12	overcast	yes
6	rainy	no	13	overcast	yes
7	overcast	yes	14	rainy	no

MLE: Bernoulli

$$p(y) = \begin{cases} \alpha & \text{if } y = \text{yes} \\ 1-\alpha & \text{if } y = \text{no} \end{cases}$$

Likelihood of the data
(n=number of yes in $\mathcal{D}$, N=number of examples):

$$\begin{aligned} P(\mathcal{D}|\alpha) &= \prod_i P(y_i|\alpha) = \prod_{i \text{ s.t. } y=\text{yes}} \alpha \prod_{i \text{ s.t. } y=\text{no}} (1-\alpha) \\ &= \alpha^n (1-\alpha)^{N-n} \\ \log P(\mathcal{D}|\alpha) &= n \log \alpha + (N-n)\log(1-\alpha) \\ \frac{d}{d\alpha} \log P(\mathcal{D}|\alpha) &= \frac{n - N\alpha}{\alpha(1-\alpha)} = 0 \iff \alpha_{\text{ML}} = \frac{n}{N} \end{aligned}$$

MLE Example: Discrete Variables

Will I play tennis dependent on the weather?

$$x \in \{\text{sunny}, \text{overcast}, \text{rainy}\}$$
$$y \in \{\text{yes}, \text{no}\}$$

Training data

i example	x_i outlook	y_i play	i example	x_i outlook	y_i play
1	sunny	no	8	sunny	no
2	sunny	no	9	sunny	yes
3	overcast	yes	10	rainy	yes
4	rainy	yes	11	sunny	yes
5	rainy	yes	12	overcast	yes
6	rainy	no	13	overcast	yes
7	overcast	yes	14	rainy	no

$$y \sim \text{Bernoulli}(\alpha)$$
$$\alpha_{\text{ML}} = \frac{9}{14}$$

MLE: Categorical

Similar derivation:

$$\lambda_{k,\text{ML}} = \frac{n_k}{N}$$

where n_k is the number of examples of the kth category

$$\begin{aligned}
x &\sim \text{Cat}(\lambda_{\text{sunny}}, \lambda_{\text{overcast}}, \lambda_{\text{rainy}}) \\
\lambda_{\text{ML}} &= \{\frac{5}{14}, \frac{4}{14}, \frac{5}{14}\} \\
x|y &\sim \text{Cat}(\lambda'_1, \dots, \lambda'_k) \\
\lambda'_{\text{ML}}(\text{yes}) &= \{\frac{2}{9}, \frac{4}{9}, \frac{3}{9}\} \\
\lambda'_{\text{ML}}(\text{no}) &= \{\frac{3}{5}, 0, \frac{2}{5}\}
\end{aligned}$$

Training data

i example	x_i outlook	y_i play	i example	x_i outlook	y_i play
1	sunny	no	8	sunny	no
2	sunny	no	9	sunny	yes
3	overcast	yes	10	rainy	yes
4	rainy	yes	11	sunny	yes
5	rainy	yes	12	overcast	yes
6	rainy	no	13	overcast	yes
7	overcast	yes	14	rainy	no

But. . . , will I play tennis?

Let's say it is rainy:

$$P(y = \text{yes}|\text{outlook=rainy}) = \frac{P(\text{outlook=rainy}|y = \text{yes})P(y = \text{yes})}{P(\text{outlook=rainy})} = \frac{\frac{3}{9}\frac{9}{14}}{\frac{5}{14}} = \frac{3}{5}$$

$$P(y = \text{no}|\text{outlook=rainy}) = \frac{P(\text{outlook=rainy}|y = \text{no})P(y = \text{no})}{P(\text{outlook=rainy})} = \frac{\frac{2}{5}\frac{5}{14}}{\frac{5}{14}} = \frac{2}{5}$$

Then

$$y_{\text{MAP}} = \arg\max_y P(y|\text{outlook=rainy}) = \text{yes}$$

$$y_{\text{ML}} = \arg\max_y P(\text{outlook=rainy}|y) = \text{no}$$

Source of confusion

We did Maximum a Posteriori (MAP) and Maximum Likelihood (ML) classification

$$\begin{aligned} y_{\mathsf{MAP}} &= \arg\max_y P(y|x, \theta_{\mathsf{ML}}) \\ y_{\mathsf{ML}} &= \arg\max_y P(x|y, \theta_{\mathsf{ML}}) \end{aligned}$$

with parameters θ estimated by Maximum Likelihood (ML):

$$\theta_{\mathsf{ML}} = \arg\max_\theta P(D|y, \theta) = \arg\max_\theta \prod_i P(x_i|y_i, \theta)$$

Problem: Curse of Dimensionality

i	$\mathbf{x}_i$				y_i
example	outlook	temperature	humidity	windy	play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

difficult to model $P(\text{outlook}, \text{temperature}, \text{humidity}, \text{windy}|\text{play})$

Problem: Curse of Dimensionality

- Size of feature space exponential in number of features.
- More features $\implies$ potential for better description of the objects but...
- More features $\implies$ more difficult to model $P(\mathbf{x} \mid y)$.

Approximation: **Naïve Bayes classifier**

- All features (dimensions) regarded as conditionally independent.
- Instead of modelling one D-dimensional distribution:
  P(outlook, temperature, humidity, windy|play)
  model D one-dimensional distributions:
  P(outlook|play), P(temperature|play),
  P(humidity|play), P(windy|play)

Naïve Bayes Classifier

- $\mathbf{x}$ is a vector $(x_1, \ldots, x_D)$ of attribute or feature values.
- Let $\mathcal{Y} = \{1, 2, \ldots, Y\}$ be the set of possible classes.
- The MAP estimate of y is

$$\begin{aligned} y_{\text{MAP}} &= \arg\max_{y \in \mathcal{Y}} P(y \mid x_1, \ldots, x_D) = \arg\max_{y \in \mathcal{Y}} \frac{P(x_1, \ldots, x_D \mid y)\, P(y)}{P(x_1, \ldots, x_D)} \\ &= \arg\max_{y \in \mathcal{Y}} P(x_1, \ldots, x_D \mid y)\, P(y) \end{aligned}$$

- Naïve Bayes assumption: $P(x_1, \ldots, x_D \mid y) = \prod_{d=1}^{D} P(x_d \mid y)$
- Naïve Bayes classifier:

$$y_{\text{MAP}} = \arg\max_{y \in \mathcal{Y}} P(y) \prod_{d=1}^{D} P(x_d \mid y)$$

Naïve Bayes Classifier

- One of the most common learning methods.

When to use:

- Moderate or large training set available.
- Features x_i of a data instance $\mathbf{x}$ are conditionally independent given classification (or at least reasonably independent, still works with a little dependence).

Successful applications:

- Medical diagnoses (symptoms independent)
- Classification of text documents (words independent)
- Acoustic modelling in Automatic Speech Recognition

Example: Play Tennis?

Question: Will I go and play tennis given the forecast?

My measurements:

- **outlook** $\in$ {sunny, overcast, rainy},
- **temperature** $\in$ {hot, mild, cool},
- **humidity** $\in$ {high, normal},
- **windy** $\in$ {false, true}.

Possible decisions: $y \in$ {yes, no}

Example: Play Tennis?

What I did in the past:

i example	outlook	$\mathbf{x}_i$ temperature	humidity	windy	y_i play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

Example: Play Tennis?

Counts of when I played tennis (did not play)

Outlook			Temperature			Humidity		Windy	
sunny	overcast	rain	hot	mild	cool	high	normal	false	true
2 (3)	4 (0)	3 (2)	2 (2)	4 (2)	3 (1)	3 (4)	6 (1)	6 (2)	3 (3)

Prior of whether I played tennis or not

Counts:

Play	
yes	no
9	5

Prior Probabilities:

Play	
yes	no
$\frac{9}{14}$	$\frac{5}{14}$

Likelihood of attribute when tennis played $P(x_i \,|\, \text{y=yes})(P(x_i \,|\, \text{y=no}))$

Outlook			Temperature			Humidity		Windy	
sunny	overcast	rain	hot	mild	cool	high	normal	false	true
$\frac{2}{9}$ ($\frac{3}{5}$)	$\frac{4}{9}$ ($\frac{0}{5}$)	$\frac{3}{9}$ ($\frac{2}{5}$)	$\frac{2}{9}$ ($\frac{2}{5}$)	$\frac{4}{9}$ ($\frac{2}{5}$)	$\frac{3}{9}$ ($\frac{1}{5}$)	$\frac{3}{9}$ ($\frac{4}{5}$)	$\frac{6}{9}$ ($\frac{1}{5}$)	$\frac{6}{9}$ ($\frac{2}{5}$)	$\frac{3}{9}$ ($\frac{3}{5}$)

Example: Play Tennis?

Inference: Use the learnt model to classify a new instance.

New instance:

$$\mathbf{x} = (\text{sunny, cool, high, true})$$

Apply Naïve Bayes Classifier:

$$y_{\text{MAP}} = \arg \max_{y \,\in\, \{\text{yes, no}\}} P(y) \prod_{i=1}^{4} P(x_i \,|\, y)$$

$$P(\text{yes})\, P(\text{sunny} \,|\, \text{yes})\, P(\text{cool} \,|\, \text{yes})\, P(\text{high} \,|\, \text{yes})\, P(\text{true} \,|\, \text{yes}) = \frac{9}{14} \times \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} = .005$$

$$P(\text{no})\, P(\text{sunny} \,|\, \text{no})\, P(\text{cool} \,|\, \text{no})\, P(\text{high} \,|\, \text{no})\, P(\text{true} \,|\, \text{no}) = \frac{5}{14} \times \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} = .021$$

$$\Longrightarrow \quad y_{\text{MAP}} = \text{no}$$

Naïve Bayes: Independence Violation

- Conditional independence assumption:

$$P(x_1, x_2, \ldots, x_D \,|\, y) = \prod_{d=1}^{D} P(x_d \,|\, y)$$

  often violated - but it works surprisingly well anyway!
- **Note:** Do not need the posterior probabilities $P(y \,|\, \mathbf{x})$ to be correct. Only need y_{MAP} to be correct.
- Since dependencies ignored, naïve Bayes posteriors often unrealistically close to 0 or 1.

  Different attributes say the same thing to a higher degree than we expect as they are correlated in reality.

Naïve Bayes: Estimating Probabilities

- **Problem:** What if none of the training instances with target value y have attribute x_i? Then

$$P(x_i \,|\, y) = 0 \quad \Longrightarrow \quad P(y) \prod_{i=1}^{D} P(x_i \,|\, y) = 0$$

- **Simple solution:** add pseudocounts to all counts so that no count is zero
- This is a form of regularization or smoothing

Logistic Regression

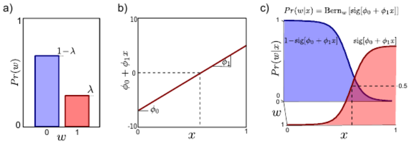


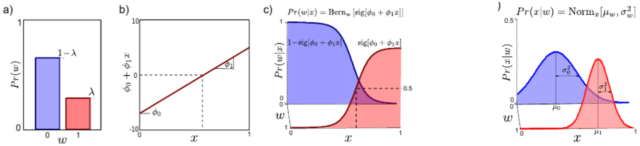
Figure from Prince

- binary classification problem: $y \in \{0, 1\}$
- treat as regression problem: $\mathbf{x} \rightarrow \lambda$ (Bernoulli parameter)

$$
\begin{aligned}
y &\sim \text{Bernoulli}(\lambda) = \lambda^y (1-\lambda)^{(1-y)} \\
\lambda &= \lambda(\mathbf{x}) = \text{sig}(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}
\end{aligned}
$$

$$
y|\mathbf{x} \sim = \lambda(\mathbf{x})^y (1 - \lambda(\mathbf{x}))^{(1-y)}
$$

Logistic Regression vs Gaussian Classifier



Figures from Prince

Same posterior $P(y|\mathbf{x})$ iff:

- equal prior distributions
- shared covariance matrix

Different learning:

- Gaussians: generative model, optimize $P(\mathbf{x}|y_0)$ and $P(\mathbf{x}|y_1)$
- Logistic Regression: discriminative model, optimize $P(y_1|\mathbf{x})$

Logistic Regression: MLE

Learning: maximize $P(Y|\mathbf{X})$ (discriminative method)

$$\begin{aligned}
P(Y|\mathbf{X},\mathbf{w}) &= \prod_{i=1}^{N} \lambda(\mathbf{x}_i)^{y_i}(1-\lambda(\mathbf{x}_i))^{(1-y_i)} \quad \Rightarrow \\
\log P(Y|\mathbf{X},\mathbf{w}) &= \sum_{i=1}^{N} \left[y_i \log \lambda(\mathbf{x}_i) + (1-y_i)\log\left(1-\lambda(\mathbf{x}_i)\right)\right] = \\
&= \sum_{i=1}^{N} \left[y_i \log \operatorname{sig}(\mathbf{w}^T\mathbf{x}_i) + (1-y_i)\log\left(1-\operatorname{sig}(\mathbf{w}^T\mathbf{x}_i)\right)\right]
\end{aligned}$$

Optimize by setting: no close form solution! Use gradient descent

$$\frac{d}{d\mathbf{w}} \log P(Y|\mathbf{X},\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - \operatorname{sig}(\mathbf{w}^T\mathbf{x}_i)\right)\mathbf{x}_i = 0$$

Hints: derivatives of sigmoid

$$\frac{d}{d\mathbf{w}}\,\mathrm{sig}(\mathbf{w}^T\mathbf{x}) = \mathrm{sig}(\mathbf{w}^T\mathbf{x})\left(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x})\right)\mathbf{x}$$

$$\frac{d}{d\mathbf{w}}\log\left(\mathrm{sig}(\mathbf{w}^T\mathbf{x})\right) = \frac{\mathrm{sig}(\mathbf{w}^T\mathbf{x})\left(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x})\right)}{\mathrm{sig}(\mathbf{w}^T\mathbf{x})}\mathbf{x} = \left(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x})\right)\mathbf{x}$$

$$\frac{d}{d\mathbf{w}}\log\left(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x})\right) = \frac{-\mathrm{sig}(\mathbf{w}^T\mathbf{x})\left(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x})\right)}{1-\mathrm{sig}(\mathbf{w}^T\mathbf{x})}\mathbf{x} = -\mathrm{sig}(\mathbf{w}^T\mathbf{x})\;\mathbf{x}$$

Logistic Regression vs Conditional Gaussian

Number of parameters (D dimensions):

Gaussian distributions (equal priors)

$2 \times D$	(mean vectors)
$D(D+1)/2$	(shared covariance)
$D(D+5)/2$	(total, quadratic in D)

Logistic Regression

D (weights)

Training:

Gaussian distributions

- closed form solution
- generative model

Logistic Regression

- gradient descent
- discriminative model

Summary

1. Introduction
   - Probabilistic Classification and Regression
   - Discriminative vs Generative Models
   - Parametric vs Non-parametric Inference
2. Maximum Likelihood Estimation
   - Regression
   - Classification
3. Special Cases
   - Naïve Bayes Classifier
   - Logistic Regression

Further Reading

Some books on Probabilistic Machine Learning

- C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Verlag, 2006.
- Kevin P. Murphy, *Machine Learning A probabilistic Perspective*, MIT Press, 2012.
- Gelman et al., *Bayesian Data Analysis*, CRC Press, 2014.
- David Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2012.