Probabilistic Models for Classification: Discriminative Classification

Piyush Rai

Probabilistic Machine Learning (CS772A)

Aug 19, 2017

Recap

Generative Classification

- A two-step procedure to learn a probabilistic classification model $p(y|\boldsymbol{x})$

Generative Classification

- A two-step procedure to learn a probabilistic classification model $p(y|\boldsymbol{x})$
    1. Learn class-conditional distribution $p(\boldsymbol{x}|y)$ and class-prior distribution $p(y)$ using training data

Generative Classification

- A two-step procedure to learn a probabilistic classification model $p(y|\boldsymbol{x})$
  1. Learn class-conditional distribution $p(\boldsymbol{x}|y)$ and class-prior distribution $p(y)$ using training data
     - We usually assume the form of $p(\boldsymbol{x}|y)$ and $p(y)$, but the parameters are unknowns

Generative Classification

- A two-step procedure to learn a probabilistic classification model $p(y|\boldsymbol{x})$
  1. Learn class-conditional distribution $p(\boldsymbol{x}|y)$ and class-prior distribution $p(y)$ using training data
     - We usually assume the form of $p(\boldsymbol{x}|y)$ and $p(y)$, but the parameters are unknowns
     - Parameter estimation can be done via MLE or MAP or fully Bayesian inference

Generative Classification

- A two-step procedure to learn a probabilistic classification model $p(y|\boldsymbol{x})$
  1. Learn class-conditional distribution $p(\boldsymbol{x}|y)$ and class-prior distribution $p(y)$ using training data
     - We usually assume the form of $p(\boldsymbol{x}|y)$ and $p(y)$, but the parameters are unknowns
     - Parameter estimation can be done via MLE or MAP or fully Bayesian inference
  2. Apply Bayes rule to predict the posterior probability of each class for any (test) input $\boldsymbol{x}_*$

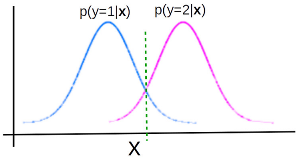
$$p(y_* = k|\boldsymbol{x}_*) = \frac{p(y_* = k)p(\boldsymbol{x}_*|y_* = k)}{p(\boldsymbol{x}_*)}$$

Generative Classification

- A two-step procedure to learn a probabilistic classification model $p(y|\boldsymbol{x})$
  1. Learn class-conditional distribution $p(\boldsymbol{x}|y)$ and class-prior distribution $p(y)$ using training data
     - We usually assume the form of $p(\boldsymbol{x}|y)$ and $p(y)$, but the parameters are unknowns
     - Parameter estimation can be done via MLE or MAP or fully Bayesian inference
  2. Apply Bayes rule to predict the posterior probability of each class for any (test) input $\boldsymbol{x}_*$

$$p(y_* = k|\boldsymbol{x}_*) = \frac{p(y_* = k)p(\boldsymbol{x}_*|y_* = k)}{p(\boldsymbol{x}_*)}$$

- The predicted class for the input $\boldsymbol{x}$ will be the one that has the largest posterior probability



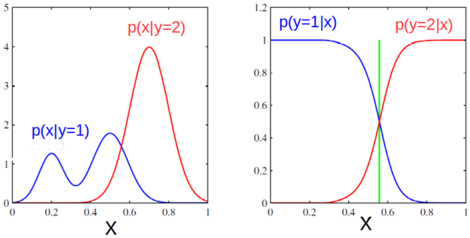
Generative Classification

- Generative classification requires modeling the input distribution (for each class)

Generative Classification

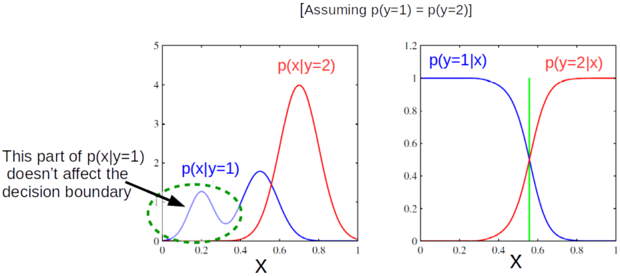
- Generative classification requires modeling the input distribution (for each class)
- In some cases, this extensive modeling effort may not be worth it (e.g., the case shown below)

[Assuming p(y=1) = p(y=2)]



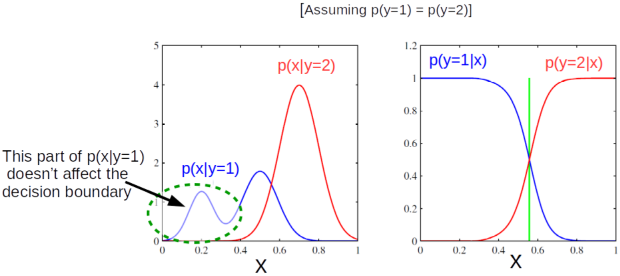
Generative Classification

- Generative classification requires modeling the input distribution (for each class)
- In some cases, this extensive modeling effort may not be worth it (e.g., the case shown below)



Generative Classification

- Generative classification requires modeling the input distribution (for each class)
- In some cases, this extensive modeling effort may not be worth it (e.g., the case shown below)

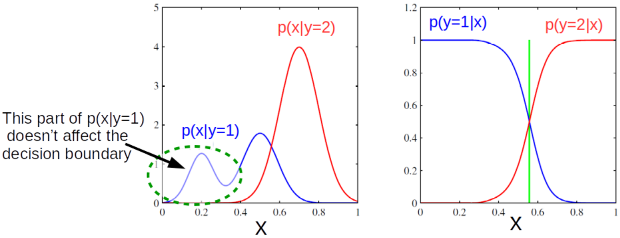


- A better approach would be to directly model $p(y|\boldsymbol{x})$, i.e., the distribution that makes the decision

Generative Classification

- Generative classification requires modeling the input distribution (for each class)
- In some cases, this extensive modeling effort may not be worth it (e.g., the case shown below)

[Assuming p(y=1) = p(y=2)]

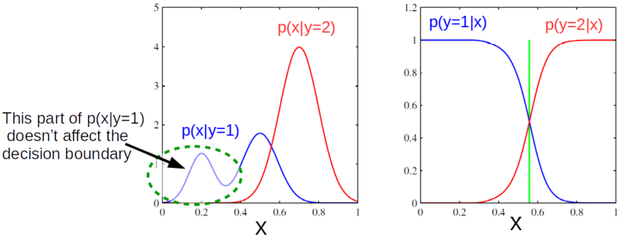


- A better approach would be to directly model $p(y|\boldsymbol{x})$, i.e., the distribution that makes the decision
  - Such an approach is called Discriminative Classification (today's topic)

Generative Classification

- Generative classification requires modeling the input distribution (for each class)
- In some cases, this extensive modeling effort may not be worth it (e.g., the case shown below)

[Assuming p(y=1) = p(y=2)]



- A better approach would be to directly model $p(y|\boldsymbol{x})$, i.e., the distribution that makes the decision
  - Such an approach is called Discriminative Classification (today's topic)
- PS: The generative approach is still appealing for many other reasons (as discussed in last lecture)

Probabilistic Models for Discriminative Classification

Probabilistic Models for Discriminative Classification

Note: Many non-probabilistic classification models can be termed as discriminative classifiers (e.g., Support Vector Machine or SVM, which directly learns a separator between classes)

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution
- Parameters of the distribution $p(y|\boldsymbol{x})$ defined by a linear/nonlinear function of $\boldsymbol{x}$

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution
- Parameters of the distribution $p(y|\boldsymbol{x})$ defined by a linear/nonlinear function of $\boldsymbol{x}$
  - A discriminative model only needs to learn these parameters!

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution
- Parameters of the distribution $p(y|\boldsymbol{x})$ defined by a linear/nonlinear function of $\boldsymbol{x}$
  - A discriminative model only needs to learn these parameters!
- For binary classification, $p(y|\boldsymbol{x})$ would be a Bernoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{Bernoulli}[f(\boldsymbol{x})]$$

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution
- Parameters of the distribution $p(y|\boldsymbol{x})$ defined by a linear/nonlinear function of $\boldsymbol{x}$
  - A discriminative model only needs to learn these parameters!
- For binary classification, $p(y|\boldsymbol{x})$ would be a Bernoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{Bernoulli}[f(\boldsymbol{x})]$$

where $f()$ is a function (defined by some params) that maps $\boldsymbol{x}$ to a probability $\mu = p(y = 1|\boldsymbol{x})$

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution
- Parameters of the distribution $p(y|\boldsymbol{x})$ defined by a linear/nonlinear function of $\boldsymbol{x}$
  - A discriminative model only needs to learn these parameters!
- For binary classification, $p(y|\boldsymbol{x})$ would be a Bernoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{Bernoulli}[f(\boldsymbol{x})]$$

  where $f()$ is a function (defined by some params) that maps $\boldsymbol{x}$ to a probability $\mu = p(y = 1|\boldsymbol{x})$
- For $K > 2$ class multiclass classification, $p(y|\boldsymbol{x})$ would be a multinoulli

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution
- Parameters of the distribution $p(y|\boldsymbol{x})$ defined by a linear/nonlinear function of $\boldsymbol{x}$
  - A discriminative model only needs to learn these parameters!
- For binary classification, $p(y|\boldsymbol{x})$ would be a Bernoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{Bernoulli}[f(\boldsymbol{x})]$$

where $f()$ is a function (defined by some params) that maps $\boldsymbol{x}$ to a probability $\mu = p(y = 1|\boldsymbol{x})$

- For $K > 2$ class multiclass classification, $p(y|\boldsymbol{x})$ would be a multinoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{multinoulli}[f(\boldsymbol{x})]$$

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution
- Parameters of the distribution $p(y|\boldsymbol{x})$ defined by a linear/nonlinear function of $\boldsymbol{x}$
  - A discriminative model only needs to learn these parameters!
- For binary classification, $p(y|\boldsymbol{x})$ would be a Bernoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{Bernoulli}[f(\boldsymbol{x})]$$

  where $f()$ is a function (defined by some params) that maps $\boldsymbol{x}$ to a probability $\mu = p(y = 1|\boldsymbol{x})$
- For $K > 2$ class multiclass classification, $p(y|\boldsymbol{x})$ would be a multinoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{multinoulli}[f(\boldsymbol{x})]$$

  where $f()$ maps $\boldsymbol{x}$ to a probability vector $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_K]$ s.t. $p(y = k|\boldsymbol{x}) = \mu_k$

(Probabilistic) Discriminative Classification

- Models $p(y|\boldsymbol{x})$ directly using an appropriate discrete distribution
- Parameters of the distribution $p(y|\boldsymbol{x})$ defined by a linear/nonlinear function of $\boldsymbol{x}$
  - A discriminative model only needs to learn these parameters!
- For binary classification, $p(y|\boldsymbol{x})$ would be a Bernoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{Bernoulli}[f(\boldsymbol{x})]$$

  where $f()$ is a function (defined by some params) that maps $\boldsymbol{x}$ to a probability $\mu = p(y = 1|\boldsymbol{x})$

- For $K > 2$ class multiclass classification, $p(y|\boldsymbol{x})$ would be a multinoulli

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, f) = \text{multinoulli}[f(\boldsymbol{x})]$$

  where $f()$ maps $\boldsymbol{x}$ to a probability vector $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_K]$ s.t. $p(y = k|\boldsymbol{x}) = \mu_k$

- Many choices of $f(\boldsymbol{x})$ possible. Lead to different types of discriminative classification models

(Probabilistic) Discriminative Classification

- The general form of $p(y|\boldsymbol{x})$ in these models is given by

$$p(y|\boldsymbol{x}) = \text{Bernoulli}[f(\boldsymbol{x})] \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}[f(\boldsymbol{x})]$$

(Probabilistic) Discriminative Classification

- The general form of $p(y|\boldsymbol{x})$ in these models is given by

$$p(y|\boldsymbol{x}) = \text{Bernoulli}[f(\boldsymbol{x})] \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}[f(\boldsymbol{x})]$$

- The function $f(\boldsymbol{x})$ is composed of two operations

(Probabilistic) Discriminative Classification

- The general form of $p(y|\boldsymbol{x})$ in these models is given by

$$p(y|\boldsymbol{x}) = \text{Bernoulli}[f(\boldsymbol{x})] \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}[f(\boldsymbol{x})]$$

- The function $f(\boldsymbol{x})$ is composed of two operations
  - Compute a score for $\boldsymbol{x}$ (e.g., using a linear model $\boldsymbol{w}^\top \boldsymbol{x}$ where $\boldsymbol{w}$ is the param. to be estimated)

(Probabilistic) Discriminative Classification

- The general form of $p(y|\boldsymbol{x})$ in these models is given by

$$p(y|\boldsymbol{x}) = \text{Bernoulli}[f(\boldsymbol{x})] \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}[f(\boldsymbol{x})]$$

- The function $f(\boldsymbol{x})$ is composed of two operations
  - Compute a score for $\boldsymbol{x}$ (e.g., using a linear model $\boldsymbol{w}^\top \boldsymbol{x}$ where $\boldsymbol{w}$ is the param. to be estimated)
  - Turn this score into a probability of $\boldsymbol{x}$ belonging to each class

(Probabilistic) Discriminative Classification

- The general form of $p(y|\boldsymbol{x})$ in these models is given by

$$p(y|\boldsymbol{x}) = \text{Bernoulli}[f(\boldsymbol{x})] \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}[f(\boldsymbol{x})]$$

- The function $f(\boldsymbol{x})$ is composed of two operations
  - Compute a score for $\boldsymbol{x}$ (e.g., using a linear model $\boldsymbol{w}^\top \boldsymbol{x}$ where $\boldsymbol{w}$ is the param. to be estimated)
  - Turn this score into a probability of $\boldsymbol{x}$ belonging to each class
- Example: $f(\boldsymbol{x}) = \mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})} \Rightarrow$ Logistic Regression; used for binary classification

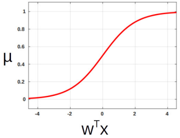
(Probabilistic) Discriminative Classification

- The general form of $p(y|\boldsymbol{x})$ in these models is given by

$$p(y|\boldsymbol{x}) = \text{Bernoulli}[f(\boldsymbol{x})] \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}[f(\boldsymbol{x})]$$

- The function $f(\boldsymbol{x})$ is composed of two operations
    - Compute a score for $\boldsymbol{x}$ (e.g., using a linear model $\boldsymbol{w}^\top \boldsymbol{x}$ where $\boldsymbol{w}$ is the param. to be estimated)
    - Turn this score into a probability of $\boldsymbol{x}$ belonging to each class
- Example: $f(\boldsymbol{x}) = \mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}$ $\Rightarrow$ Logistic Regression; used for binary classification

$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}$$



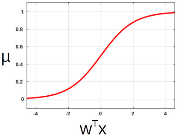
(Probabilistic) Discriminative Classification

- The general form of $p(y|\boldsymbol{x})$ in these models is given by

$$p(y|\boldsymbol{x}) = \text{Bernoulli}[f(\boldsymbol{x})] \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}[f(\boldsymbol{x})]$$

- The function $f(\boldsymbol{x})$ is composed of two operations
  - Compute a score for $\boldsymbol{x}$ (e.g., using a linear model $\boldsymbol{w}^\top \boldsymbol{x}$ where $\boldsymbol{w}$ is the param. to be estimated)
  - Turn this score into a probability of $\boldsymbol{x}$ belonging to each class
- Example: $f(\boldsymbol{x}) = \mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}$ $\Rightarrow$ Logistic Regression; used for binary classification

$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}$$



- Another example: $f(\boldsymbol{x}) = \boldsymbol{\mu}$ where $\mu_k = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x})}{\sum_{\ell=1}^K \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x})}$ $\Rightarrow$ Softmax Regression; used for multiclass

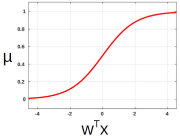
(Probabilistic) Discriminative Classification

- The general form of $p(y|\boldsymbol{x})$ in these models is given by

$$p(y|\boldsymbol{x}) = \text{Bernoulli}[f(\boldsymbol{x})] \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}[f(\boldsymbol{x})]$$

- The function $f(\boldsymbol{x})$ is composed of two operations
  - Compute a score for $\boldsymbol{x}$ (e.g., using a linear model $\boldsymbol{w}^\top \boldsymbol{x}$ where $\boldsymbol{w}$ is the param. to be estimated)
  - Turn this score into a probability of $\boldsymbol{x}$ belonging to each class
- Example: $f(\boldsymbol{x}) = \mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})} \Rightarrow$ Logistic Regression; used for binary classification

$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x})}$$



- Another example: $f(\boldsymbol{x}) = \boldsymbol{\mu}$ where $\mu_k = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x})}{\sum_{\ell=1}^{K} \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x})} \Rightarrow$ Softmax Regression; used for multiclass
- Can also use nonlinear models to compute the scores (e.g., deep NN or Gaussian Process)

Logistic Regression

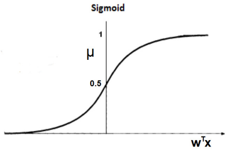
- Perhaps the simplest discriminative model for linear binary classification
- Defines $\mu = p(y = 1|\boldsymbol{x})$ using the sigmoid function

$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1 + \exp(\boldsymbol{w}^\top \boldsymbol{x})}$$

Logistic Regression

- Perhaps the simplest discriminative model for linear binary classification
- Defines $\mu = p(y = 1|\boldsymbol{x})$ using the sigmoid function

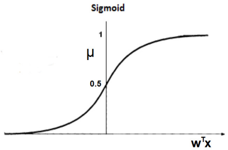
$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1 + \exp(\boldsymbol{w}^\top \boldsymbol{x})}$$



Logistic Regression

- Perhaps the simplest discriminative model for linear binary classification
- Defines $\mu = p(y = 1|\boldsymbol{x})$ using the sigmoid function

$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1 + \exp(\boldsymbol{w}^\top \boldsymbol{x})}$$

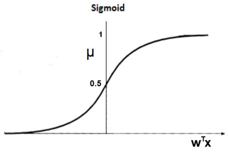


- Here $\boldsymbol{w}^\top \boldsymbol{x}$ is the score for input $\boldsymbol{x}$.

Logistic Regression

- Perhaps the simplest discriminative model for linear binary classification
- Defines $\mu = p(y = 1|\boldsymbol{x})$ using the sigmoid function

$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1 + \exp(\boldsymbol{w}^\top \boldsymbol{x})}$$

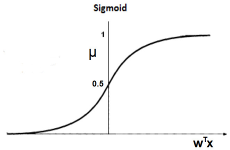


- Here $\boldsymbol{w}^\top \boldsymbol{x}$ is the score for input $\boldsymbol{x}$. The sigmoid turns it into a probability.

Logistic Regression

- Perhaps the simplest discriminative model for linear binary classification
- Defines $\mu = p(y = 1|\boldsymbol{x})$ using the sigmoid function

$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1 + \exp(\boldsymbol{w}^\top \boldsymbol{x})}$$



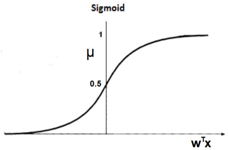
- Here $\boldsymbol{w}^\top \boldsymbol{x}$ is the score for input $\boldsymbol{x}$. The sigmoid turns it into a probability. Thus we have

$$\begin{aligned} p(y = 1|\boldsymbol{x}, \boldsymbol{w}) &= \mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1 + \exp(\boldsymbol{w}^\top \boldsymbol{x})} \\ p(y = 0|\boldsymbol{x}, \boldsymbol{w}) &= 1 - \mu = 1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + \exp(\boldsymbol{w}^\top \boldsymbol{x})} \end{aligned}$$

Logistic Regression

- Perhaps the simplest discriminative model for linear binary classification
- Defines $\mu = p(y = 1|\boldsymbol{x})$ using the sigmoid function

$$\mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})}$$



- Here $\boldsymbol{w}^\top \boldsymbol{x}$ is the score for input $\boldsymbol{x}$. The sigmoid turns it into a probability. Thus we have

$$\begin{aligned} p(y=1|\boldsymbol{x},\boldsymbol{w}) &= \mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})} \\ p(y=0|\boldsymbol{x},\boldsymbol{w}) &= 1-\mu = 1-\sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})} \end{aligned}$$

- **Note:** If we assume $y \in \{-1,+1\}$ instead of $y \in \{0,1\}$ then $p(y|\boldsymbol{x},\boldsymbol{w}) = \frac{1}{1+\exp(-y\boldsymbol{w}^\top \boldsymbol{x})}$

Logistic Regression: A Closer Look..

- At the decision boundary where both classes are equiprobable:

$$
\begin{aligned}
p(y=1|\boldsymbol{x},\boldsymbol{w}) &= p(y=0|\boldsymbol{x},\boldsymbol{w}) \\
\frac{\exp(\boldsymbol{w}^\top\boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})} &= \frac{1}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})} \\
\exp(\boldsymbol{w}^\top\boldsymbol{x}) &= 1 \\
\boldsymbol{w}^\top\boldsymbol{x} &= 0
\end{aligned}
$$

Logistic Regression: A Closer Look..

- At the decision boundary where both classes are equiprobable:

$$
\begin{aligned}
p(y=1|\boldsymbol{x},\boldsymbol{w}) &= p(y=0|\boldsymbol{x},\boldsymbol{w}) \\
\frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})} &= \frac{1}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})} \\
\exp(\boldsymbol{w}^\top \boldsymbol{x}) &= 1 \\
\boldsymbol{w}^\top \boldsymbol{x} &= 0
\end{aligned}
$$

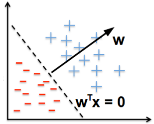
- Thus the decision boundary of LR is a linear hyperplane, just like Perceptron, SVM, etc.

Logistic Regression: A Closer Look..

- At the decision boundary where both classes are equiprobable:

$$
\begin{aligned}
p(y=1|\boldsymbol{x},\boldsymbol{w}) &= p(y=0|\boldsymbol{x},\boldsymbol{w}) \\
\frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})} &= \frac{1}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})} \\
\exp(\boldsymbol{w}^\top \boldsymbol{x}) &= 1 \\
\boldsymbol{w}^\top \boldsymbol{x} &= 0
\end{aligned}
$$

- Thus the decision boundary of LR is a linear hyperplane, just like Perceptron, SVM, etc.
- Therefore $y = 1$ if $\boldsymbol{w}^\top \boldsymbol{x} \geq 0$, otherwise $y = 0$

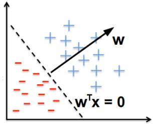


Logistic Regression: A Closer Look..

- At the decision boundary where both classes are equiprobable:

$$
\begin{aligned}
p(y=1|\boldsymbol{x},\boldsymbol{w}) &= p(y=0|\boldsymbol{x},\boldsymbol{w})\\
\frac{\exp(\boldsymbol{w}^\top\boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})} &= \frac{1}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})}\\
\exp(\boldsymbol{w}^\top\boldsymbol{x}) &= 1\\
\boldsymbol{w}^\top\boldsymbol{x} &= 0
\end{aligned}
$$

- Thus the decision boundary of LR is a linear hyperplane, just like Perceptron, SVM, etc.
- Therefore $y=1$ if $\boldsymbol{w}^\top\boldsymbol{x}\geq 0$, otherwise $y=0$



- High positive (negative) score $\boldsymbol{w}^\top\boldsymbol{x}$: High (low) probability of label 1

MLE for Logistic Regression

- Each label y_n is binary with probability $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$. Since the likelihood is Bernoulli:

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \prod_{n=1}^{N} \mu_n^{y_n}(1-\mu_n)^{1-y_n}$$

MLE for Logistic Regression

- Each label y_n is binary with probability $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$. Since the likelihood is Bernoulli:

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \prod_{n=1}^{N} \mu_n^{y_n}(1-\mu_n)^{1-y_n}$$

- Negative log-likelihood

$$\text{NLL}(\boldsymbol{w}) = -\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \log \mu_n + (1-y_n)\log(1-\mu_n))$$

MLE for Logistic Regression

- Each label y_n is binary with probability $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$. Since the likelihood is Bernoulli:

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \prod_{n=1}^{N} \mu_n^{y_n}(1-\mu_n)^{1-y_n}$$

- Negative log-likelihood

$$\text{NLL}(\boldsymbol{w}) = -\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \log \mu_n + (1-y_n)\log(1-\mu_n))$$

- Plugging in $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$ and chugging, we get (verify yourself)

$$\boxed{\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))}$$

MLE for Logistic Regression

- Each label y_n is binary with probability $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$. Since the likelihood is Bernoulli:

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \prod_{n=1}^{N} \mu_n^{y_n}(1-\mu_n)^{1-y_n}$$

- Negative log-likelihood

$$\text{NLL}(\boldsymbol{w}) = -\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \log \mu_n + (1-y_n)\log(1-\mu_n))$$

- Plugging in $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$ and chugging, we get (verify yourself)

$$\boxed{\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))}$$

- MLE solution: $\boldsymbol{w}_{MLE} = \arg\min_{\boldsymbol{w}} \text{NLL}(\boldsymbol{w})$

MLE for Logistic Regression

- Each label y_n is binary with probability $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$. Since the likelihood is Bernoulli:

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \prod_{n=1}^{N} \mu_n^{y_n}(1-\mu_n)^{1-y_n}$$

- Negative log-likelihood

$$\text{NLL}(\boldsymbol{w}) = -\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \log \mu_n + (1-y_n)\log(1-\mu_n))$$

- Plugging in $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$ and chugging, we get (verify yourself)

$$\boxed{\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))}$$

- MLE solution: $\boldsymbol{w}_{MLE} = \arg\min_{\boldsymbol{w}} \text{NLL}(\boldsymbol{w})$
- **Important note:** $\text{NLL}(\boldsymbol{w})$ is convex in $\boldsymbol{w}$, so global minima can be found

MLE for Logistic Regression

- We have $\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1 + \exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))$

MLE for Logistic Regression

- We have $\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1 + \exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))$
- Taking the derivative of $\text{NLL}(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}$

$$
\begin{aligned}
\frac{\partial \text{NLL}(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}}[-\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1 + \exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))] \\
&= -\sum_{n=1}^{N}\left(y_n \boldsymbol{x}_n - \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{(1 + \exp(\boldsymbol{w}^\top \boldsymbol{x}_n))}\boldsymbol{x}_n\right)
\end{aligned}
$$

MLE for Logistic Regression

- We have $\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))$
- Taking the derivative of $\text{NLL}(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}$

$$\begin{aligned}
\frac{\partial \text{NLL}(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}}[-\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))] \\
&= -\sum_{n=1}^{N}\left(y_n \boldsymbol{x}_n - \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n))}\boldsymbol{x}_n\right)
\end{aligned}$$

- Can't get a closed form estimate for $\boldsymbol{w}$ by setting the derivative to zero

MLE for Logistic Regression

- We have $\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))$
- Taking the derivative of $\text{NLL}(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}$

$$\begin{aligned}\frac{\partial \text{NLL}(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}}[-\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))] \\ &= -\sum_{n=1}^{N}\left(y_n \boldsymbol{x}_n - \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n))}\boldsymbol{x}_n\right)\end{aligned}$$

- Can't get a closed form estimate for $\boldsymbol{w}$ by setting the derivative to zero
- One solution: Iterative minimization via gradient descent $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \mathbf{g}_t$. The gradient is:

$$\boxed{\mathbf{g} = \frac{\partial \text{NLL}(\boldsymbol{w})}{\partial \boldsymbol{w}} = -\sum_{n=1}^{N}(y_n - \mu_n)\boldsymbol{x}_n = \mathbf{X}^\top(\boldsymbol{\mu} - \boldsymbol{y})}$$

MLE for Logistic Regression

- We have $\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1 + \exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))$
- Taking the derivative of $\text{NLL}(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}$

$$\begin{aligned}\frac{\partial \text{NLL}(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}}[-\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1 + \exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))] \\ &= -\sum_{n=1}^{N}\left(y_n \boldsymbol{x}_n - \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{(1 + \exp(\boldsymbol{w}^\top \boldsymbol{x}_n))}\boldsymbol{x}_n\right)\end{aligned}$$

- Can't get a closed form estimate for $\boldsymbol{w}$ by setting the derivative to zero
- One solution: Iterative minimization via gradient descent $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \mathbf{g}_t$. The gradient is:

$$\boxed{\mathbf{g} = \frac{\partial \text{NLL}(\boldsymbol{w})}{\partial \boldsymbol{w}} = -\sum_{n=1}^{N}(y_n - \mu_n)\boldsymbol{x}_n = \mathbf{X}^\top(\boldsymbol{\mu} - \boldsymbol{y})}$$

- A large error on $\boldsymbol{x}_n \Rightarrow (y_n - \mu_n)$ will be large $\Rightarrow$ large contribution of $\boldsymbol{x}_n$ to the gradient

MLE for Logistic Regression

- We have $\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))$
- Taking the derivative of $\text{NLL}(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}$

$$
\begin{aligned}
\frac{\partial \text{NLL}(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}}[-\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))] \\
&= -\sum_{n=1}^{N}\left(y_n \boldsymbol{x}_n - \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n))}\boldsymbol{x}_n\right)
\end{aligned}
$$

- Can't get a closed form estimate for $\boldsymbol{w}$ by setting the derivative to zero
- One solution: Iterative minimization via gradient descent $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \mathbf{g}_t$. The gradient is:

$$
\boxed{\mathbf{g} = \frac{\partial \text{NLL}(\boldsymbol{w})}{\partial \boldsymbol{w}} = -\sum_{n=1}^{N}(y_n - \mu_n)\boldsymbol{x}_n = \mathbf{X}^\top(\boldsymbol{\mu} - \boldsymbol{y})}
$$

- A large error on $\boldsymbol{x}_n \Rightarrow (y_n - \mu_n)$ will be large $\Rightarrow$ large contribution of $\boldsymbol{x}_n$ to the gradient
- More sophisticated methods (e.g., Newton's method) can also be used (I'll provide a note)

MAP Estimation for Logisic Regression

- MLE estimate of $\boldsymbol{w}$ can lead to overfitting. Solution: use a prior $p(\boldsymbol{w})$ on $\boldsymbol{w}$
- Just like the linear regression case, let's put a Gausian prior on $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I}_D) \propto \exp(-\frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w})$$

MAP Estimation for Logisic Regression

- MLE estimate of $\boldsymbol{w}$ can lead to overfitting. Solution: use a prior $p(\boldsymbol{w})$ on $\boldsymbol{w}$
- Just like the linear regression case, let's put a Gausian prior on $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I}_D) \propto \exp(-\frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w})$$

- MAP objective (log of posterior) = MLE objective + $\log p(\boldsymbol{w})$
- Leads to the objective (negative of log posterior, ignoring constants):

$$\text{NLL}(\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}$$

MAP Estimation for Logisic Regression

- MLE estimate of $\boldsymbol{w}$ can lead to overfitting. Solution: use a prior $p(\boldsymbol{w})$ on $\boldsymbol{w}$
- Just like the linear regression case, let's put a Gausian prior on $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I}_D) \propto \exp(-\frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w})$$

- MAP objective (log of posterior) = MLE objective + $\log p(\boldsymbol{w})$
- Leads to the objective (negative of log posterior, ignoring constants):

$$\text{NLL}(\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}$$

- Estimation of $\boldsymbol{w}_{MAP}$ proceeds the same way as MLE using gradient based methods, with

$$\text{Gradient:} \quad \mathbf{g} = \mathbf{X}^\top(\boldsymbol{\mu} - \boldsymbol{y}) + \lambda \boldsymbol{w}$$

Fully Bayesian Estimation for Logistic Regression

- Doing fully Bayesian inference would require computing the posterior

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}$$

Fully Bayesian Estimation for Logistic Regression

- Doing fully Bayesian inference would require computing the posterior

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}} = \frac{\prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})}{\int \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}$$

- Intractable. Reason: likelihood (logistic-Bernoulli) and prior (Gaussian) here are **not conjugate**

Fully Bayesian Estimation for Logistic Regression

- Doing fully Bayesian inference would require computing the posterior

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}} = \frac{\prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})}{\int \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}$$

- Intractable. Reason: likelihood (logistic-Bernoulli) and prior (Gaussian) here are **not conjugate**
- Need to do *approximate inference* in this case

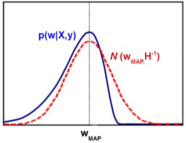
Fully Bayesian Estimation for Logistic Regression

- Doing fully Bayesian inference would require computing the posterior

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}} = \frac{\prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})}{\int \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}$$

- Intractable. Reason: likelihood (logistic-Bernoulli) and prior (Gaussian) here are **not conjugate**
- Need to do *approximate inference* in this case
- A crude approximation: **Laplace approximation**: Approximate a posterior by a **Gaussian** with mean $=\boldsymbol{w}_{MAP}$ and covariance = inverse hessian (hessian = second derivative of log $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})$)

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{w}_{MAP}, \mathbf{H}^{-1})$$



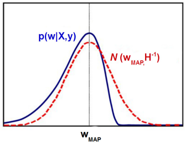
Fully Bayesian Estimation for Logistic Regression

- Doing fully Bayesian inference would require computing the posterior

$$p(\boldsymbol{w}|\mathbf{X},\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}} = \frac{\prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n,\boldsymbol{w})p(\boldsymbol{w})}{\int \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n,\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}$$

- Intractable. Reason: likelihood (logistic-Bernoulli) and prior (Gaussian) here are **not conjugate**
- Need to do *approximate inference* in this case
- A crude approximation: **Laplace approximation**: Approximate a posterior by a **Gaussian** with mean $=\boldsymbol{w}_{MAP}$ and covariance = inverse hessian (hessian = second derivative of log $p(\boldsymbol{w}|\mathbf{X},\boldsymbol{y})$)

$$p(\boldsymbol{w}|\mathbf{X},\boldsymbol{y}) = \mathcal{N}(\boldsymbol{w}_{MAP},\mathbf{H}^{-1})$$



- Will look at Laplace and other approximate inference methods later during the semester

Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) = \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*)$$

Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$
\begin{aligned}
p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*) \\
p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*))
\end{aligned}
$$

Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$
\begin{aligned}
p(y_* = 1 | \boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*) \\
p(y_* | \boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*))
\end{aligned}
$$

- When using MAP, the predictive distribution will be

$$
p(y_* | \boldsymbol{x}_*, \boldsymbol{w}_{MAP}) = \text{Bernoulli}(\sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*))
$$

Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$
\begin{aligned}
p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*) \\
p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*))
\end{aligned}
$$

- When using MAP, the predictive distribution will be

$$
\begin{aligned}
p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*)) \\
p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*)
\end{aligned}
$$

Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$\begin{aligned} p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*) \\ p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*)) \end{aligned}$$

- When using MAP, the predictive distribution will be

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*)) \\ p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*) \end{aligned}$$

- When using Bayesian inference, the posterior predictive distribution, based on posterior averaging

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}) p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) d\boldsymbol{w}$$

Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$\begin{aligned} p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*) \\ p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*)) \end{aligned}$$

- When using MAP, the predictive distribution will be

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*)) \\ p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*) \end{aligned}$$

- When using Bayesian inference, the posterior predictive distribution, based on posterior averaging

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w})p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})d\boldsymbol{w} = \int \sigma(\boldsymbol{w}^\top \boldsymbol{x}_*)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})d\boldsymbol{w}$$

Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$\begin{aligned} p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*) \\ p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*)) \end{aligned}$$

- When using MAP, the predictive distribution will be

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*)) \\ p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*) \end{aligned}$$

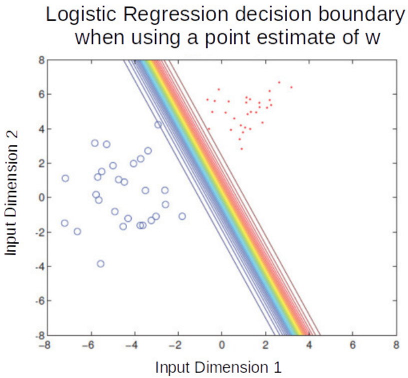
- When using Bayesian inference, the posterior predictive distribution, based on posterior averaging

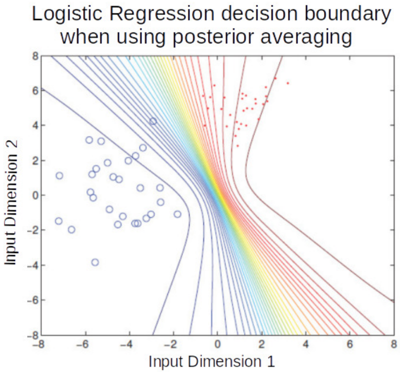
$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w})p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})d\boldsymbol{w} = \int \sigma(\boldsymbol{w}^\top \boldsymbol{x}_*)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})d\boldsymbol{w}$$

- Note: Unlike the linear regression case, for logistic regression (and for non-conjugate models in general), posterior averaging can be intractable (and may require approximations)

Logistic Regression: Plug-in Prediction vs Bayesian Averaging

- (Left) Predictive distribution when using a point estimate uses only a single linear hyperplane $\boldsymbol{w}$
- (Right) Posterior predictive distribution averages over many linear hyperplanes $\boldsymbol{w}$





Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \ldots, K\}$ and label probabilities are defined as

$$p(y_n = k | \boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^{K} \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk}$$

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \ldots, K\}$ and label probabilities are defined as

$$p(y_n = k|\boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^{K} \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk} \quad \text{and} \quad \sum_{\ell=1}^{K} \mu_{n\ell} = 1$$

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \ldots, K\}$ and label probabilities are defined as

$$p(y_n = k | \boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^K \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk} \quad \text{and} \quad \sum_{\ell=1}^K \mu_{n\ell} = 1$$

- $\mathbf{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \ldots \ \boldsymbol{w}_K]$ is $D \times K$ weight matrix.

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \dots, K\}$ and label probabilities are defined as

$$p(y_n = k|\boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^{K} \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk} \quad \text{and} \quad \sum_{\ell=1}^{K} \mu_{n\ell} = 1$$

- $\mathbf{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \dots \ \boldsymbol{w}_K]$ is $D \times K$ weight matrix. $\boldsymbol{w}_1 = \mathbf{0}_{D\times 1}$ (assumed for identifiability)

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \ldots, K\}$ and label probabilities are defined as

$$p(y_n = k|\boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^{K} \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk} \quad \text{and} \quad \sum_{\ell=1}^{K} \mu_{n\ell} = 1$$

- $\mathbf{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \ldots \ \boldsymbol{w}_K]$ is $D \times K$ weight matrix. $\boldsymbol{w}_1 = \mathbf{0}_{D\times 1}$ (assumed for identifiability)
- Called softmax because.. ?

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \dots, K\}$ and label probabilities are defined as

$$p(y_n = k|\boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^K \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk} \quad \text{and} \quad \sum_{\ell=1}^K \mu_{n\ell} = 1$$

- $\mathbf{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \dots \ \boldsymbol{w}_K]$ is $D \times K$ weight matrix. $\boldsymbol{w}_1 = \mathbf{0}_{D\times 1}$ (assumed for identifiability)
- Called softmax because.. ?
- Each likelihood $p(y_n|\boldsymbol{x}_n, \mathbf{W})$ is a multinoulli. Therefore

$$p(\boldsymbol{y}|\mathbf{X}, \mathbf{W}) = \prod_{n=1}^N \prod_{\ell=1}^K \mu_{n\ell}^{y_{n\ell}}$$

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \ldots, K\}$ and label probabilities are defined as

$$p(y_n = k|\boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^K \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk} \quad \text{and} \quad \sum_{\ell=1}^K \mu_{n\ell} = 1$$

- $\mathbf{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \ldots \ \boldsymbol{w}_K]$ is $D \times K$ weight matrix. $\boldsymbol{w}_1 = \mathbf{0}_{D\times 1}$ (assumed for identifiability)
- Called softmax because.. ?
- Each likelihood $p(y_n|\boldsymbol{x}_n, \mathbf{W})$ is a multinoulli. Therefore

$$p(\boldsymbol{y}|\mathbf{X}, \mathbf{W}) = \prod_{n=1}^N \prod_{\ell=1}^K \mu_{n\ell}^{y_{n\ell}}$$

where $y_{n\ell} = 1$ if true class of example n is ℓ and $y_{n\ell'} = 0$ for all other $\ell' \neq \ell$

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \ldots, K\}$ and label probabilities are defined as

$$p(y_n = k | \boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^{K} \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk} \quad \text{and} \quad \sum_{\ell=1}^{K} \mu_{n\ell} = 1$$

- $\mathbf{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \ldots \ \boldsymbol{w}_K]$ is $D \times K$ weight matrix. $\boldsymbol{w}_1 = \mathbf{0}_{D \times 1}$ (assumed for identifiability)
- Called softmax because.. ?
- Each likelihood $p(y_n | \boldsymbol{x}_n, \mathbf{W})$ is a multinoulli. Therefore

$$p(\boldsymbol{y} | \mathbf{X}, \mathbf{W}) = \prod_{n=1}^{N} \prod_{\ell=1}^{K} \mu_{n\ell}^{y_{n\ell}}$$

where $y_{n\ell} = 1$ if true class of example n is ℓ and $y_{n\ell'} = 0$ for all other $\ell' \neq \ell$

- Can do MLE/MAP/fully Bayesian estimation for $\mathbf{W}$ similar to the logistic regression model

Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, logistic regression for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \ldots, K\}$ and label probabilities are defined as

$$p(y_n = k | \boldsymbol{x}_n, \mathbf{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^K \exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk} \quad \text{and} \quad \sum_{\ell=1}^K \mu_{n\ell} = 1$$

- $\mathbf{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \ldots \ \boldsymbol{w}_K]$ is $D \times K$ weight matrix. $\boldsymbol{w}_1 = \mathbf{0}_{D\times 1}$ (assumed for identifiability)
- Called softmax because.. ?
- Each likelihood $p(y_n | \boldsymbol{x}_n, \mathbf{W})$ is a multinoulli. Therefore

$$p(\boldsymbol{y} | \mathbf{X}, \mathbf{W}) = \prod_{n=1}^N \prod_{\ell=1}^K \mu_{n\ell}^{y_{n\ell}}$$

where $y_{n\ell} = 1$ if true class of example n is ℓ and $y_{n\ell'} = 0$ for all other $\ell' \neq \ell$

- Can do MLE/MAP/fully Bayesian estimation for $\mathbf{W}$ similar to the logistic regression model
- Doing MLE is like minimizing the cross-entropy loss: $-\sum_{n=1}^N \sum_{\ell=1}^L y_{n\ell} \log \mu_{n\ell}$

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification
- These are special cases of the more general form of discriminative models for classification

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(f(\boldsymbol{x})) \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}(f(\boldsymbol{x}))$$

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification
- These are special cases of the more general form of discriminative models for classification

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(f(\boldsymbol{x})) \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}(f(\boldsymbol{x}))$$

- Logistic and softmax also belong to the class of "Generalized Linear Models" (GLM)

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification
- These are special cases of the more general form of discriminative models for classification

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(f(\boldsymbol{x})) \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}(f(\boldsymbol{x}))$$

- Logistic and softmax also belong to the class of "Generalized Linear Models" (GLM)
  - GLM can model outputs of various types, e.g., real, categorical, counts, positive reals, etc.

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification
- These are special cases of the more general form of discriminative models for classification

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(f(\boldsymbol{x})) \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}(f(\boldsymbol{x}))$$

- Logistic and softmax also belong to the class of "Generalized Linear Models" (GLM)
  - GLM can model outputs of various types, e.g., real, categorical, counts, positive reals, etc.
  - Will look at GLM when we discuss exponential families

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification
- These are special cases of the more general form of discriminative models for classification

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(f(\boldsymbol{x})) \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}(f(\boldsymbol{x}))$$

- Logistic and softmax also belong to the class of "Generalized Linear Models" (GLM)
    - GLM can model outputs of various types, e.g., real, categorical, counts, positive reals, etc.
    - Will look at GLM when we discuss exponential families
- Logistic/softmax can also be extended to handle nonlinear classification, e.g.,

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification
- These are special cases of the more general form of discriminative models for classification

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(f(\boldsymbol{x})) \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}(f(\boldsymbol{x}))$$

- Logistic and softmax also belong to the class of "Generalized Linear Models" (GLM)
  - GLM can model outputs of various types, e.g., real, categorical, counts, positive reals, etc.
  - Will look at GLM when we discuss exponential families
- Logistic/softmax can also be extended to handle nonlinear classification, e.g.,
  - Using a deep neural network + sigmoid/softmax to model $f(x)$

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification
- These are special cases of the more general form of discriminative models for classification

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(f(\boldsymbol{x})) \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}(f(\boldsymbol{x}))$$

- Logistic and softmax also belong to the class of "Generalized Linear Models" (GLM)
  - GLM can model outputs of various types, e.g., real, categorical, counts, positive reals, etc.
  - Will look at GLM when we discuss exponential families
- Logistic/softmax can also be extended to handle nonlinear classification, e.g.,
  - Using a deep neural network + sigmoid/softmax to model $f(x)$
  - Using Gaussian Processes (GP) which are essentially Bayesian kernel methods

Some Thoughts on Logistic and Softmax Classification

- Both logistic and softmax classification are discriminative model for linear classification
- These are special cases of the more general form of discriminative models for classification

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(f(\boldsymbol{x})) \qquad \text{OR} \qquad p(y|\boldsymbol{x}) = \text{multinoulli}(f(\boldsymbol{x}))$$

- Logistic and softmax also belong to the class of "Generalized Linear Models" (GLM)
  - GLM can model outputs of various types, e.g., real, categorical, counts, positive reals, etc.
  - Will look at GLM when we discuss exponential families
- Logistic/softmax can also be extended to handle nonlinear classification, e.g.,
  - Using a deep neural network + sigmoid/softmax to model $f(x)$
  - Using Gaussian Processes (GP) which are essentially Bayesian kernel methods
  - We will look at such nonlinear classification models later..

Some Thoughts on Generative vs Discriminative Classification

- Number of parameters: Discriminative models have fewer parameters to be learned (e.g., just the weight vector/matrix $\boldsymbol{w}/\mathbf{W}$ in case of logistic/softmax classification

Some Thoughts on Generative vs Discriminative Classification

- Number of parameters: Discriminative models have fewer parameters to be learned (e.g., just the weight vector/matrix $\boldsymbol{w}/\mathbf{W}$ in case of logistic/softmax classification
- Ease of parameter estimation: Debatable as to which one is easier

Some Thoughts on Generative vs Discriminative Classification

- Number of parameters: Discriminative models have fewer parameters to be learned (e.g., just the weight vector/matrix $\boldsymbol{w}/\mathbf{W}$ in case of logistic/softmax classification
- Ease of parameter estimation: Debatable as to which one is easier
  - For "simple" class-conditionals (e.g., naïve Bayes assumption, diagonal covariance for Gaussian $p(\boldsymbol{x}|y)$), it is easier for generative classification model (often closed-form solution)

Some Thoughts on Generative vs Discriminative Classification

- Number of parameters: Discriminative models have fewer parameters to be learned (e.g., just the weight vector/matrix $\boldsymbol{w}/\mathbf{W}$ in case of logistic/softmax classification
- Ease of parameter estimation: Debatable as to which one is easier
  - For "simple" class-conditionals (e.g., naïve Bayes assumption, diagonal covariance for Gaussian $p(\boldsymbol{x}|y)$), it is easier for generative classification model (often closed-form solution)
  - Parameter estimation for discriminative models (logistic/softmax) usually requires iterative methods (although objective functions usaully have global optima)

Some Thoughts on Generative vs Discriminative Classification

- Number of parameters: Discriminative models have fewer parameters to be learned (e.g., just the weight vector/matrix $\boldsymbol{w}/\mathbf{W}$ in case of logistic/softmax classification
- Ease of parameter estimation: Debatable as to which one is easier
  - For "simple" class-conditionals (e.g., naïve Bayes assumption, diagonal covariance for Gaussian $p(\boldsymbol{x}|y)$), it is easier for generative classification model (often closed-form solution)
  - Parameter estimation for discriminative models (logistic/softmax) usually requires iterative methods (although objective functions usaully have global optima)
- Dealing with missing features in the inputs: Generative models can handle this easily (e.g., by integrating out the missing features while estimating the parameters)

Some Thoughts on Generative vs Discriminative Classification

- Number of parameters: Discriminative models have fewer parameters to be learned (e.g., just the weight vector/matrix $\boldsymbol{w}/\mathbf{W}$ in case of logistic/softmax classification
- Ease of parameter estimation: Debatable as to which one is easier
  - For "simple" class-conditionals (e.g., naïve Bayes assumption, diagonal covariance for Gaussian $p(\boldsymbol{x}|y)$), it is easier for generative classification model (often closed-form solution)
  - Parameter estimation for discriminative models (logistic/softmax) usually requires iterative methods (although objective functions usaully have global optima)
- Dealing with missing features in the inputs: Generative models can handle this easily (e.g., by integrating out the missing features while estimating the parameters)
- Inputs with features having mixed types: Naturally handled by a generative model using an appropriate $p(\boldsymbol{x}|y)$ for each type of feature in the input. Difficult for discriminative models

Some Thoughts on Generative vs Discriminative Classification

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and ideal for Semi-supervised Learning. Discriminative models can't do it easily

Some Thoughts on Generative vs Discriminative Classification

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and ideal for Semi-supervised Learning. Discriminative models can't do it easily
- Adding data from new classes: Discriminative model will need to be re-trained. Generative model will just require estimating the class-conditionals of newly added classes

Some Thoughts on Generative vs Discriminative Classification

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and ideal for Semi-supervised Learning. Discriminative models can't do it easily
- Adding data from new classes: Discriminative model will need to be re-trained. Generative model will just require estimating the class-conditionals of newly added classes
- Have lots of labeled data: Discriminative models usually work very well

Some Thoughts on Generative vs Discriminative Classification

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and ideal for Semi-supervised Learning. Discriminative models can't do it easily
- Adding data from new classes: Discriminative model will need to be re-trained. Generative model will just require estimating the class-conditionals of newly added classes
- Have lots of labeled data: Discriminative models usually work very well
- Final Verdict? Despite generative classification having some clear advantages, both methods can be equally powerful (the actual choice may be dictated by the problem)

Some Thoughts on Generative vs Discriminative Classification

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and ideal for Semi-supervised Learning. Discriminative models can't do it easily
- Adding data from new classes: Discriminative model will need to be re-trained. Generative model will just require estimating the class-conditionals of newly added classes
- Have lots of labeled data: Discriminative models usually work very well
- Final Verdict? Despite generative classification having some clear advantages, both methods can be equally powerful (the actual choice may be dictated by the problem)
  - Important to be aware of their strengths/weaknesses, and also the connections between these models

Some Thoughts on Generative vs Discriminative Classification

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and ideal for Semi-supervised Learning. Discriminative models can't do it easily
- Adding data from new classes: Discriminative model will need to be re-trained. Generative model will just require estimating the class-conditionals of newly added classes
- Have lots of labeled data: Discriminative models usually work very well
- Final Verdict? Despite generative classification having some clear advantages, both methods can be equally powerful (the actual choice may be dictated by the problem)
  - Important to be aware of their strengths/weaknesses, and also the connections between these models
- Possibility of a Hybrid Design? Yes, Generative and Discriminative models also be combined, e.g.,

Some Thoughts on Generative vs Discriminative Classification

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and ideal for Semi-supervised Learning. Discriminative models can't do it easily
- Adding data from new classes: Discriminative model will need to be re-trained. Generative model will just require estimating the class-conditionals of newly added classes
- Have lots of labeled data: Discriminative models usually work very well
- Final Verdict? Despite generative classification having some clear advantages, both methods can be equally powerful (the actual choice may be dictated by the problem)
  - Important to be aware of their strengths/weaknesses, and also the connections between these models
- Possibility of a Hybrid Design? Yes, Generative and Discriminative models also be combined, e.g.,
  - "Principled Hybrids of Generative and Discriminative Models" (Lassere et al, 2006)
  - "Deep Hybrid Models: Bridging Discriminative & Generative Approaches" (Kuleshov & Ermon, 2017)