Generative v. Discriminative classifiers – Intuition

- **Want to Learn:** $h: X \rightarrow Y$
  - $X$ – features
  - $Y$ – target classes
- **Bayes optimal classifier** – $P(Y|X)$
- **Generative classifier**, e.g., Naïve Bayes:
  - Assume some functional form for $P(X|Y)$, $P(Y)$
  - Estimate parameters of $P(X|Y)$, $P(Y)$ directly from training data
  - Use Bayes rule to calculate $P(Y|X=x)$
  - This is a ‘*generative*’ model
    - Indirect computation of $P(Y|X)$ through Bayes rule
    - But, can generate a *sample of the data*, $P(X) = \sum_y P(y) P(X|y)$
- **Discriminative classifiers**, e.g., Logistic Regression:
  - Assume some functional form for $P(Y|X)$
  - Estimate parameters of $P(Y|X)$ directly from training data
  - This is the ‘*discriminative*’ model
    - Directly learn $P(Y|X)$
    - But cannot obtain a *sample of the data*, because $P(X)$ is not available
Logistic Regression

- Learn $P(Y|X)$ directly!
  - Assume a particular functional form
  - Sigmoid applied to a linear function of the data:

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i x_i)}$$

- Features can be discrete or continuous!

Understanding the sigmoid

$$g(w_0 + \sum_{i} w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_{i} w_i x_i}}$$

- $w_0 = -2, w_i = -1$
- $w_0 = 0, w_1 = -1$
- $w_0 = 0, w_1 = -0.5$
Logistic Regression – a Linear classifier

\[ g(w_0 + \sum_i w_ix_i) = \frac{1}{1 + e^{w_0 + \sum_i w_ix_i}} \]

Very convenient!

\[ P(Y = 1|X = < X_1, ..., X_n >) = \frac{1}{1 + \exp(w_0 + \sum_i w_iX_i)} \]

implies

\[ P(Y = 0|X = < X_1, ..., X_n >) = \frac{\exp(w_0 + \sum_i w_iX_i)}{1 + \exp(w_0 + \sum_i w_iX_i)} \]

implies

\[ \frac{P(Y = 0|X)}{P(Y = 1|X)} = \exp(w_0 + \sum_i w_iX_i) \]

implies

\[ \ln \frac{P(Y = 0|X)}{P(Y = 1|X)} = w_0 + \sum_i w_iX_i \]
What if we have continuous $X_i$?

Eg., character recognition: $X_i$ is $i^{th}$ pixel

Gaussian Naïve Bayes (GNB):

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

Sometimes assume variance
- is independent of $Y$ (i.e., $\sigma_y$),
- or independent of $X_i$ (i.e., $\sigma_x$)
- or both (i.e., $\sigma$)

Example: GNB for classifying mental states

~1 mm resolution
~2 images per sec.
15,000 voxels/image
non-invasive, safe
measures Blood Oxygen Level Dependent (BOLD) response

Typical impulse response
Learned Bayes Models – Means for $P(\text{BrainActivity} \mid \text{WordCategory})$

Pairwise classification accuracy: 85%

People words  Animal words

Logistic regression v. Naïve Bayes

- Consider learning $f: X \rightarrow Y$, where
  - $X$ is a vector of real-valued features, $< X_1 \ldots X_n >$
  - $Y$ is boolean
- Could use a Gaussian Naïve Bayes classifier
  - assume all $X_i$ are conditionally independent given $Y$
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik}, \sigma_i)$
  - model $P(Y)$ as Bernoulli($\theta$, 1-$\theta$)
- What does that imply about the form of $P(Y|X)$?

$$P(Y = 1|X = < X_1, ..., X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

Cool!!!!
Derive form for $P(Y|X)$ for continuous $X_i$

$$P(Y = 1|X) = \frac{P(Y = 1)P(X|Y = 1)}{P(Y = 1)P(X|Y = 1) + P(Y = 0)P(X|Y = 0)}$$

$$= \frac{1}{1 + \frac{P(Y = 0)P(X|Y = 0)}{P(Y = 1)P(X|Y = 1)}}$$

$$= \frac{1}{1 + \exp(\ln\frac{P(Y = 0)P(X|Y = 0)}{P(Y = 1)P(X|Y = 1)})}$$

$$= \frac{1}{1 + \exp(\ln\left(\frac{1}{\theta}\right) + \sum_i \ln\frac{P(X_i|Y = 0)}{P(X_i|Y = 1)})}$$

---

Ratio of class-conditional probabilities

$$\ln\frac{P(X_i|Y = 0)}{P(X_i|Y = 1)}$$

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-y_k)^2}{2\sigma^2}}$$
Derive form for $P(Y|X)$ for continuous $X_i$

$$P(Y = 1|X) = \frac{P(Y = 1)P(X|Y = 1)}{P(Y = 1)P(X|Y = 1) + P(Y = 0)P(X|Y = 0)} = \frac{1}{1 + \exp\left(\ln \frac{1}{\theta} + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right)} \sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2}X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}\right)$$

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^n w_iX_i)}$$

Gaussian Naïve Bayes v. Logistic Regression

Set of Gaussian Naïve Bayes parameters (feature variance independent of class label)  
Set of Logistic Regression parameters

- Representation equivalence
  - But only in a special case!!! (GNB with class-independent variances)
- But what’s the difference???
- LR makes no assumptions about $P(X|Y)$ in learning!!!
- Loss function!!!
  - Optimize different functions → Obtain different solutions
Logistic regression for more than 2 classes

- Logistic regression in more general case, where \( Y \in \{Y_1 \ldots Y_R\} \): learn \( R-1 \) sets of weights

---

Logistic regression more generally

- Logistic regression in more general case, where \( Y \in \{Y_1 \ldots Y_R\} \): learn \( R-1 \) sets of weights

For \( k < R \)

\[
P(Y = y_k | X) = \frac{\exp(w_{k0} + \sum_{i=1}^{n} w_{ki}X_i)}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}
\]

For \( k = R \) (normalization, so no weights for this class)

\[
P(Y = y_R | X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}
\]

Features can be discrete or continuous!
Loss functions: Likelihood v. Conditional Likelihood

- Generative (Naïve Bayes) Loss function:
  
  \[ \ln P(D \mid w) = \sum_{j=1}^{N} \ln P(x^j, y^j \mid w) \]
  
  \[ = \sum_{j=1}^{N} \ln P(y^j \mid x^j, w) + \sum_{j=1}^{N} \ln P(x^j \mid w) \]

- Discriminative models cannot compute \( P(x \mid w) \)!
- But, discriminative (logistic regression) loss function:

  **Conditional Data Likelihood**

  \[ \ln P(D_Y \mid D_X, w) = \sum_{j=1}^{N} \ln P(y^j \mid x^j, w) \]

  - Doesn’t waste effort learning \( P(X) \) – focuses on \( P(Y \mid X) \) all that matters for classification

Expressing Conditional Log Likelihood

\[ l(w) \equiv \sum_{j} \ln P(y^j \mid x^j, w) \]

\[ P(Y = 0 \mid X, w) = \frac{1}{1 + \exp(w_0 + \sum_i w_i x_i)} \]

\[ P(Y = 1 \mid X, w) = \frac{\exp(w_0 + \sum_i w_i x_i)}{1 + \exp(w_0 + \sum_i w_i x_i)} \]

\[ l(w) = \sum_{j} \left[ y^j \ln P(y = 1 \mid x^j, w) + (1 - y^j) \ln P(y = 0 \mid x^j, w) \right] \]
Maximizing Conditional Log Likelihood

\( l(w) \equiv \ln \prod_j P(y_j | x_j, w) \)

\[
l(w) = \sum_j \left[ y_j (w_0 + \sum_i w_i x_{ij}) - \ln(1 + \exp(w_0 + \sum_i w_i x_{ij})) \right]
\]

**Good news**: \( l(w) \) is concave function of \( w \) → no locally optimal solutions

**Bad news**: no closed-form solution to maximize \( l(w) \)

**Good news**: concave functions easy to optimize

Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave → Find optimum with gradient ascent

Gradient:

\[
\nabla_w l(w) = \left[ \frac{\partial l(w)}{\partial w_0}, \ldots, \frac{\partial l(w)}{\partial w_n} \right]'
\]

Update rule:

\[
\Delta w = \eta \nabla_w l(w)
\]

\[w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(w)}{\partial w_i}
\]

- Gradient ascent is simplest of optimization approaches
  - e.g., Conjugate gradient ascent much better (see reading)
Maximize Conditional Log Likelihood: Gradient ascent

\[ t(w) = \sum_j y^j (w_0 + \sum_i w_i x_i^j) - \ln (1 + \exp (w_0 + \sum_i w_i x_i^j)) \]

Gradient Descent for LR

Gradient ascent algorithm: iterate until change < \( \varepsilon \)

For \( i = 1 \ldots n \),

\[ w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid x^j, w)] \]

repeat
That’s all M(C)LE. How about MAP?

\[ p(w | Y, X) \propto P(Y | X, w)p(w) \]

- One common approach is to define priors on \( w \)
  - Normal distribution, zero mean, identity covariance
  - “Pushes” parameters towards zero
- Corresponds to Regularization
  - Helps avoid very large weights and overfitting
  - More on this later in the semester

- MAP estimate

\[
w^* = \arg \max_w \ln \left( p(w) \prod_{j=1}^{N} P(y_j \mid x_j, w) \right)\]

### M(C)AP as Regularization

\[
\ln \left( p(w) \prod_{j=1}^{N} P(y_j \mid x_j, w) \right) \quad p(w) = \prod_{i} \frac{1}{\kappa \sqrt{2\pi}} e^{-\frac{w_i^2}{2\kappa^2}}
\]

Penalizes high weights, also applicable in linear regression
Gradient of M(C)AP

\[ \frac{\partial}{\partial w_i} \ln \left( p(w) \prod_{j=1}^{N} P(y^j \mid x^j, w) \right) \]

\[ p(w) = \prod_i \frac{1}{\sqrt{2\pi}} e^{-\frac{w_i^2}{2\pi}} \]

MLE vs MAP

- Maximum conditional likelihood estimate

\[ w^* = \arg \max_w \ln \left( \prod_{j=1}^{N} P(y^j \mid x^j, w) \right) \]

\[ w_i^{(t+1)} = w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid x^j, w)] \]

- Maximum conditional a posteriori estimate

\[ w^* = \arg \max_w \ln \left( p(w) \prod_{j=1}^{N} P(y^j \mid x^j, w) \right) \]

\[ w_i^{(t+1)} = w_i^{(t)} - \lambda w_i^{(t)} + \eta \left\{ -\sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid x^j, w)] \right\} \]
Naïve Bayes vs Logistic Regression

Consider $Y$ boolean, $X_i$ continuous, $X = <X_1 \ldots X_n>$

Number of parameters:
- NB: $4n + 1$
- LR: $n + 1$

Estimation method:
- NB parameter estimates are uncoupled
- LR parameter estimates are coupled

G. Naïve Bayes vs. Logistic Regression 1

[Ng & Jordan, 2002]

- Generative and Discriminative classifiers
- Asymptotic comparison (# training examples $\rightarrow$ infinity)
  - when model correct
    - GNB, LR produce identical classifiers
  - when model incorrect
    - LR is less biased – does not assume conditional independence
      - therefore LR expected to outperform GNB
G. Naïve Bayes vs. Logistic Regression 2

[Ng & Jordan, 2002]

- Generative and Discriminative classifiers

- Non-asymptotic analysis
  - Convergence rate of parameter estimates, $n = \# \text{of attributes in } X$
    - Size of training data to get close to infinite data solution
    - GNB needs $O(\log n)$ samples
    - LR needs $O(n)$ samples

- GNB converges more quickly to its (perhaps less helpful) asymptotic estimates

Some experiments from UCI data sets

Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. $m$ (averaged over 1000 random splits). Dashed line is logistic regression, solid line is naïve Bayes.
What you should know about Logistic Regression (LR)

- Gaussian Naïve Bayes with class-independent variances representationally equivalent to LR
  - Solution differs because of objective (loss) function
- In general, NB and LR make different assumptions
  - NB: Features independent given class → assumption on $P(X|Y)$
  - LR: Functional form of $P(Y|X)$, no assumption on $P(X|Y)$
- LR is a linear classifier
  - Decision rule is a hyperplane
- LR optimized by conditional likelihood
  - No closed-form solution
  - Concave → global optimum with gradient ascent
  - Maximum conditional a posteriori corresponds to regularization
- Convergence rates
  - GNB (usually) needs less data
  - LR (usually) gets to better solutions in the limit