Probabilistic Models for Classification

Binary Classification Problem

- N iid training samples: $\{x_n, c_n\}$
- Class label: $c_n \in \{0,1\}$
- Feature vector: $X \in R^d$

- Focus on modeling conditional probabilities $P(C|X)$
- Needs to be followed by a decision step

Generative models for classification

- Model joint probability
  - $P(C, X) = P(C)P(X|C)$
- Class posterior probabilities via Bayes rule
  - $P(C|X) \propto P(C, X)$

- Prior probability of a class: $P(C = k)$
- Class conditional probabilities: $P(X = x|C = k)$

Generative Process for Data

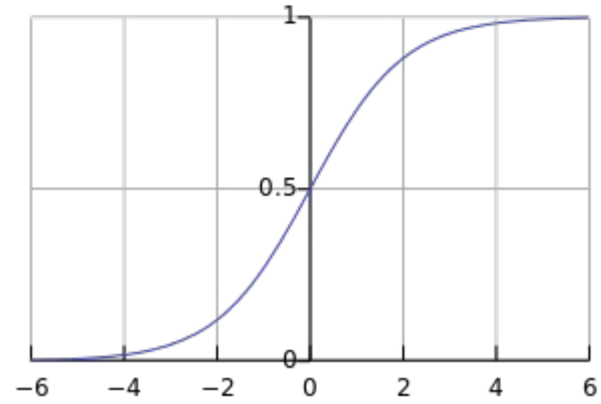
- Enables generation of new data points
- Repeat N times
  - Sample class $c_i \sim p(c)$
  - Sample feature value $x_i \sim p(x|c_i)$

Conditional Probability in a Generative Model

- $$P(C=1|x) = \frac{P(C=1)P(x|C=1)}{P(C=1)P(x|C=1) + P(C=0)P(x|C=0)} = \frac{1}{1+\exp\{-a\}} \stackrel{\text{def}}{=} \sigma(a)$$

  where $a = \ln\left(\frac{P(C=1)P(x|C=1)}{P(C=0)P(x|C=0)}\right)$

- Logistic function $\sigma()$
- Independent of specific form of class conditional probabilities



Case: Binary classification with Gaussians

- Prior class probability

$$C \sim Ber(\pi)$$
$$P(c;\pi) = \pi^c(1-\pi)^{1-c}$$

- Gaussian class densities

$$P(x|C=k) = N(\mu_k, \Sigma)$$
$$= \frac{1}{(2\pi)^{M/2}|\Sigma|^{1/2}} \exp\{(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\}$$

- Parameters $\Theta = \{\pi, \mu_0, \mu_1, \Sigma\}$
- Note: Covariance parameter is shared

Case: Binary classification with Gaussians

- $$P(C=1|x) = \sigma(w^T x + w_0)$$

  Where

  $$w = \Sigma^{-1}(\mu_1 - \mu_0)$$

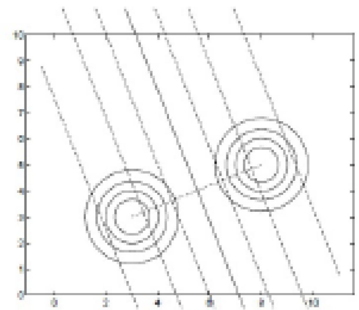
  $$w_0 = -\frac{1}{2}\mu_1{}_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + \log\frac{\pi}{1-\pi}$$

- Quadratic term cancels out

- Linear classification model
- Class boundary $w^T x + w_0 = 0$

Special Cases

- $\Sigma = I; \pi = 1 - \pi = 0.5$
  - Class boundary: $x = \frac{1}{2}(\mu_0 + \mu_1)$
- $\Sigma = I; \pi \neq 1 - \pi$
  - Class boundary shifts by $\log \frac{\pi}{1-\pi}$
- Arbitrary Σ
  - Decision boundary still linear but not orthogonal to the hyper-plane joining the two means



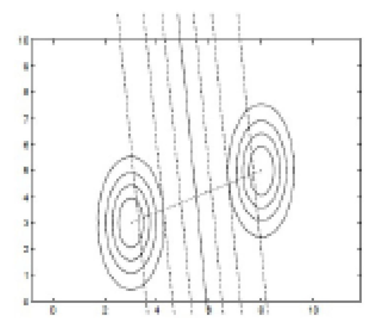


Image from Michael Jordan's book

MLE for Binary Gaussian

- Formulate loglikelihood in terms of parameters

$$l(\Theta) = \sum_i \log p(c_i) p(x_i | c_i)$$

$$= \sum_i c_i \log \pi + (1 - c_i) \log(1 - \pi)$$

$$+ c_i \log N(x_i | \mu_1, \Sigma) + (1 - c_i) \log N(x_i | \mu_0, \Sigma)$$

- Maximize loglikelihood wrt parameters

$$\frac{\partial l}{\partial \mu_1} = 0 \Rightarrow \hat{\mu}_{1_{ML}} = \frac{\sum_i c_i x_i}{\sum_i c_i}$$

$$\frac{\partial l}{\partial \pi} = 0 \Rightarrow \hat{\pi}_{ML} = \frac{\sum_i c_i}{N}$$

$$\hat{\Sigma}_{ML} = ?$$

Case: Gaussian Multi-class Classification

- $C \in \{1,2,\dots,K\}$
- Prior $P(C=k)=\pi_k; \pi_k \geq 0, \sum_k \pi_k = 0$
- Class conditional densities $P(x|C=k) = N(\mu_k, \Sigma)$

$$P(C=k|x) = \frac{\exp\{a_k\}}{\sum_l \exp\{a_l\}}$$

where $a_k = \log p(C=k)p(x|C=k)$

- Soft-max / normalized exponential function
- For Gaussian class conditionals
  - $a_k = \mu_k^T \Sigma^{-1} x - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k^T + \log \pi_k$
  - The decision boundaries are still lines in the feature space

MLE for Gaussian Multi-class

- Similar to the Binary case

Case: Naïve Bayes

- Similar to Gaussian setting, only features are discrete (binary, for simplicity)

- "Naïve" Assumption: Feature dimensions X_j conditionally independent given class label
  - Very different from independence assumption

Case: Naïve Bayes

- Class conditional probability

$$p(x|C = k; \eta) = \prod_{j=1}^{M} p(x_j|C = k; \eta) = \prod_{j=1}^{M} \eta_{kj}^{x_j}(1 - \eta_{kj})^{1-x_j}$$

- Posterior probability

$$P(C = k|x) = \frac{\exp\{a_k\}}{\sum_l \exp\{a_l\}}$$

Where $a_k = \log \pi_k + \sum_j [x_j \log \eta_{kj} + (1 - x_j) \log 1 - \eta_{kj}]$

MLE for Naïve Bayes

- Formulate loglikelihood in terms of parameters

$$\Theta = \{\pi, \eta\}$$

$$l(\Theta) = \sum_n \sum_j \sum_k c_{nk}[x_{nj} \log \eta_{kj} + (1 - x_{nj} \log(1 - \eta_{kj}))] + \sum_n \sum_k c_{nk} \log \pi_k$$

- Maximize likelihood wrt parameters

$$\widehat{\Theta}_{ML} = \arg\max \mathrm{l}(\Theta) \;\; s.t. \sum_k \pi_k = 1$$

$$\hat{\eta}_{kj_{ML}} = \frac{\sum_n x_{nj} c_{nk}}{\sum_n c_{nk}}$$

$$\hat{\pi}_{k_{ML}} = \frac{\sum_n c_{nk}}{N}$$

- MLE overfits
  - Susceptible to 0 frequencies in training data

Bayesian Estimation for Naïve Bayes

- Model the parameters as random variables and analyze posterior distributions
- Take point estimates if necessary

$$\pi \sim Beta(\alpha, \beta)$$
$$\eta_{kj} \sim iid\ Beta(\alpha_k, \beta_k)$$

$$\hat{\pi}_{k_{ML}} = \frac{\sum_n c_{nk} + \alpha - 1}{N + \alpha + \beta - 2}$$
$$\hat{\eta}_{kj_{MAP}} = \frac{\sum_n x_{nj} c_{nk} + \alpha_k - 1}{\sum_n c_{nk} + \alpha_k + \beta_k - 2}$$

Discriminative Models for Classification

- Familiar form for posterior class distribution

$$P(C = k|x) = \frac{\exp\{w_k^T x + w_0\}}{\sum_l \exp\{w_l^T x + w_0\}}$$

- Model posterior distribution directly
- Advantages as classification model
  - Fewer assumptions, fewer parameters

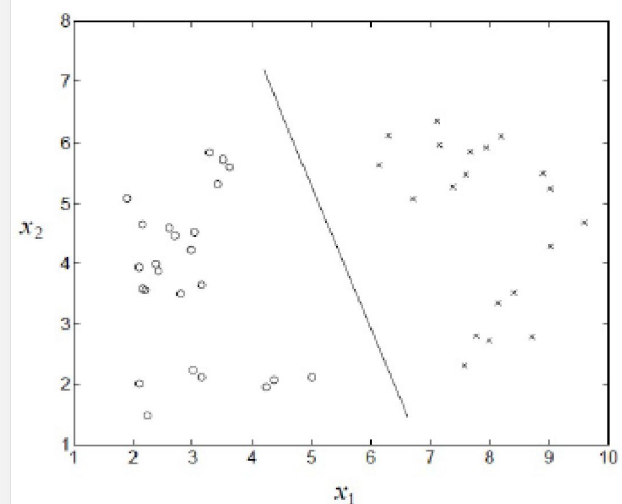


Image from Michael Jordan's book

Logistic Regression for Binary Classification

- Apply model for binary setting

$$\mu(x) \equiv P(C = 1|x) = \frac{1}{1 + \exp\{-w^T x\}}$$

- Formulate likelihood with weights as parameters

$$L(w) = \prod_n \mu(x_n)^{c_n} \left(1 - \mu(x_n)\right)^{1-c_n}$$

$$l(w) = \sum_n c_n \log \mu + (1 - c_n) \log(1 - \mu)$$

$$\text{where } \mu = \frac{1}{1+\exp\{-w^T x_n\}}$$

MLE for Binary Logistic Regression

- Maximize likelihood wrt weights

$$\frac{\partial l(w)}{\partial w} = X^T(c - \mu)$$

- No closed form solution

MLE for Binary Logistic Regression

- Not quadratic but still convex
- Iterative optimization using gradient descent (LMS algorithm)
- Batch gradient update
  - $w^{(t+1)} = w^t + \rho \sum_n x_n (c_n - \mu(x_n))$
- Stochastic gradient descent update
  - $w^{(t+1)} = w^t + \rho x_n (c_n - \mu(x_n))$
- Faster algorithm – Newton's Method
  - Iterative Re-weighted least squares (IRLS)

Bayesian Binary Logistic Regression

- Bayesian model exists, but intractable
  - Conjugacy breaks down because of the sigmoid function
  - Laplace approximation for the posterior

- Major challenge for Bayesian framework

Soft-max regression for Multi-class Classification

- Left as exercise

Choices for the activation function

- Probit function: CDF of the Gaussian
- Complementary log-log model: CDF of exponential

Generative vs Discriminative: Summary

- Generative models
  - Easy parameter estimation
  - Require more parameters OR simplifying assumptions
  - Models and "understands" each class
  - Easy to accommodate unlabeled data
  - Poorly calibrated probabilities

- Discriminative models
  - Complicated estimation problem
  - Fewer parameters and fewer assumptions
  - No understanding of individual classes
  - Difficult to accommodate unlabeled data
  - Better calibrated probabilities

Decision Theory

- From posterior distributions to actions
- Loss functions measure extent of error
- Optimal action depends on loss function

- Reject option for classification problems

Loss functions

- 0-1 loss
  - $L(y, a) = I(y \neq a) = \begin{matrix} 0 \; if \; a = y \\ 1 \; if \; a \neq y \end{matrix}$
  - Minimized by MAP estimate (posterior mode)

- l_2 loss
  - $L(y, a) = (y - a)^2$
  - Expected loss: $E[(y-a)^2|x]$ (Min mean squared error)
  - Minimized by Bayes estimate (posterior mean)

- l_1 loss
  - $L(y, a) = |y - a|$

    Minimized by posterior median

Evaluation of Binary Classification Models

- Consider class conditional distribution $P(C|X)$
- Decision rule: $C = 1\ if\ P(C|X) > \ t$

- Confusion Matrix

ROC curves

ROC curves

- Plot TPR and FPR for different values of decision threshold
- Quality of classifier measured by area under the curve (AUC)

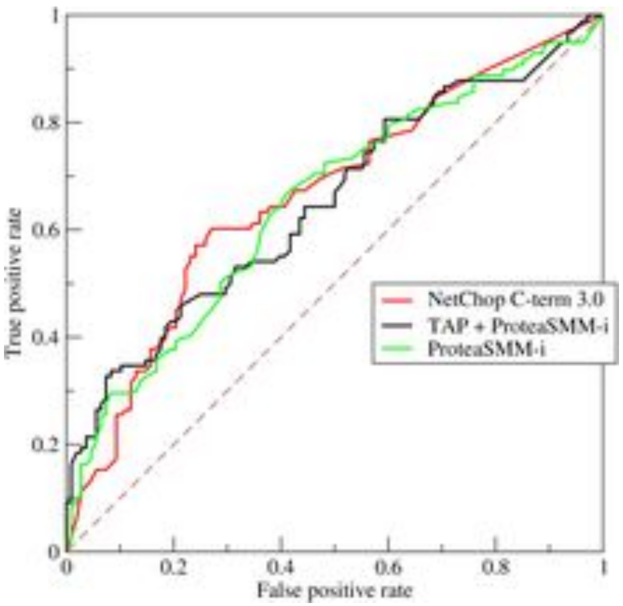


Image from wikipedia

Precision-recall curves

- In settings such as information retrieval, $N_- \gg N_+$
- Precision = $\frac{TP}{\widehat{N}_+}$
- Recall = $\frac{TP}{N_+}$
- Plot precision vs recall for varying values of threshold
- Quality of classifier measured by area under the curve (AUC) or by specific values e.g. P@k

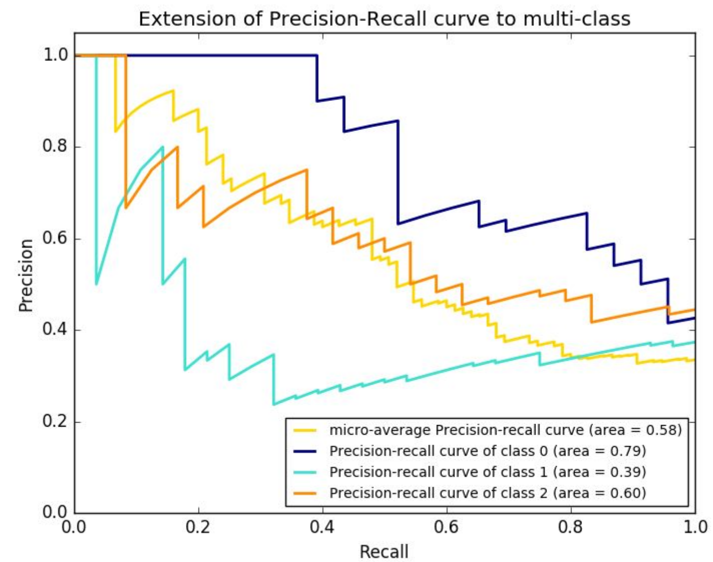


Image from scikit-learn

F1-scores

- To evaluate at a single threshold, need to combine precision and recall
- $F1 = \frac{2PR}{P+R}$
- $F_\beta = \frac{(1+\beta^2)P.R}{\beta^2 P+R}$ when P and R and not equally important
- Harmonic mean
  - Why?

Estimating generalization error

- Training set performance is not a good indicator of generalization error
  - A more complex model overfits, a less complex one underfits
  - Which model do I select?

- Validation set
  - Typically 80%, 20%
  - Wastes valuable labeled data

- Cross validation
  - Split training data into K folds
  - For i^{th} iteration, train on K/i folds, test on i^{th} fold
  - Average generalization error over all folds
  - Leave one out cross validation: K=N

Summary

- Generative models
  - Gaussian Discriminant Analysis
  - Naïve Bayes
- Discriminative models
  - Logistics regression
  - Iterative algorithms for training
- Binary vs Multiclass
- Evaluation of classification models
- Generalization performance
  - Cross validation