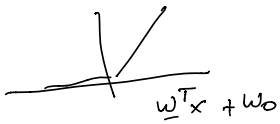
CS 181 | Prob Classification | Sprng '21

Recall $\hat{y}_n = \begin{cases} +1 & \text{if } \underline{w}^T \underline{x} + w_0 > 0 \\ -1 & \text{otherwise} \end{cases}$

train via hinge loss



(neg. example)

But, what if we'd like a probability that an example is positive?

Approach 1: Discriminative

MLE

$$\arg\max_{\underline{w}} \prod_n p(y_n | \underline{x}_n, \underline{w})$$

conditional likelihood

✓ simple

→ Logistic regression

Approach 2: Generative

MLE

$$\arg\max_{\underline{w}} \prod_n p(\underline{x}_n, y_n | \underline{w})$$

joint likelihood

✓ flexible ✓ missing labels
✓ add knowledge

→ multivariate Gaussian

→ naive Bayes

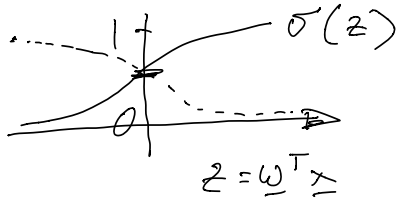
Note ① Convenient to adopt $\{0, 1\}$

② $\hat{y} = \begin{cases} 1 & \text{if } p(y=1|\underline{x}) > p(y=0|\underline{x}) \\ 0 & \text{otherwise} \end{cases}$

(1) Discriminative (Logistic regression)

Model $p(y|\underline{x})$ through a sigmoid (logistic)

$$p(y=1|\underline{x}) = \frac{1}{1+\exp(-\underline{w}^T\underline{x})}$$



Write shorthand $h = \underline{w}^T\underline{x}$.

Note $p(y=1|\underline{x}) = \frac{1}{1+e^{-h}}$ $\quad p(y=0|\underline{x}) = \frac{1}{1+e^{h}}$

Since $y \in \{0,1\}$, can write

$$p(y|\underline{x}) = p(y=1|\underline{x})^y \, p(y=0|\underline{x})^{(1-y)}$$

MLE Loss is negative log likelihood

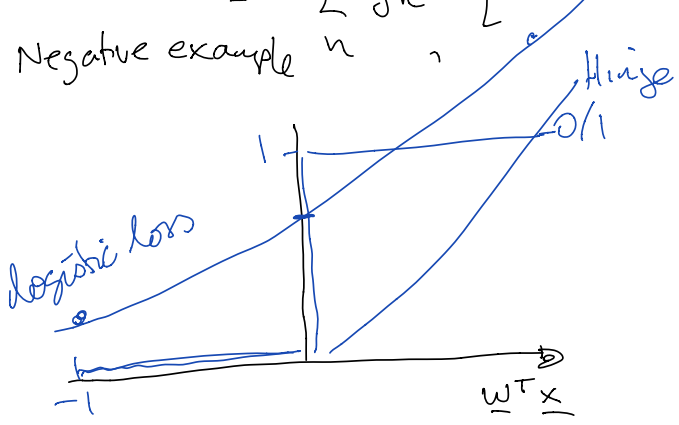
$$L_{ll}(\underline{w}) = -\sum_n \ln\left[p(y_n|\underline{x}_n)\right]$$

$$= -\sum_n y_n \ln\left[\sigma(h_n)\right] - \sum_n (1-y_n)\ln\left[\sigma(-h_n)\right]$$

$$= \sum_n y_n \ln\left[1+\exp(-h_n)\right] + \sum_n (1-y_n)\ln\left[1+\exp(h_n)\right]$$

Negative example n,

loss on a -ve ex



Differentiable & convex

Prefers to make better decisions on correct predictions

Stochastic gradient descent

	Positive example	Negative example
Hinge:	If $\hat{y}_n \neq y_n$:	If $\hat{y}_n \neq y_n$
	$\underline{w}_{t+1} \leftarrow \underline{w}_t + \eta \underline{x}_n$	$\underline{w}_{t+1} \leftarrow \underline{w}_t - \eta \underline{x}_n$
Logistic:	$\underline{w}_{t+1} \leftarrow \underline{w}_t + \eta \underline{x}_n p(y_n = 0 \mid \underline{x}_n)$	$\underline{w}_{t+1} \leftarrow \underline{w}_t + \eta \underline{x}_n p(y_n = 1 \mid \underline{x}_n)$

Algorithms come from loss functions

Loss + SGD $\rightarrow$ Algorithm

$$\underline{w}_{t+1} \leftarrow \underline{w}_t - \eta \frac{\partial L(\underline{w})}{\partial(\underline{w})}$$ SGD

[2] Generative approach

Model $p(\underline{x}, y)$ directly

(1) Decide on a generating process

(2) Parametric model

(3) Minimize & negative log. likelihood

Use the chain (product) rule:

$$p(\underline{x}, y) = p(\underline{x}|y)\,p(y)$$

$y \rightarrow \underline{x}$

class conditional — continuous — Gaussians — closed — Naive Bayes

class prior — Bernoulli

MLE $\arg\max_{\underline{w}} \prod_n p(\underline{x}_n, y_n) = \arg\max_{\underline{w}} \prod_n p(\underline{x}_n|y_n)\,p(y_n)$

$$= \arg\min_{\underline{w}} -\sum_n \ln\left[p(\underline{x}_n|y_n)\right] - \sum_n \ln\left[p(y_n)\right]$$

solve for parts of $\underline{w}^*$ corresponding to class conditional

solve for part of $\underline{w}^*$ corresponding to class prior

Aside Classify via Bayes rule

$$p(y=1 \mid \underline{x}) \propto p(y=1)\, p(\underline{x} \mid y=1)$$

Estimate Class prior

Model as Bernoulli r.v. with prob θ

$$p(y) = \theta^{y}(1-\theta)^{(1-y)}$$

Set $\frac{\partial}{\partial \theta} L(\theta) = 0$

MLE: $\theta^{*} = \sum_{n} y_n / N$

$$L(\theta) = -\sum_{n} y_n \log(\theta) - \sum_{n} (1-y_n) \log(1-\theta)$$

⋮

solve analytically

Class conditional (continuous)

$p(\underline{x}|y)$ $\underline{x}$ continuous

$\underline{x}|y=0 \sim N(\underline{\mu}_0, \Sigma_0)$ $\quad \underline{x}|y=1 \sim N(\underline{\mu}_1, \Sigma_1)$

$\underline{w} = \{\underline{\mu}_0, \Sigma_0, \underline{\mu}_1, \Sigma_1\}$ $\quad (+\theta)$

Intuition: MLE seperates by class

For class $y=0$, use

$$\{\underline{x} : (\underline{x}_n, y_n) \in D, y_n = 0\}$$

↳ estimate $\underline{\mu}_0, \Sigma_0$

For class $y=1$, use

$$\{\underline{x} : (\underline{x}_n, y_n) \in D, y_n = 1\}$$

↳ estimate $\underline{\mu}_1, \Sigma_1$

Example

MLE $\underline{\mu}_0 = \sum_{n: y=0} \underline{x}_n \Big/ N_0$ ← # examples with $y=0$

↳ Understand decision boundaries!

Note: Same covariance $\Sigma_1 = \Sigma_2$ then linear decision boundaries, otherwise get quadratic boundaries.

Classify $\hat{y} = \begin{cases} 1 & \text{if } p(y=1)p(\underline{x}|y=1) > p(y=0)p(\underline{x}|y=0) \\ 0 & \text{otherwise} \end{cases}$

Boundary:

$$S = \left\{ \underline{x} : \frac{p(\underline{x}|y=1)}{p(\underline{x}|y=0)} = \text{constant} \right\}$$

(1) Equivalently, decision boundary S:

$$\ln\left[p(\underline{x}|y=1)\right] - \ln\left[p(\underline{x}|y=0)\right] = \text{constant}$$

Plug in

$$N(\underline{x}|\underline{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \exp\left[-\frac{1}{2}(\underline{x}-\underline{\mu})^T \Sigma^{-1} (\underline{x}-\underline{\mu})\right]$$

& algebra

(see section notes)

Class conditional (discrete) "Naive Bayes"

Now x_d takes on one of $\{1, \ldots, J\}$ discrete values. e.g., hair color

Key assumption (Naive Bayes)

Each dimension of $\underline{x}$ is independent, conditioned on the class

$$p(\underline{x} \mid y=1) = \prod_d p(x_d \mid y=1)$$

For feature x_d, model as a categorical distribution.

Notation

$\pi_{kdj} \geq 0$

Probability in class k of feature d taking on value j

$\sum_j \pi_{kdj} = 1$ for all k, all d

Write x_d as a "one-hot" vector

$x_d = [1 \; 0 \; 0]^T$ for blond hair

Can write

$$p(\underline{x} \mid y=1) = \prod_d \prod_j \pi_{1dj}^{x_{dj}}$$

raise to power x_{dj}, where x_{dj} is 0 or 1

One-hot $x_{dj} = \begin{cases} 1 & \text{if } d^{th} \text{ feature has } j^{th} \text{ value} \\ 0 & \text{otherwise} \end{cases}$

Then find parameters $\underline{\pi}_0$ of class 0 and parameters $\underline{\pi}_1$ of class 1 to minimize negated log likelihood

$$\underset{\underline{\pi}_0, \underline{\pi}_1}{\arg\min} \; -\sum_n \ln[p(\underline{x}_n \mid y_n)]$$

Can write $\underline{\pi}_0$ as a "stacked" $(D \times J)$-dim vector; write $\underline{x}$ as a "stacked" $(D \times J)$-dim vector.

MLE fit

$$\underline{\pi}_0 = \sum_{n: y_n = 0} \underline{x}_n / N_0 \quad (\#\text{ examples with } y_n = 0)$$

$$\underline{\pi}_1 = \sum_{n: y_n = 1} \underline{x}_n / N_1$$

Discriminative

Separate parameters $\underline{w}_k$ for each class

$\exp[\underline{w}_k^T \underline{x}]$

$\sum_k \exp[\underline{w}_k^T \underline{x}]$

"softmax"

sum to 1

- ✓ train very quickly
- ✓ interpretable
- ✓ good on small data sets

Decision boundary From (1) this is x s.t.

$$\ln[p(x|y=1)] - \ln[p(x|y=0)] = \text{constant}$$

For categorical model, this is

$$\ln\left[\prod_d \prod_j \pi_{1dj}^{x_{dj}}\right] - \ln\left[\prod_d \prod_j \pi_{0dj}^{x_{dj}}\right] = \text{constant}$$

$$\Leftrightarrow \sum_d \sum_j x_{dj} \ln\left(\frac{\pi_{1dj}}{\pi_{0dj}}\right) = \text{constant}$$

$$\Leftarrow \quad \underline{x}^T \ln\left[\frac{\underline{\pi}_1}{\underline{\pi}_2}\right] = \text{constant}$$

(writing these as stacked $D \times J$ - dim vectors)

Conclude that decision boundary of categorical classification model is linear ∎

Note! Multiclass generalizations

$C = \{C_1, \ldots, C_K\}$

Generatively → easy, just use a categorical class prior (not Bernoulli) and estimate class conditionals in same way

Classify as $\arg\max_k p(\underline{x} | y = C_k) p(y = C_k)$

Discriminative

Now have seperate parameters $\underline{w}_k$ for each class

$$p(y = C_k | \underline{x}) = \frac{\exp\left[\underline{w}_k^T \underline{x}\right]}{\sum_{\ell=1}^{K} \exp\left[\underline{w}_\ell^T \underline{x}\right]}$$

"softmax" sums to 1

"Multiclass Logistic regression"