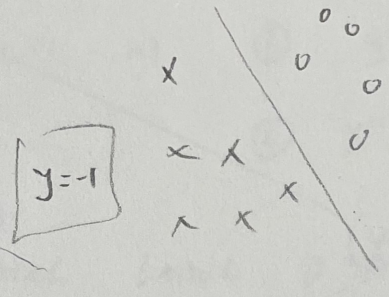
Lecture 02 Recap

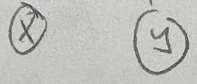
Linear Decision Boundary Learners:



INFERENCE:

$$\hat{y} = \text{Sign}(xw^T + b)$$

$P_D(x, y)$ ← unobserved, intractable to learn directly!

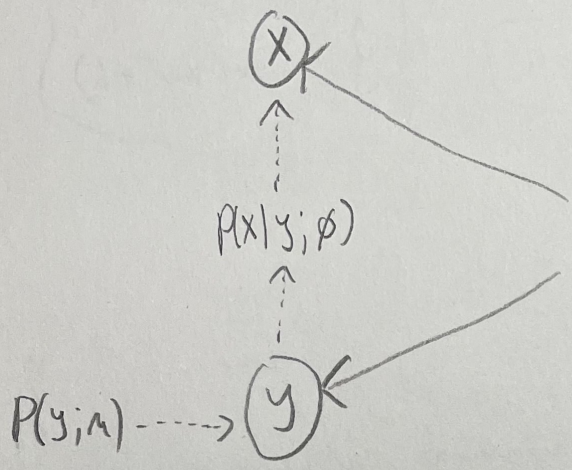


Statistical machine learning is a general class of problems of the form:

Given some input x, predict the most likely output y using the information from M independent draws from $x^{(i)}, y^{(i)} \sim P_D(x, y)$

STATISTICAL ML TAXONOMY

GENERATIVE MODELING



PARAMETER ESTIMATION

a) $\hat{M} = \underset{M}{\text{ARGMIN}} \sum_{i=1}^{m} -\log P(y^{(i)};M)$

b) $\hat{\phi} = \underset{\phi}{\text{ARGMIN}} \sum_{i=1}^{m} -\log P(x^{(i)}|y^{(i)};\phi)$

INFERENCE:

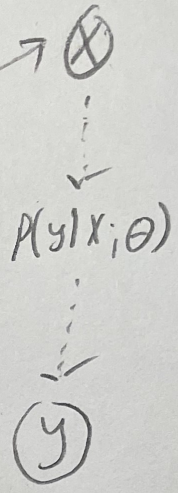
$P(y|x;\hat{M},\hat{\phi}) \propto P(x|y;\hat{\phi})\,P(y;\hat{M})$

↳ BAYES' RULE

[redacted]

$\hat{y} = \underset{y}{\text{ARGMAX}} \log P(x|y;\hat{\phi}) + \log P(y;\hat{M})$

DISCRIMINATIVE MODELING



PARAMETER ESTIMATION

a) $\hat{\theta} = \underset{\theta}{\text{ARGMIN}} \sum_{i=1}^{m} -\log P(y^{(i)}|x^{(i)};\theta)$

INFERENCE:

$\hat{y} = \underset{y}{\text{ARGMAX}}\ P(y|x;\hat{\theta})$

GENERATIVE MODEL: NAIVE BAYES

THE DATA GENERATING PROCESS:

$\forall i \in \{1, \dots, M\}$ ① $y^{(i)} \sim$ CATEGORICAL (π) LABEL

② $x^{(i)} \sim$ MULTINOMIAL (ϕ) FEATURE

① CATEGORICAL LABEL DISTRIBUTION

$p(y; \pi) = [\pi_1, \dots, \pi_K]$, $\sum_{k=1}^{K} \pi_k = 1$

↳ ARBITRARY DISTRIBUTION OVER K CLASSES

② MULTINOMIAL FEATURE DISTRIBUTION

$X =$ "CATS RULE, DOGS DROOL" ... SEQUENCE LENGTH: $= T = 4$

$p(x|y;\phi)$: MODELS T-FLIPS OF AN N-SIDED COIN

$= p(x_1, \dots, x_n | y; \phi)$ → INTRACTABLE

$= p(x_1|y) \cdot p(x_2|x_1, y) \cdot \dots, p(x_n|x_1, \dots, x_{n-1}, y)$ ⇒ INTRACTABLE

⇒ NAIVE BAYES ASSUMPTION: WORD COUNTS ARE INDEPENDENT GIVEN A CLASS LABEL y.

$= p(x_1|y) \cdot p(x_2|y) \cdot \dots \cdot p(x_n|y)$ ⇒ TRACTABLE!

NAIVE BAYES CONT...

$p(x \mid y; \phi)$ is a MULTINOMIAL DISTRIBUTION

$$= B(x) \prod_{j=1}^{N} \phi_{kj}^{x_j}, \quad B(x) = \frac{\left(\sum_{j=1}^{N} x_j\right)!}{\prod_{j=1}^{N} x_j!}$$

Where $\phi \in [0,1]^{K \times N}$, $\forall_k \sum_{j=1}^{N} \phi_{kj} = 1$

NAIVE BAYES: M ESTIMATION

$$\hat{\mu} = \underset{\mu}{\text{ARGMAX}} \underbrace{\sum_{i=1}^{m} \log \mu_{y^{(i)}}}_{\ell(\mu)} \quad \text{S.T.} \quad \underbrace{\sum_{k=1}^{K} \mu_k = 1}_{C(\mu)}$$

↳ Solve using method of Lagrange multipliers: $\forall_k \nabla_{\mu_k} \ell(\mu) = \lambda \nabla_{\mu_k} C(\mu)$

LHS $\forall_k \nabla_{\mu_k} \left[\sum_{i=1}^{m} \log \mu_{y^{(i)}} \right] = \sum_{i=1}^{m} \delta(y^{(i)} = k) \cdot \frac{1}{\mu_k} = \frac{M_k}{\mu_k}$

RHS $\forall_k \lambda \nabla_{\mu_k} \left[\sum_{k'=1}^{K} \mu_{k'} \right] = \lambda$

↳ $\forall_k \quad \mu_k = \frac{M_k}{\lambda}$ → Solve for λ

$$\sum_{k=1}^{K} \mu_k = \frac{1}{\lambda} \sum_{k=1}^{K} M_k \iff 1 = \frac{M}{\lambda} \implies \lambda = M$$

$$\therefore \quad \forall_{k \in \{1,\dots,K\}} \quad \hat{\mu}_k = \frac{\sum_{i=1}^{m} \delta(y^{(i)} = k)}{M}$$

↳ LABEL DISTRIBUTION IS THE RELATIVE FREQUENCY OF EACH CLASS IN THE DATA

NAIVE BAYES: ϕ ESTIMATION

$$\hat{\phi} = \underset{\phi}{\text{ARGMAX}} \sum_{i=1}^{M} \log B(x^{(i)}) + \sum_{j=1}^{N} x_j^{(i)} \log \phi_{y^{(i)}j} \qquad \ell(\phi)$$

$$\text{s.t.} \quad \forall_k \sum_{j=1}^{N} \phi_{kj} = 1 \qquad c(\phi)$$

↳ METHOD of LAGRANGE MULTIPLIERS: $\forall_k \nabla_{\phi_{kj}} \ell(\phi) = \lambda \nabla_{\phi_{kj}} c(\phi)$

RHS: $\forall_k \lambda \nabla_{\phi_{kj}} \left(\sum_{j'=1}^{N} \phi_{kj'} \right) = \lambda$

LHS: $\forall_k \nabla_{\phi_{kj}} \left(\sum_{i=1}^{M} \delta(y^{(i)}=k) \sum_{j'=1}^{N} x_{j'}^{(i)} \log \phi_{kj} \right) = \sum_{i=1}^{M} \delta(y^{(i)}=k) \cdot \frac{x_j^{(i)}}{\phi_{kj}}$

↳ SOLVE for λ:

$$\forall_{k,j} \quad \lambda = \sum_{i=1}^{M} \delta(y^{(i)}=k) \frac{x_j^{(i)}}{\phi_{kj}}$$

$$\forall_k \quad \lambda \sum_{j=1}^{N} \phi_{kj} = \sum_{i=1}^{M} \delta(y^{(i)}=k) \sum_{j=1}^{N} x_j^{(i)}$$

$$\forall_k \quad \lambda = \sum_{i=1}^{M} \delta(y^{(i)}=k) \sum_{j=1}^{N} x_j^{(i)}$$

λ_k = TOTAL NUMBER of WORDS IN DOCUMENTS WITH LABEL $y=k$

↳ SOLVE for $\hat{\phi}$:

$$\forall_{k,j} \quad \sum_{i=1}^{M} \delta(y^{(i)}=k) x_j^{(i)} = \lambda \phi_{kj}$$

$$\therefore \hat{\phi}_{kj} = \frac{\sum_{i=1}^{M} \delta(y^{(i)}=k) x_j^{(i)}}{\sum_{i=1}^{M} \delta(y^{(i)}=k) \sum_{j'=1}^{N} x_{j'}^{(i)}} = \frac{\text{COUNT}(y=k, x=j)}{\text{COUNT}(y=k, \text{ALL WORDS})}$$

NAIVE BAYES INFERENCE

WE USE BAYES' RULE TO COMPUTE $P(y|x)$:

$$P(y|x) \propto P(x|y;\hat{\phi})P(y;\hat{\lambda})$$

$$\hat{y} = \underset{y}{\text{ARGMAX}} \log P(x|y;\hat{\phi}) + \log P(y;\hat{\mu})$$

$$\hat{y} = \underset{k}{\text{ARGMAX}} \sum_{j=1}^{N} x_j \log \hat{\phi}_{kj} + \log \hat{\mu}_k$$

↳ NOTE: THIS IS A LINEAR MODEL!

↳ $\hat{y} = \underset{k}{\text{ARGMAX}} \ \vec{x} (\log \phi_k)^T + \log \mu_k$

SMOOTHING: MANY WORDS WILL NEVER APPEAR IN DOCUMENTS OF A GIVEN CLASS, AND THEREFORE MLE OF ϕ WILL BE ZERO IN REAL WORLD PROBLEMS. THIS IS PROBLEMATIC BECAUSE $\log(0)$ IS UNDEFINED.

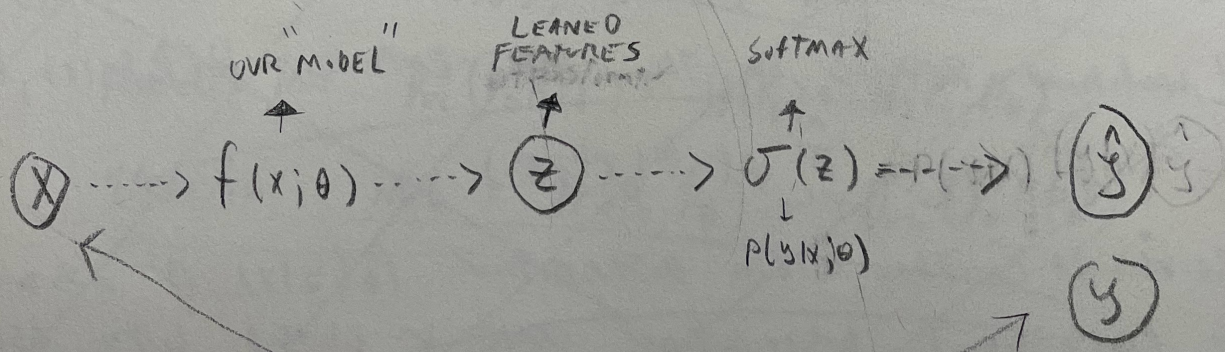
LAPLACE SMOOTHING: $$\hat{\phi}_{kj} = \frac{\alpha + \text{COUNT}(y=k, x=j)}{\alpha N + \text{COUNT}(y=k, \text{all } j)}$$

WHERE $\alpha \in \mathbb{R}$, $\alpha \geq 0$

Discriminative Learning

Our approach to discriminative learning is most often done by transforming our feature space onto our label space, converting those transformed features into a probability distribution over the class labels, and performing gradient descent to learn a transformation that yields high likelihood $P(D;\theta)$. This encompasses nearly all of deep learning.

The high level machinery is:



$x \in \mathbb{R}^N$ or $\mathbb{Z}^N$

$y \in \{0,1\}^K, \sum_{k=1}^{K} y_k = 1$

$z \in \mathbb{R}^K$

$\sigma(z) \in [0,1]^K, \sum_{k=1}^{N} \sigma(z_k) = 1$

$P_D(X,Y)$

Data distribution

Softmax

$$\sigma(z) = \frac{e^z}{\sum_{z'} e^{z'}}$$

Discriminative Learning: $\hat{\theta}$ Estimation

Negative Log-Likelihood:

$$NLL = -\sum_{i=1}^{M} \sum_{i=1}^{N} \log p(y^{(i)} | x^{(i)}; \theta)$$

Remember that our empirical label distribution is a one-hot encoded vector with all probability mass placed on the ground truth label

$$NLL = -\sum_{i=1}^{m} \sum_{k=1}^{K} P_D(y^{(i)}=k | x^{(i)}) \log p(y^{(i)}=k | x^{(i)}; \theta)$$

$$= -\sum_{i=1}^{m} E_{y^{(i)} \sim P_D(y|x)} [\log p(y^{(i)} | x^{(i)}; \theta)] \rightarrow \text{(cross entropy } H(P_D, P_\theta))$$

[redacted]

Recall that $H(P_D) = \emptyset$ by definition (one-hot encoding)

Recall that $D_{KL}(P_D \| P_\theta) = H(P_D) + H(P_D, P_\theta)$

Therefore $NLL = H(P_D, P_\theta) = D_{KL}(P_D \| P_\theta)$!

Softmax Regression

$(X) \dashrightarrow f(x;\theta) \dashrightarrow (z) \rightarrow \sigma(z)$

$$z = f(x;\theta) = xW^T + b \quad \Rightarrow \quad \theta = \{W, b\}$$

where $W \in \mathbb{R}^{K \times N}$, $b \in \mathbb{R}^K$

] our model

$$\sigma(z) = \frac{e^z}{\sum_{z'} e^{z'}} = \frac{e^{xW_K^T + b_K}}{\sum_{K'=1}^{K} e^{xW_{K'}^T + b_{K'}}}$$

] softmax aka $P(y|x;\theta)$

$$\hat{y} = \underset{K}{\text{ARGMAX}}\ \sigma(z)$$

] decision rule

Parameter Estimation

$$NLL = -\sum_{i=1}^{m} \sum_{K=1}^{K} P_0(y_K^{(i)} | x^{(i)}) \log P(y^{(i)} | x^{(i)}; \theta)$$

$$= -\sum_{i=1}^{m} \sum_{K=1}^{K} y_K^{(i)} \log P(y_K^{(i)} | x^{(i)}; \theta)$$

$$= -\sum_{i=1}^{m} \sum_{K=1}^{K} y_K^{(i)} \log \left[\frac{e^{x^{(i)} W_K^T + b_K}}{\sum_{K'=1}^{K} e^{x^{(i)} W_{K'}^T + b_{K'}}} \right]$$

$$= -\sum_{i=1}^{m} \left[\sum_{K=1}^{K} y_K^{(i)} (x^{(i)} W_K^T + b_K) - y_K^{(i)} \log \left[\sum_{K'=1}^{K} e^{x^{(i)} W_{K'}^T + b_{K'}} \right] \right]$$

$$= -\sum_{i=1}^{m} \left[\left(\sum_{K=1}^{K} y_K^{(i)} (x^{(i)} W_K^T + b_K) \right) - \log \left[\sum_{K=1}^{K} e^{x^{(i)} W_{K'}^T + b_K} \right] \right]$$

Softmax Regression Continued

WE COMPUTE $\hat{\theta}$ USING GRADIENT DESCENT

STEP 1: COMPUTE $\nabla_\theta NLL(\theta | D)$

STEP 2: UPDATE θ: $\theta \leftarrow \theta - \eta \nabla_\theta$, $\eta \in +\mathbb{R}$

$\hat{W}$ ESTIMATION

$$\nabla_{W_k} NLL(W, b | D) \in \mathbb{R}^N$$

$$= -\sum_{i=1}^{M} y_k^{(i)} x^{(i)} - x^{(i)} \cdot \frac{e^{x^{(i)} W_k^T + b_k}}{\sum_{k'=1}^{K} e^{x^{(i)} W_{k'}^T + b_{k'}}}$$

$$= \sum_{i=1}^{M} x^{(i)} \left(P(y_k^{(i)} | x^{(i)}; \theta) - y_k^{(i)} \right)$$

$\hat{b}$ ESTIMATION

$$\nabla_{b_k} NLL(W, b | D) \in \mathbb{R}$$

$$= -\sum_{i=1}^{M} y_k^{(i)} - \frac{e^{x^{(i)} W_k^T + b_k}}{\sum_{k'=1}^{K} e^{x_i W_{k'}^T + b_{k'}}}$$

$$= \sum_{i=1}^{M} P(y_k^{(i)} | x^{(i)}; \theta) - y_k^{(i)}$$

GRADIENT DESCENT VERSIONS

"VANILLA": $\nabla_\theta \sum_{i=1}^{M} -\log P(y^{(i)} | x^{(i)}; \theta)$

"STOCHASTIC": $\nabla_\theta -\log P(y^{(i)} | x^{(i)}; \theta)$

"MINI-BATCH": $\nabla_\theta \sum_{i=1}^{M'} -\log P(y^{(i)} | x^{(i)}; \theta)$

where $M' \ll M$

↳ IN PRACTICE we typically USE MINI-BATCH GRADIENT DESCENT.