Supervised Learning learn model using collection of training e.g.s.

Regression predict a scalar value

Classification predict a discrete class target

K-NEAREST NEIGHBOURS non-parametric

Classify a new input x based on its k nearest neighbours

Decision Boundary: boundary between regions of input space associated w/ diff. categories

Hyperparameter k controls # of neighbours

- ↓ good at capturing fine-grains, may overfit
- ↑ stable predictions, may underfit

Curse of Dimensionality ↑dimensionality, # of data points needed for good performance ↑ exponentially

Sensitive to data scales ⟶ just normalize stuff

More memory/computation cost

DECISION TREES parametric

Make predictions by splitting features according to a tree

Decision boundary: axis aligned planes

Greedy algorithm based on picking features that maximize information gain

ENTROPY

$$H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y)$$

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x) H(Y|X=x)$$

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X=x) = -\sum_{x} \sum_{y} p(x,y) \log_2 p(y|x)$$

$$IG(Y|X) = H(Y) - H(Y|X)$$

LINEAR REGRESSION

Linear functions of features $y = w^T x + b$

b often added into w by padding x with 1's

Loss function $\mathcal{L}(y,t)$ used for learning

Cost function $J(w,b) = \mathcal{L}(y,t)$ averaged

Can vectorize inputs to abouse GPU

Minimize avg. training error via direct soln, GD methods

Feature mapping to put features on a diff. linearly separable space

GRADIENT DESCENT

$$w_j \leftarrow w_j - \alpha \frac{\partial J}{\partial w_j}, \quad w \leftarrow w - \frac{\alpha}{N} \sum_{i=1}^{N} (y^{(i)} - t^{(i)}) x^{(i)}$$

Stochastic update weights using just one training example

Mini batch split dataset into mini-batches, average each batch

If **convex**, all critical points are global optimum

↳ $f((1-\lambda)x_0 + \lambda x_1) \leq (1-\lambda) f(x_0) + \lambda f(x_1)$

RL $Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{k \geq 0} \gamma^k R_{t+k} \mid s,a\right] = r(s,a) + \gamma \sum_{s'} P(s'|a,s) \sum_{a'} \pi(a'|s') Q^{\pi}(s',a')$

$R_t + \gamma G_{t+1}$

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^*(s',a')$$

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t|s] = \sum_{a} \pi(a|s) \left[r(s,a) + \gamma \sum_{s'} P(s'|a,s) V^{\pi}(s') \right]$$

REGULARIZATION

Adding penalties (e.g. L^2 penalty $R(w) = \frac{1}{2}\|w\|_2^2$

↳ $J_{reg}(w) = J(w) + \lambda R(w)$, so $w \leftarrow (1 - \alpha\lambda) w - \alpha \frac{\partial J}{\partial w}$

LINEAR CLASSIFICATION

Linear functions of features $z = w^T x + b$ along w/ decision rule $y = \mathbb{I}[z \geq 0][1] + \mathbb{I}[z < 0][0]$

If data is linearly separable, we can find decision boundary

Logstic regression uses $y = \sigma(z)$

Loss function 0-1: ∇ issues, squared loss: blows up extremes

logistic + SE: makes extreme misclassifications look normal

Binary Cross Entropy $\mathcal{L}_{CE} = -t \log y - (1-t) \log(1-y)$

Closed-form solution

MULTI-CLASS CLASSIFICATION

Targets belong to discrete set, one hot encoding $\in R^K$

$z = Wx + b$, $y_k = \text{Softmax}(z_1, \ldots, z_K) = \frac{e^{z_k}}{\sum_{k'} e^{z_{k'}}}$

Loss function $\mathcal{L}(y,t) = -\sum_k t_k \log y_k = -t^T (\log y)$

NEURAL NETWORKS parametric

$y = f^L \circ \cdots \circ f^1(x)$, $f^{(i)} = \phi(Wx + b)$

Hidden unit, layer, weight, activation function

Each layer is a feature detector

Expressivity universal func. approximators (non-lin. act. func.)

Regularize by early stopping

Backpropagation learning algorithm ($\bar{y} = \frac{\partial \mathcal{L}}{\partial y}$ error signal)

Convolutional NN each set of hidden units looks at local region

weights shared between all image locations

Convolution detection, filters, produces feature maps (equivariant)

Pooling reduce size of rep, build invariance, max selects max (translations not applied)

MODEL COMPLEXITY

Bias inability for method to capture true relationship, underfitting

Boosting merge different types of predictions, ↓bias

Variance difference in fits between datasets, overfitting

$$\mathbb{E}[(y-t)^2 | x] = \underbrace{(y_* - \mathbb{E}[y])^2}_{\text{bias}} + \underbrace{\text{Var}(y)}_{\text{variance}} + \underbrace{\text{Var}(t)}_{\text{Bayes error}}$$

Bagging merge same type of predictions, ↓ variance

Bayes Optimality Bayes error due to inherent unpredictability of targets, achieving Bayes error is Bayes optimal

Decide H_i if $P(H_i|z) > P(H_j|z)\ \forall i \neq j$

Hyperparameter tuned using validation set, not learned

HYPOTHESIS SPACE

Hypothesis $f: \mathcal{X} \to \mathcal{T}$ from input to target space

If datasets were not naturally biased, all ML algos. would be the same

If $\mathcal{H}_B \subseteq \mathcal{H}_A$, model A is more expressive than B

Generally, the larger the more accurate.

PROBABILISTIC MODELS

Assuming outcomes $\{x_1,\dots,x_N\}$ are iid, log-likelihood of Bernoulli RV is $\ell(\theta)=\log\left(\prod_{i=1}^N \theta^{x_i}(1-\theta)^{(1-x_i)}\right)$ (likelihood)

$=\sum_{i=1}^N x_i\log\theta+(1-x_i)\log(1-\theta)$

Maximum likelihood criteria pick θ s.t. $\hat\theta=\arg\max_\theta \ell(\theta)$

Find optimal solutions by finding critical points, $\frac{d\ell}{d\theta}=0$.

Discriminative classifiers feature $x\to$ class prob. $p(y\mid x)$

How do I learn classes from inputs?

Generative class $p(y)\to$ prob. of class given label $p(x\mid y)$

What does data for a class look like? Use Bayes' Rule

Naïve Bayes model models distribution of inputs for a label

Assumes features are cond. indep. given class

$$p(c\mid x)=\frac{p(x\mid c)\,p(x)}{\sum_{c'}p(c')p(x\mid c')}$$ ← prior; posterior for class

$$\ell(\theta)=\sum_{i=1}^N \log p(c^{(i)},x^{(i)})=\sum_{i=1}^N \log\left(p(x^{(i)}\mid c^{(i)})\,p(c^{(i)})\right)$$

$$=\sum_{i=1}^N \log p(c^{(i)})+\sum_{j=1}^D\sum_{i=1}^N \log p(x_j^{(i)}\mid c^{(i)})$$

labels, prior; feature x_j, posterior

Beta distribution $p(\theta;a,b)=\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\theta^{a-1}(1-\theta)^{b-1}$, $\mathbb{E}[\theta]=\frac{a}{a+b}$

Posterior distribution $p(\theta\mid D)=\frac{p(\theta)\,p(D\mid\theta)}{\int p(\theta')\,p(D\mid\theta')\,d\theta'}$

Maximum A-Posteriori $\hat\theta_{MAP}=\arg\max_\theta \log p(\theta)+\log p(D\mid\theta)$ ($p(\theta\mid D)$, $p(\theta)p(D\mid\theta)$; posterior probability)

MULTIVARIATE GAUSSIAN (normal)

(μ,Σ) uniquely define a multivariate Gaussian distribution

$$N(\mu,\Sigma)=\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

$\hat x=Sx+b$, $\mathbb{E}[\hat x]=b$, $\mathrm{Cov}[\hat x]=SS^T$

Since $\Sigma=\mathbb{E}[(x-\mu)^T(x-\mu)]$ is symmetric, $\Sigma=Q\Lambda Q^T$

$$\ell(\mu,\Sigma)=\sum_{i=1}^N -\log(2\pi)^{d/2}-\log|\Sigma|^{1/2}-\frac{1}{2}(x^{(i)}-\mu)^T\Sigma^{-1}(x^{(i)}-\mu)$$

$$\hat\mu=\frac{1}{N}\sum_{i=1}^N x^{(i)} \qquad \hat\Sigma=\frac{1}{N}(X-1\mu^T)^T(X-1\mu^T)^T$$

Linear Regression $t\mid x\sim N(w^Tx,\sigma^2)$, max log-likelihood

$$\frac{1}{N}\sum_{i=1}^N \log p(t^{(i)}\mid x^{(i)};w,b)=\dots=\text{const.}-\frac{1}{2N\sigma^2}\sum_{i=1}^N (t^{(i)}-w^Tx^{(i)})^2$$

L_2 regularization MAP w/ Gaussian prior, $w\sim N(m,S)$

$$\log p(w)=\frac{1}{2}(w-m)^TS^{-1}(w+m)+\text{const.}=-\frac{1}{2\eta}\|w\|^2+\text{const.}$$

GAUSSIAN DISCRIMINANT ANAYLSIS

Assumes $p(x\mid t)$ distributed according to multivar Gaussian.

assuming binary class. $\phi=\frac{1}{N}\sum_{i=1}^N r_1^{(i)}$ $\mu_k=\frac{\sum_{i=1}^N r_k^{(i)}\cdot x^{(i)}}{\sum_{i=1}^N r_k^{(i)}}$ $\Sigma_k=\frac{1}{\sum n_k^{(i)}}\sum_{i=1}^N r_k^{(i)}\|x^{(i)}-\mu_k\|^2$

Decision boundary

$$\log p(t_k\mid x)=-\frac{D}{2}\log(2\pi)-\frac{1}{2}\log|\Sigma_k|-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)+\log p(t_k)-\log p(x)$$

$\Sigma_k=\Sigma_\ell$ then linear boundary $(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)=(x-\mu_\ell)^T\Sigma_\ell^{-1}(x-\mu_\ell)+\text{const}$ conic

Gaussian Naïve Bayes

$$\mu_{jk}=\frac{\sum r_k^{(i)}x_j^{(i)}}{\sum r_k^{(i)}} \qquad \sigma_{jk}^2=\frac{\sum r_k^{(i)}(x_j^{(i)}-\mu_{jk})^2}{\sum r_k^{(i)}} \qquad r_k^{(i)}=\mathbb{I}[t^{(i)}=k]$$

Isotropic?

uniform $p(t)$ for simplicity. If $\Sigma=\sigma^2I$, we only need one parameter

$$\log p(t_k\mid x)-\log p(t_\ell\mid x)=-\frac{1}{2\sigma^2}\left[\|x-\mu_k\|^2-\|x-\mu_\ell\|^2\right]$$

PRINCIPAL COMPONENT ANALYSIS

Dimensionality Reduction map to lower dim. space to retain only useful info, ideally close to intrinsic dimension

PCA is unsupervised, linear dimensionality reduction, projections

$\mathrm{Proj}_S(x)=\sum_{i=1}^K z_iu_i$ where $z_i=x^Tu_i$, $\mathrm{Proj}_S(x)=Uz$ where $z=U^Tx$

$\hat x=Uz+\hat\mu$ can just shift to arbitrary origin.

Minimize reconstruction error $\min_U \frac{1}{N}\sum\|x^{(i)}-\tilde x^{(i)}\|^2$

Maximize variance of reconstructions $\max_U \frac{1}{N}\sum\|\tilde x^{(i)}-\hat\mu\|^2$

Empirical $\mathrm{Cov}(x)=\hat\Sigma=\sum_{i=1}^N (x^{(i)}-\hat\mu)(x^{(i)}-\hat\mu)^T$ symmetric, PSD

Principal components eigenvectors, optimal subspace spanned by top k.

↳ $k=1$, $\frac{1}{N}\sum\|\tilde x^{(i)}-\hat\mu\|^2=\dots=u^T\hat\Sigma u=u^TQ\Lambda Q^Tu=a^T\Lambda a=\sum_{j=1}^D\lambda_ja_j^2$

MATRIX FACTORIZATION

Assume $x^{(i)}\approx\tilde x^{(i)}=Uz^{(i)}$ centered at $\hat\mu=0$.

Squared reconstruction error $\sum_{i=1}^N\|x^{(i)}-Uz^{(i)}\|^2=\|X-ZU^T\|_F^2$

Sparse data matrix $R\approx UZ^T$, need to fill in values w/ alternating least squares

Squared error loss $\min_{U,Z}\frac{1}{2}\sum_{(i,j)\in O}(R_{ij}-u_i^Tz_j)^2=\sum_i\sum_{j:(i,j)\in O}(R_{ij}-u_i^Tz_j)^T$

Minimize wrt. u_i first, $u_i=\left(\sum_{j:(i,j)\in O}z_jz_j^T\right)^{-1}\sum_{j:(i,j)\in O}R_{ij}z_j$, then z_j

AUTOENCODER

Feed-forward NN to take input x & predict x

Encoder (hopefully removes nois), Decoder recreates data

1 hidden layer linear reduce dimensions for code vector, ^{2} loss

Nonlinear learns to map to a manifold, more powerful

PCA, matrix factor., auto. are latent variable models

Assume data depends on some never used latent variables.

K-MEANS (multimodal)

Clustering grouping data points in clusters w/ no labels

Find cluster centers m & assignments r to min. sum of ^{2} distances

$$\min_{\{m\},\{r\}} J(\{m\},\{r\})=\min_{m,r}\sum_{n=1}^N\sum_{k=1}^K r_k^{(n)}\|m_k-x^{(n)}\|^2$$

Assignment $r=\begin{cases}1 & \text{if } k=\arg\min_j\|m_j-x^{(n)}\|^2\\ 0 & \text{otherwise}\end{cases}$ Refitting $m_k=\frac{\sum_n r_k^{(n)}x^{(n)}}{\sum_n r_k^{(n)}}$

Alternating minimization, block coordinate descent

Soft $r_k^{(n)}=\mathrm{softmax}(-\beta\{\|m_k-x^{(n)}\|^2\}_{k=1}^K)$, $\beta\to\infty$ then → normal k-means

GAUSSIAN MIXTURE MODELS

$p(z)$ in $p(x)=\sum_z p(x,z)=\sum_z p(x\mid z)p(z)$ categorical → mixture model

GMM $p(x)=\sum_{k=1}^K\pi_kN(x\mid\mu_k,\Sigma_k)$, fit params. w/ max likelihood.

EXPECTATION-MAXIMIZATION

Find (local) max likelihood params. of model

$$\log p(X;\theta)=\sum_{i=1}^N\log\sum_{k=1}^K p(x^{(i)}\mid z^{(i)};\{u_k\},\{\Sigma_k\})\,p(z^{(i)}\mid\pi)$$

E-step compute post. prob. over z, given θ compute $z^{(i)}$

Assign responsibility $r_k^{(i)}=\Pr(z^{(i)}=k\mid x^{(i)};\theta)$

M-step given $z^{(i)}$ learn θ, apply max likelihood updates

Gaussian discriminant analysis