CS489: Machine Learning

Michael Noukhovitch

Winter 2017,

Notes written from Pascal Poupart's lectures.

Contents

1 Introduction

machine learning giving computers ability to learn without being explicitly programmed

A machine learns from experience E wrt to some class of tasks T and performance measure P if its performance in task T, as measured by P, improves with E

1.1 Supervised Learning

Definition. given a training set of examples $(x, f(x))$, return a hypothesis h that approximates h

Two types:

classification where output space consists of *categorical* values

regression where output space consists of *numerical* values

1.1.1 Hypothesis Space

hypothesis space set of all hypotheses H that the learner may consider

consistent if hypothesis h agrees with f on all examples

realizable if the hypothesis space constains the consistent function

our objective can be restated as a search problem to find the hypothesis h in hypothesis space H that minimizes some objective

1.2 Unsupervised Learning

2 Nearest Neighbour

2.1 Basic NN

nearest neighbours label any example with the label of its nearest neighbours

classification: $h(x) = y_{x*}$
where $y_{x*} = \text{argmin}_{x'} d(x, x')$ is the label associated with the nearest neighbour

2.2 KNN

k-nearest neighbours assign the most frequent label among k nearest neighbours

let $knn(x)$ be the k nearest neighbours
then $y_x = \text{mode}(y_{x'} | x' \in knn(x))$

overfitting a hypothesis h with training accuracy higher than its own testing accuracy $\max(0, \text{trainAccuracy}(h) - \text{testAccuracy}(h))$

- classifier too expressive
- noisy data

- lack of data

underfitting a hypothesis h with training accuracy lower than testing accuracy of some other hypothesis h', $\max(0, \max_{h'} \text{trainAccuracy}(h) - \text{testAccuracy}(h'))$

- classifier not expressive enough

k-fold cross validation split data in k equal subsets, run k experiments testing on one subset, and training on all the others. Report average accuracy

weighted knn weight each neighbour by distance

knn regression y_x is a real value, $y_x \leftarrow average(y_{x'}|x' \in knn(x))$

3 Linear Regression

3.1 Least Squares

find linear hypothesis h: $t = w^T \bar{x}$, find w to minimize euclidean L2 loss

$$w* = argmin_w \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \bar{x}_n)^2$$

where $\bar{x} = \begin{pmatrix} 1 \\ x \end{pmatrix}$ and we can solve with $w = A^{-1}b$ or $Aw = b$ which can be solved as a linear system

3.1.1 Regularization

Least squares can be unstable, overfit, so change optimization

$$w* = argmin_w \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \bar{x}_n)^2 + \frac{\lambda}{2} \|w\|_2^2$$

or $(\lambda I + A)w = b$

3.2 Maximum Likelihood

derive the same thing but from a different perspective: assume $y = w^T \bar{x}+$ gaussian noise so

$$\begin{aligned} Pr(y|\bar{X}, w, \sigma) &= N(y|w^T \bar{X}, \sigma^2) \\ w* &= argmax_w Pr(y|\bar{X}, w, \sigma) \\ &... \\ &= argmin_w \sum_n (y_n - w^T \bar{x}_n)^2 \end{aligned}$$

which is the same as least squares

3.3 Maximum A Posteriori

find $w*$ with highest posterior probability, knowing that prior $P(w) = N(0, \Sigma)$

$$Pr(w|X, y) \propto Pr(w)Pr(y|X, w)$$

therefore for optimization:

$$\begin{aligned} w* &= argmax_w Pr(w|\bar{X}, y) \\ &... \\ &= argmin_w \sum_n (y_n - w^T \bar{x}_n)^2 + w^T \Sigma^{-1} w \end{aligned}$$

let $\Sigma^{-1} = \lambda I$

$$= argmin_w \sum_n (y_n - w^T \bar{x}_n)^2 + \lambda \|w\|_2^2$$

and we arrive at least squares with regularization

3.4 Expected Squared Loss

$$\begin{aligned} E[loss] &= \int_{x,y} Pr(x, y)(y - w^T \bar{x})^2 dxdy \\ &= \int_{x,y} Pr(x, y)(y - f(x))^2 + \int_x Pr(x)(f(x) - w^T \bar{x})^2 dx \\ &= \text{noise (constant)} + \text{error (relative to } w) \end{aligned}$$

lets consider the expected error wrt our dataset S

$$\begin{aligned} E[error] &= E_S[(f(x) - \bar{w_S^T})^2] \\ &= (f(x) - E_S[w_S^T \bar{x}])^2 + E_S[(E_S[w_S^T \bar{x}] - w_S^T \bar{x})^2] \\ &= \text{bias}^2 + \text{variance} \end{aligned}$$

therefore putting it together

$$E[loss] = \text{bias}^2 + \text{variance} + \text{noise}$$

3.5 Bayesian Linear Regression

instead of using $w*$, compute weighted avg prediction using $Pr(w|\bar{X}, y)$

$$Pr(w|\bar{X}, y) = N(\bar{w}, A^{-1})$$

where $w = \sigma^{-2} A^{-1} \bar{X}^T y$

$A = \sigma^{-2} \bar{X}^T \bar{X} + \Sigma^{-1}$

let x_* be the input for which we predict y_*

$$\begin{aligned} Pr(y_*|\bar{x}_*, \bar{X}, y) &= \int_w Pr(y_*|\bar{x}_*, w) Pr(w|\bar{X}, y) dw \\ &\dots \\ &= N(\bar{x}_*^T A^{-1} \bar{X}^T y, \bar{x}_*^T A^{-1} \bar{x}_*) \end{aligned}$$

4 Statistical Learning

4.1 Introduction

probability distribution a specific probability for each event in our sample space

joint distribution spec of probabilities for all combinations of events $Pr(A \wedge B)$

conditional probabilties $Pr(A|B) = Pr(A \wedge B)/Pr(B)$

4.2 Bayes Rules

$$Pr(B|A) = \frac{Pr(A|B)Pr(B)}{Pr(A)}$$

posterior P(B|A)

likelihood P(A|B)

prior P(B)

normalizing P(A)

evidence A

4.3 Bayesian Learning

computing the posterior of hypothesis given evidence using Bayes' theorem:

$$Pr(H|e) = kPr(e|H)Pr(H)$$

properties:

+ optimal (given prior)
+ no overfitting (all hypotheses considered)
- intractable if hypothesis space is large

4.4 Approximate Bayesian Learning

Maximum A Posteriori make prediction based on most probable hypothesis (vs basing on all hypotheses weighted by probability)

$$h_{map} = argmax_{h_i} Pr(h_i|e)$$

+ controlled overfitting
+ converges as data increases
- less accurate than Bayesian prediction
- maybe be intractable!

Maximum Likelihood simplify MAP by assuming uniform prior $Pr(h_i) = Pr(h_j) \forall i, j$

$$h_{ml} = argmax_{h_i} Pr(e|h_i)$$

+ still converges
- least accurate because ignore prior info
- overfits

also, can be easier than MAP: $h_{ml} = argmax_h \sum_n \log Pr(e_n|h)$