Probability - L#4

DATE

Uncertainity
→ risks falsehood
→ leads to conclusions that are too weak for decision making.
→ leads to non-optimal decision making.

Probability
Ω → sample space

$\omega \in \Omega$ is a sample point / possible world / atomic event

Prob. space
Func ⇒ $P: \Omega \to \mathbb{R}$ such that $0 \leq P(\omega) \leq 1$

$\sum_{\omega \in \Omega} P(\omega) = 1$

Random variable
- A random variable (outcome of a random phenomenon) is a function from the sample space Ω to some range

$X: \Omega \to \mathbb{B} / \mathbb{R}$

Odd : $\Omega \to$ Boolean

X is a variable & a function

a) $X = x_i$: the random variable X has value $x_i \in \mathbb{B}$

b) $X = x_i = \{\omega \in \Omega \mid X(\omega) = x_i\}$

Propositions
A prop. is the event (subset of Ω) where an assignment to a random variable holds

event $a \equiv A = \text{true} \equiv \{\omega \in \Omega \mid A(\omega) = \text{true}\}$

Prop. can be combined using standard logic operators e.g.

$\neg a \equiv A = \text{false} = \{\omega \in \Omega \mid A(\omega) = \text{false}\}$, etc.

Prior Prob.

↳ belief prior to arrival of any (new) evidence.

DATE

Joint Prob. Dist

→ for a set of random variables gives the prob. of every atomic joint event on those random vars.

Conditional / Posterior Prob

$P(Cavity = true \mid weather = sunny) \neq P(Cavity = true, weather = sunny)$

$\neq P(Cavity = true)$

Conditional Prob. Distribution

Rep. of all values of conditional prob. of random variables.

Total Probabilities

$$P(a) = P(a|b)P(b) + P(a|\neg b)P(\neg b)$$

In general for any random var Y,

$$P(X) = \sum_{y_i \in D(Y)} P(X|Y=y_i)(P(Y=y_i))$$

↳ set of values for variable Y.

Inf. by Enumeration

- For any prop. ϕ, sum the atomic events where it is true.

$$\boxed{P(\phi) = \Sigma_{w:w=\phi} P(w)}$$

$P(cavity \vee toothache) = P(cavity \ldots)$

Normalization

$$P(A|B) = \alpha P(A, B)$$

Conditional Independence

X conditionally independent from Y given Z iff

$$P(X|Y,Z) = P(X|Z)$$

$$P(X, Y|Z) = P(X|Y,Z)P(Y|Z) = P(X|Z)P(Y|Z)$$

Y_i conditionally independent from Y_j given Z.

$$P(Y_1, \ldots, Y_n | Z) = P(Y_1|Z)\,P(Y_2|Z) \ldots P(Y_n|Z)$$

Bayes Rule

DATE

- Product rule $\Rightarrow$ $P(a \cap b) = P(a|b)P(b) = P(b|a)P(a)$

$$\boxed{P(a|b) = \frac{P(b|a)(P(a))}{P(b)}}$$

in dist. form,

$$P(Y|X) = \frac{P(X|Y)(P(Y))}{P(X)} = \alpha\, P(X|Y)P(Y)$$

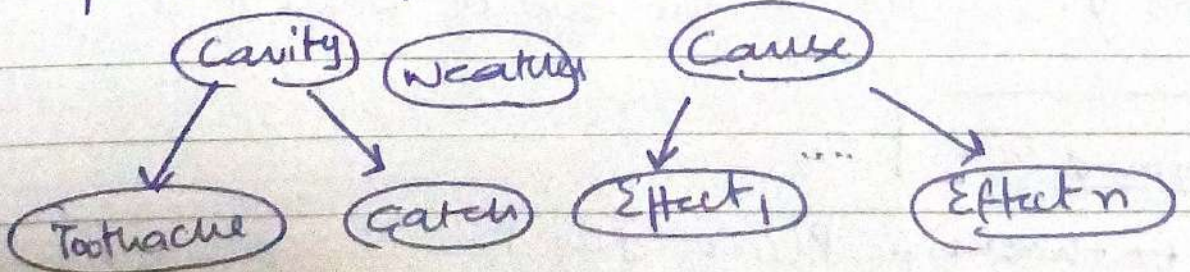
$\alpha \rightarrow \frac{1}{P(X)}$

↳ can be useful for assessing diagnostic prob. from causal prob.

$$P(\text{Cause}|\text{Effect}) = \frac{P(\text{Effect}|\text{Cause})P(\text{Cause})}{P(\text{Effect})}$$

Bayesian Networks

↳ a simple graphical notation for conditional independence assertions and hence for compact specification of full joint dist.



- weather is independent and Toothache and catch are conditionally dep. given cavity.

So the BN model would be:

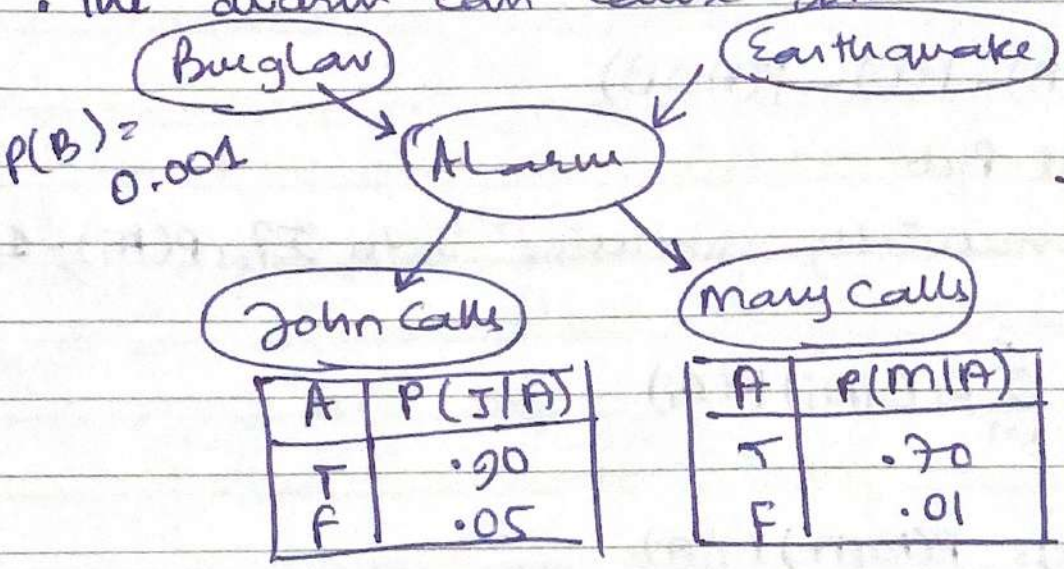
$\Rightarrow$ $P(X_i \mid \text{Parents}(X_i))$ for each variable X_i

$\rightarrow$ $P(\text{weather})$, $P(\text{Cavity})$, $P(\text{Toothache}|\text{Cavity})$, $P(\text{Catch}|\text{Cavity})$

Burglar BN

Var: Burglar, Earthquake, John Calls, Mary Calls

- A burglar can set the alarm. DATE
- An earthquake can set the alarm.
- The alarm can cause Mary to call.
- The alarm can cause John to call.



$P(B) = 0.001$

$P(E) = 0.002$

B	E	P(A | B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J | A)
T	.90
F	.05

A	P(M | A)
T	.70
F	.01

A CPT for Boolean variable X_i with k Boolean parents has 2^k rows for combs of parent values.

- Each row requires one number p for X_i = true (for X_i = false, 1-p)
- If each variable has no more than k parents, then network requires $O(n \cdot 2^k)$ numbers
- Grows linearly with n as compared to 2^n for full joint prob. dist.

Bayesian Learning – L#5

Basic Formulas

DATE

Product Rule

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

Sum Rule

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Theorem of Total Prob

If $A_1 \ldots A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$

then

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

Bayes Rule

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

Classification as Probabilistic Estimation

Given a target function $f: X \to V$, dataset D and a new instance x', best prediction $\hat{f}(x') = v^*$

$$\boxed{v^* = \underset{v \in V}{\operatorname{argmax}}\, P(v|x', D)}$$

Given D and x', compute prob. distribution over V

$$P(V|x', D)$$

Learning as Prob. Estimation

Given a dataset D and hypothesis space H compute a prob. dist H over D. $P(H|D)$

Bayes rule applying

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$P(h)$ → prior prob. of hypothesis

$P(D)$ → prior prob. of training data

MAP Hypotheses

Generally we want the most probable hypothesis h given D.
Maximum a posteriori hypothesis (h_MAP)

DATE

$$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}} P(h|D) = \underset{h \in H}{\operatorname{argmax}} \frac{P(D|h)(P(h))}{P(D)}$$

$$\boxed{h_{MAP} = \underset{h \in H}{\operatorname{argmax}} P(D|h) P(h)} \quad \text{— (1)}$$

ML Hypothesis

If we assume $P(h_i) = P(h_j)$ we can further simply eq (1) and say

$$\boxed{h_{ML} = \underset{h \in D}{\operatorname{argmax}} P(D|h)}$$

Brute Force MAP Hypotheses Learner

- for each $h \in H$, calculate posterior prob.

$$P(h|D) = \frac{P(D|h) P(h)}{P(D)}$$

- Output the hypothesis with highest posterior prob.

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} P(h|D)$$

Bayes Optimal classifier

Consider target function $f: X \to V$, $V = \{v_1, \ldots, v_k\}$ dataset D and a new instance $x \notin D$

$$\boxed{P(v_j|x, D) = \sum_{h \in H} P(v_j|x, h_i) P(h_i|D)}$$

Total prob. over H

$P(v_j|x, h_i)$: prob. that $h_i(x) = v_j$ is indep from D given h.

$$\boxed{P(v_j|x, h_i) = P(v_j|x, h_i, D)}$$

BOC

class of a new instance x

$\Rightarrow V_{OB} = \underset{v_j \in V}{\operatorname{argmax}} \sum_{h_i \in H} P(v_j | x, h_i) P(h_i | D)$

DATE

No other method using same hypothesis space & same prior knowledge can outperform this method on average

Very powerful: Labelling new instances x with $\underset{v_j \in V}{\operatorname{argmax}} P(v_j | x, D)$ can correspond to none of hypotheses in H.

Example

$P(h_i | \{d_1\}) = \alpha P(\{d_1\} | h_i) P(h_i)$

$P(h_i | \{d_1, d_2\}) = \alpha P(\{d_1, d_2\} | h_i) P(h_i)$

$= \alpha P(\{d_2\} | h_i) P(\{d_1\} | h_i) P(h_i)$

• because independent data samples.

General Approach

Given dataset $D = \{d_i\}$ with $d_i \in \{0, 1\}$, assuming a prob. distribution $P(d_i, \theta)$

MLE $\Rightarrow \theta_{ML} = \underset{\theta}{\operatorname{argmax}} \log P(d_i | \theta)$

For Bernoulli dist,

$P(X = k; \theta) = \theta^k (1-\theta)^{1-k}$

$\theta_{ML} = \dots = \frac{|\{d_i = 1\}|}{|D|}$

Bernoulli Dist

→ Prob. dist of binary random variable $X \in \{0, 1\}$

$P(X = 1) = \theta$

$P(X = 0) = 1 - \theta$

Multi-variate Bernoulli Dist

Joint prob. dist of a set of binary random variables $X_1, \ldots, X_n$, each random var. following Bernoulli dist.

DATE

$$P(X_1 = k_1, \ldots, X_n = k_n ; \theta_1, \ldots, \theta_n)$$

$k_i \in \{0, 1\}$

Under the assumption that random variables X_i are mutually indep, Multivariate Bernoulli dist is the product of n Bernoulli distributions.

$$P(X_1 = k_1, \ldots ; \theta_1, \ldots, \theta_n) = \prod_{i=1}^{n} P(X_i = k_i ; \theta_i) = \prod_{i=1}^{n} \theta_i^{k_i} (1-\theta_i)^{1-k_i}$$

Binomial Dist

prob. dist of k outcomes from n Bernoulli-trials.

$$P(X = k ; n, \theta) = \binom{n}{k} \theta^k (1-\theta)^{1-k}$$

Multinomial Dist

↳ Generalization of binomial dist for discrete valued random variables with d possible outcomes.

Prob. dist of k_1 outcomes for $X_1, \ldots, k_d$ outcomes for X_d after n trials with $\sum_{i=1 \ldots d} k_i = n$

$$P(X_1 = k_1, \ldots, X_d = k_d ; n ; \theta_1, \ldots, \theta_d) = \frac{n!}{k_1! \ldots k_n!} \theta_1^{k_1} \ldots \theta_d^{k_d}$$

Naive Bayes Classifier

↳ uses conditional independence to approximate the solution.

X is conditionally indep. of Y given Z.

$$P(X, Y|Z) = P(X|Y, Z)\, P(Y|Z) = P(X|Z)\, P(Y|Z)$$

Assume target function DATE

$f: X \to V$ where each instance x is described by attributes $\langle a_1, a_2 \ldots, a_n \rangle$

Compute

$$\underset{v_j \in V}{\text{argmax}}\, P(v_j | x, D) = \underset{v_j \in V}{\text{argmax}}\, P(v_j | a_1, \ldots a_n, D)$$

→ No explicit representation of hypotheses.

Given a dataset D and a new instance $x = \langle a_1, \ldots a_n \rangle$

most prob. value of $f(x)$ is:

$$v_{MAP} = \underset{v_j \in V}{\text{argmax}}\, P(v_j | a_1, a_2 \ldots, a_n, D)$$

$$= \underset{v_j \in V}{\text{argmax}}\, \frac{P(a_1, a_2, \ldots, a_n | v_j, D)\, P(v_j | D)}{P(a_1, a_2, \ldots, a_n | D)}$$

$$\boxed{v_{MAP} = \underset{v_j \in V}{\text{argmax}}\, P(a_1, a_2 \ldots a_n | v_j, D)\, P(v_j | D)}$$

Naive Bayes assumption

$$P(a_1, a_2 \ldots a_n | v_j, D) = \prod_i P(a_i | v_j, D)$$

class of new instance x

$$\boxed{v_{NB} = \underset{v_j \in V}{\text{argmax}}\, P(v_j | D) \prod_i P(a_i | v_j, D)}$$

$$\hat{P}(v_j|D) = \frac{|\{\langle \ldots, v_j \rangle\}|}{|D|}$$

$$\hat{P}(a_i|v_j, D) = \frac{|\{\langle \ldots, a_i, \ldots, v_j \rangle\}|}{|\{\langle \ldots, v_j \rangle\}|}$$

DATE

Typical solution is Bayesian estimate with prior estimates

$$\hat{P}(a_i|v_j, D) = \frac{|\{\langle \ldots, a_i, \ldots v_j \rangle\}| + mp}{|\{\ldots, v_j\}| + m}$$

$p \rightarrow$ prior estimate for $P(a_i|v_j, D)$

$m \rightarrow$ weight given to the prior estimate.

Probabilistic Models for Classification – L#6

Given $f: x \rightarrow C$, $D = \{(x_i, C_i)_{i=1}^n\}$ and $x \notin D$ estimate

$P(C_i|x, D)$

$P(C_i|x) \rightarrow$ posterior

$P(x|C_i) \rightarrow$ class-conditional densities

Two families of models

Generative: estimate $P(x|C_i)$ and then compute $P(C_i|x)$

Discriminative: estimate $P(C_i|x)$ directly

→ Prob. Generative Models

Consider a case of two classes

Find the conditional prob.

$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x)} = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

$$= \frac{1}{1+e^{-a}} = \sigma(a)$$

$$\hookrightarrow a = \ln \frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}$$

Assume $p(x|C_i) = \mathcal{N}(x; \mu_i, \Sigma)$ - same covariance matrix

$$a = \ln \frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}$$

DATE

$$= \ln \frac{\mathcal{N}(x; \mu_1, \Sigma)P(C_1)}{\mathcal{N}(x; \mu_2, \Sigma)P(C_2)} = w^T x + w_0$$

$$w = \Sigma^{-1}(\mu_1 - \mu_2)$$

$$w_0 = -\frac{1}{2}\mu_1^T \Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^T \Sigma^{-1}\mu_2 + \ln \frac{P(C_1)}{P(C_2)}$$

Thus, $\boxed{P(C_1|x) = \sigma(w^T x + w_0)}$

Multi-class

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{\sum_j P(x|C_j)P(C_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

(normalized exponential & softmax fn)

$a_k = \ln P(x|C_k)P(C_k)$

MLE for 2 classes

Assume $P(C_1) = \pi$ ($P(C_2) = 1-\pi$), $P(x|C_i) = \mathcal{N}(x, \mu_i, \Sigma)$

Given a dataset $D = \{(x_n, t_n)_{n=1}^N\}$, $t_n = 1$ if x_n belongs to class C_1, $t_n = 0$ if x_n belongs to class C_2.

Let N_1 be the # of samples in D belonging to C_1 and N_2 be the # of samples of C_2 ($N_1 + N_2 = N$)

Likelihood

$$P(t|\pi, \mu_1, \mu_2, \Sigma, D) = \prod_{n=1}^{N} [\pi \mathcal{N}(x_n, \mu_1, \Sigma)]^{t_n} [(1-\pi)\mathcal{N}(x_n, \mu_2, \Sigma)]^{(1-t_n)}$$

Maximize log-likelihood

$$\pi = N_1/N$$

$$\mu_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n x_n$$

$$\mu_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1-t_n) x_n$$

$$\Sigma = \frac{N_1}{N} S_1 + \frac{N_2}{N} S_2$$

$$S_i = \frac{1}{N_i} \sum_{n \in C_i} (x_n - \mu_i)(x_n - \mu_i)^T; \; i = 1, 2$$

For k- classes

$$P(C_k) = \pi_k; \; P(x|C_k) = \mathcal{N}(x, \mu_k, \Sigma)$$

Dataset $D = \{(x_n, t_n)_{n=1}^{N}\}$ with t_n 1-to-K encoding

$$\boxed{\pi_k = \frac{N_k}{N}}$$

$$\boxed{\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} t_{nk} x_n}$$

$$\Sigma = \sum_{k=1}^{K} \frac{N_k}{N} S_k, \; S_k = \frac{1}{N_k} \sum_{n=1}^{N} t_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

Prob. Models → represent posterior distributions with parametric models

For two classes $\boxed{P(C_1|x) = \sigma(a)}$

For $k \geq 2$ classes $\boxed{P(C_i|x) = \frac{e^{a_k}}{\sum_j e^{a_j}}}$

$$a_k = w^T x + w_0$$

Compact Notation

$$w^T x + w_0 = (w_0 \; w)\begin{pmatrix} 1 \\ x \end{pmatrix}$$

$$\tilde{w} = \begin{pmatrix} w_0 \\ w \end{pmatrix}, \tilde{x} = \begin{pmatrix} 1 \\ x \end{pmatrix}$$

DATE

$$a_k = w^T x + w_0 = \tilde{w}^T \tilde{x}$$

MLE for Parametric Models

Likelihood $M_\theta : P(t|\theta, D)$. $D = \langle x, t \rangle$

Solution $\theta^* = \underset{\theta}{\text{argmax}} \ln P(t|\theta, X)$

When M_θ belongs to exponential family

$$M_\theta : P(t|\theta, D)$$

Likelihood $P(t|\theta, X)$ can be expressed in the form $P(t|\tilde{w}, X)$ with maximum likelihood

$$\boxed{\tilde{w}^* = \underset{\tilde{w}}{\text{argmax}} \ln P(t|\tilde{w}, X)}$$

Probabilistic Discriminative Models

Estimate directly

$$P(C_k|\tilde{x}, D) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}, \quad a_k = \tilde{w}^T \tilde{x}$$

MLE $\tilde{w}^* = \underset{\tilde{w}}{\text{argmax}} \ln P(t|\tilde{w}, X)$

$$= \tilde{w}^* = \underset{\tilde{w}}{\text{argmax}} \ln P(t|\tilde{w})$$

without estimating the model parameters.

Logistic Regression

↳ Prob. discriminative model based on MLE.

DATE

Two classes

Given dataset $D = \{(\vec{x}_n, t_n)_{n=1}^N\}$ with $t_n \in \{0,1\}$

Likelihood fkn

$$p(t|\vec{w}) = \prod_{n=1}^{N} y_n^{t_n} (1-y_n)^{1-t_n}$$

with $y_n = p(C_1|\vec{x}_n) = \sigma(\vec{w}^T \vec{x}_n)$

$t_n \rightarrow$ value in data corresponding to x_n

y_n: posterior pred. of current model $\vec{w}$ for x_n.

Cross-entropy error function

$$E(\vec{w}) = -\ln p(t|\vec{w}) = -\sum_{n=1}^{N} [t_n \ln y_n + (1-t_n) \ln(1-y_n)]$$

Solve opt. problem

$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}}\, E(\vec{w})$$

can be solved by either N-R / Gradient-Descent, etc.

Newton-Raphson,

$$\nabla E(\vec{w}) = \sum_{n=1}^{N} (y_n - t_n)\vec{x}_n$$

Gradient descent step,

$$\vec{w} \leftarrow \vec{w} - H(\vec{w})^{-1} \nabla E(\vec{w})$$

$H(\vec{w}) = \nabla\nabla E(\vec{w})$ is the Hessian matrix of $E(\vec{w})$

Iterative Reweighted Least Squares

Given

$$\vec{x} = \begin{pmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_n^T \end{pmatrix} \quad t = \begin{pmatrix} t_1 \\ \vdots \\ t_N \end{pmatrix}$$

$y(\vec{w}) = (y_1, \ldots, y_n)^T$ posterior predictions

$R(\vec{w}) \Rightarrow$ diagonal matrix $R_{nn} = y_n(1-y_n)$

$$\nabla E(\vec{w}) = \vec{X}^T (y(\vec{w}) - t)$$

$$H(\vec{w}) = \nabla\nabla E(\vec{w}) = \sum y_n(1-y_n)\vec{x}_n\vec{x}_n^T = \vec{X}^T R(\vec{w}) \vec{X}$$

DATE

Iterative method

- initialize $\vec{w}$
- Repeat until termination

$$\vec{w} \leftarrow \vec{w} - (\vec{X}^T R(w) \vec{X})^{-1} \vec{X}^T (y(w) - t)$$

For multi-class,

$$P(C_k|\vec{x}) = \frac{e^{a_k}}{\sum_j e^{a_j}} \qquad a_k = \vec{w}_k^T \vec{x}$$

Model,

$$p(T|\vec{w}_1, \ldots, \vec{w}_K) = \prod_{n=1}^{N}\prod_{k=1}^{K} P(C_k|x_n)^{t_{nk}} = \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

$y_{nk} = Y[n,k]$

$t_{nk} = T[n,k]$

Cross Entropy

$$E(\vec{w}_1, \ldots, \vec{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}$$

Kernels – L#9

Objects are rep. with fixed-length feature vectors $x \in \mathbb{R}^m$

what about variable length objs or infinite dims? or $\phi(x)$

↳ images, time-series, strings, etc

- use a similarity measure b/w instances x, x'.

$k(x, x')$ → kernel fcn

↳ similarity measure

$\Rightarrow k(x, x') \geq 0$

If $\phi(x)$, then we can use

$$k(x, x') = \phi(x')\phi^T(x)$$

Kernel function: a real valued function $k(x, x')$ where $\mathcal{X}$ is some abstract space for $x, x' \in \mathcal{X}$.

DATE

Typically k is: symmetric

$$k(x, x') = k(x', x)$$

non-negative

$$k(x, x') \geq 0$$

Note: Not strictly required.

- Input data D must be normalized in order for the kernel to be a good similarity measure

1) Min-max normalization: $\bar{x} = \frac{\bar{x} - \min}{\max - \min}$

2) Standardization: $\bar{x} = \frac{x - M}{\sigma}$

3) Unit Vector: $\bar{x} = \frac{x}{\|x\|}$

Kernel families

Linear $\Rightarrow$ $k(x, x') = x^T x'$

Poly $\Rightarrow$ $k(x, x') = (\beta x^T x' + \gamma)^d$, $d \in (2, 3, \dots)$

RBF $\Rightarrow$ $k(x, x') = \exp(-\beta |x - x'|^2)$

Sigmoid $\Rightarrow$ $k(x, x') = \tanh(\beta x^T x' + \gamma)$

Consider a linear model $y(x; w) = w^T x$ with dataset

$$D = \{(x_n, t_n)_{n=1}^N\}$$

Minimize $J(w) = (t - Xw)^T (t - Xw) + \lambda \|w\|^2$

$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix}$ design matrix, $t = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}$ output vector

Optimal solution

$\hat{w} = (X^TX + \lambda I_N)^{-1} X^T t = X^T (XX^T + \lambda I_N)^{-1} t$

with I_N the $N \times N$ → identity matrix

DATE

Kernelized Linear Models

$\alpha = (XX^T + \lambda I_N)^{-1} t$

$\hat{w} = X^T \alpha = \sum_{n=1}^{N} \alpha_n x_n^T X$

If we consider a linear kernel $k(x, x') = x^T x'$, we can re-write the model as:

$$y(x; \hat{w}) = \sum_{n=1}^{N} \alpha_n k(x, x')$$

$\alpha = (K + \lambda I_N)^{-1}$

$K = XX^T$ → Gram matrix

↳

$$\begin{bmatrix} x_1^T x_1 & \cdots & x_1^T x_N \\ \vdots & & \\ x_N^T x_1 & \cdots & x_N^T x_N \end{bmatrix}$$

$k(x_1, x_1)$

linear model with kernel $k(x, x') = x^T x'$

$$y(x; \alpha) = \sum_{n=1}^{N} \alpha_n x_n^T x$$

$\alpha = (K + \lambda I_N)^{-1} t$

Gram matrix

$$K = \begin{bmatrix} x_1^T x_1 & \cdots \\ & \\ & \end{bmatrix}$$

$$K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix}$$

Kernel Trick

(kernel substitution)

DATE

- If input vector x appears in an algorithm only in the form of an inner product $x^T x'$, replace the inner product with some kernel $k(x, x')$
- Can be applied to any x (even infinite size)
- No need to know $\phi(x)$
- Directly extend many well-known algorithms

Kernelized SVM - classification

In SVM, the solution has the form

$$\hat{w} = \sum_{n=1}^{N} \alpha_n x_n$$

Linear model (linear kernel)

$$y(x, \alpha) = -\left(w_0 + \sum_{n=1}^{N} \alpha_n x_n^T x\right)$$

with kernel trick,

$$y(x, \alpha) = -\left(w_0 + \sum_{n=1}^{N} \alpha_n k(x_n, x)\right)$$

Langrangian Problem for Kernelized SVM

$$\hat{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} a_n a_m t_n t_m k(x_n, x_m)$$

$a_n = \dots$

$$\Rightarrow w_0 = \frac{1}{|SV|} \sum_{x_i \in SV} \left(t_i - \sum_{x_j \in S} a_j t_j k(x_i, x_j)\right)$$

Kernelized Linear Regression

Linear model for regression $y = w^T x$ and data set $D = \{(x_n, t_n)_{n=1}^N\}$

DATE

Minimize the regularized loss fun

$$J(w) = \sum_{n=1}^{N} E(y_n, t_n) + \lambda \|w\|^2$$

where $y_n = w^T x_n$

$$E(y_n, t_n) = (y_n - t_n)^2$$

$$\hat{w} = (X^T X + \lambda I_N)^{-1} X^T t = X^T \alpha$$

$$\alpha = (\underline{X^T X} + \lambda I_N)^{-1} t$$

To make predictions

$$\Rightarrow y(x; \hat{w}) = \sum_{n=1}^{N} \alpha_n \underline{x_n^T x}$$

$$\sum_{n=1}^{N} \alpha_n k(x, x)$$

$$\alpha = (K + \lambda I_N)^{-1} t$$

gram matrix

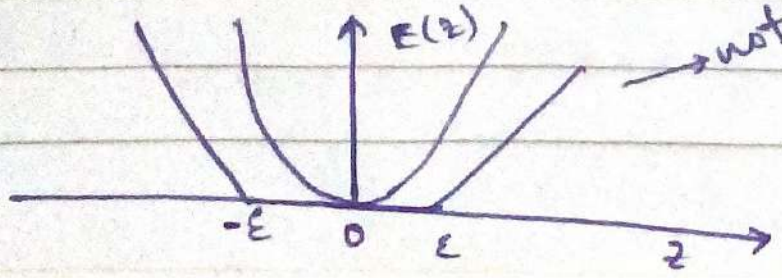
$\Rightarrow$ computation of K requires $|D|^2$ operations & K is not sparse.

Consider,

$$J(w) = C \sum_{n=1}^{N} E_\varepsilon(y_n, t_n) + \frac{1}{2} \|w\|^2$$

with C inverse of λ and an ε-insensitive error function.

$$E_\varepsilon(y, t) = \begin{cases} 0 & \text{if } |y - t| < \varepsilon \\ |y - t| - \varepsilon & \text{otherwise} \end{cases}$$



→ not differentiable → difficult to solve

Introduce slack variables $\xi_n^+, \xi_n^- \geq 0$

$$t_n \leq y_n + \varepsilon + \xi_n^+$$
$$t_n \geq y_n - \varepsilon - \xi_n^-$$

DATE

Points inside the ε-tube $y_n - \varepsilon \leq t_n \leq y_n + \varepsilon \Rightarrow \xi_n^+ = \xi_n^- = 0$

$$\left.\begin{aligned} \xi_n^+ > 0 &\Rightarrow t_n > y_n + \varepsilon \\ \xi_n^- > 0 &\Rightarrow t_n < y_n - \varepsilon \end{aligned}\right\} y_n = y(x_n; w)$$

Loss - re written

$$J(w) = C \sum_{n=1}^{N} (\xi_n^+ + \xi_n^-) + \frac{1}{2}\|w\|^2$$

subj. to constraints

$$t_n \leq y(x_n; w) + \varepsilon + \xi_n^+$$
$$t_n \geq y(x_n; w) - \varepsilon - \xi_n^-$$
$$\xi_n^+ \geq 0, \quad \xi_n^- \geq 0$$

→ This is a standard quadratic program, can be easily solved.

Lagrangian Problem

$$\tilde{L}(a, a') = \cdots \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m \cdots k(x_n, x_m) \cdots$$

compute $\hat{a}_n$, $\hat{a}_m$ (sparse) and

$$\hat{w}_0 = t_n - \varepsilon - \sum_{m=1}^{N} (\hat{a}_m - \hat{a}_m') K(x_n, x_m)$$

for some data point n such that $0 < a_n < C$

Pred.

$$\boxed{y(x) = \sum_{n=1}^{N} (\hat{a}_n - \hat{a}_n') K(x, x_n) + \hat{w}_0}$$

From Karush-Kuhn-Tucker (KKT) conditions,

Support vectors contribute to predict

$\hat{a}_n > 0 \rightarrow \varepsilon + \xi_n + y_n - t_n = 0$

↳ data point lies on or above the upper boundary of ε-tube

$a_n' > 0 \Rightarrow \varepsilon + \xi_n - y_n + t = 0$

↳ data point lies on or below lower boundary of ε-tube.

→ All other data points inside the ε-tube have $\hat{a}_n = 0$, and $\hat{a}_n' = 0$ and thus do not contribute to pred.

DATE

~~Decision Process~~

Dimensionality Reduction – L#15

For data with structure *

- We expect fewer distortions than dimensions
- Data live in a lower dimension manifold

Conclusion: Deal with high dimensional data by looking for lower dimensional embedding

Principal Component Analysis (PCA)

↳ widely used technique for various tasks such as:

- dimensionality reduction
- data visualization
- data compression
- feature extraction

Given data $\{x_n\} \in \mathbb{R}^D$

Goal: Maximize data variance after projection to some direction u_1

Projected points: $u_1^T x_n$

Note: $u_1^T u_1 = 1$

Mean value $\Rightarrow \bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$

Data centered matrix (X) $(N \times D)$:

$$X = \begin{bmatrix} (x_1 - \bar{x})^T \\ \vdots \\ (x_n - \bar{x})^T \\ \vdots \\ (x_N - \bar{x})^T \end{bmatrix}$$

Mean of projected points: $u_1^T \bar{x}$

Variance of projected points: $\frac{1}{N}\sum_{n=1}^{N}[u_1^T x_n - u_1^T \bar{x}]^2 = u_1^T S u_1$

with S ($D \times D$) covariance matrix of the data.

DATE

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T = \frac{1}{N}X^T X$$

Maximize the projected variance: $\max_{u_1} u_1^T S u_1$

subj. to constraint $u_1^T u_1 = 1$

Equivalent to unconstrained maximization with a Lagrange multiplier

$\Rightarrow \max_{u_1} u_1^T S u_1 + \lambda_1(1 - u_1^T u_1)$

Setting derivative w.r.t u_1 to zero we have

$\rightarrow \boxed{S u_1 = \lambda_1 u_1}$

u_1 must be an eigenvector of S

Multiply u_1^T

$u_1^T S u_1 = \lambda_1 \underbrace{u_1^T u_1}_{1}$

$\boxed{u_1^T S u_1 = \lambda_1}$ $\rightarrow$ variance after projection

$\rightarrow$ Variance is maximal when u_1 is the eigenvector corresponding to largest eigenvalue λ_1.

This is called the first principal component.

$\Rightarrow$ Repeat to find other directions which

- Maximize variance of projected data
- Are orthogonal to the previous directions

Steps: To perform PCA in an M-dimensional projection space with $M < D$

- compute $\bar{x}$: mean of data
- compute S: covar. mat of data
- compute M eigenvectors of S corresponding to M larg- eigenvalues

PCA Error Minimization

Consider a complete orthonormal D-dimensional basis

$$u_i^T u_j = \delta_{ij}$$

DATE

with $\delta_{ij} = \begin{cases} 0 & \text{otherwise} \\ 1 & \text{if } i=j \end{cases}$

Each data point can be written as:

$$x_n = \sum_{i=1}^{D} a_{ni} u_i$$

Using the orthonormality prop. we have,

$$\underline{a_{nj} = x_n^T u_j}$$

$$x_n = \sum_{n=1}^{D} (x_n^T u_i) u_i$$

Goal: Approx. x_n using a lower dimensional rep.

$$\tilde{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i$$

Evaluate approx error as:

$$J = \frac{1}{N} \sum_{n=1}^{N} \| x_n - \tilde{x}_n \|^2$$

Minimizing w.r.t z_{nj} we get

$$z_{nj} = x_n^T u_j, \; j = 1 \ldots M$$

Minimizing w.r.t b_j we get

$$b_j = \bar{x}^T u_j, \; j = M+1 \ldots D$$

Using these expressions we get,

$$\Rightarrow x_n - \tilde{x}_n = \sum_{i=M+1}^{D} [(x_n - \bar{x})^T u_i] u_i$$

The overall approx. error becomes

$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} (x_n^T u_i - \bar{x}^T u_i) = \sum_{i=M+1}^{D} u_i^T S u_i$$

Minimize the approx. error subj. to $u_i^T u_i = 1$

$\rightarrow J = \sum_{i=M+1}^{D} u_i^T S u_i + \lambda_i (1 - u_i^T u_i)$

Set $d(u_i)' = 0$

DATE

$Su_i = \lambda_i u_i$

u_i is an eigen vector of S with eigenvalue λ_i.

→ The error: $J = \sum_{i=M+1}^{D} \lambda_i$ ~ This is minimized using u_i as the eigenvectors corresponding to $D-M$ smallest eigenvalues

- Choosing $D-M$ smallest eigenvalues of S corresponds of M highest eigenvalues of S in max. var. dir.

PCA for high dimensionality ($N < D$)

At least $D-N+1$ eigenvalues of S are zero.

Ex: small-set of high-resolution images.

In this case finding eigenvalues of S ($D \times D$) matrix is inefficient.

For $N < D$:

Let X be the $N \times D$ centred data matrix (i.e. n-th row is $(x_n - \bar{x})^T$) and corresponding covariance mat.

$$S = \frac{1}{N} X^T X$$

The corresponding eigenvector eqs. are:

$$\frac{1}{N} X^T X u_i = \lambda_i u_i$$

Left-multip. by X,

$$\frac{1}{N} XX^T(Xu_i) = \lambda_i (Xu_i)$$

DATE

$$v_i = Xu_i$$

$$\Rightarrow \frac{1}{N} XX^T v_i = \lambda_i v_i$$

- XX^T has same eigenvalues $(N-1)$ of X^TX (others are 0)
- XX^T is $N \times N$ matrix whose eigenvalues can be computed efficiently.

Given eigenvalues λ_i of XX^T, to find eigenvectors we left-multiply by X^T

$$\Rightarrow \left(\frac{1}{N} X^TX\right)(X^Tv_i) = \lambda_i \underbrace{(X^Tv_i)}_{\text{eigenvector}}$$

(eigenvalue)

$$u_i = \frac{1}{\sqrt{N\lambda_i}} X^Tv_i$$

Summing up, $N \ll D$

- Consider the centered data matrix X
- Compute $N-1$ eigenvalues λ_i and eigenvectors v_i of XX^T
- $u_i = \frac{1}{\sqrt{N\lambda_i}} X^Tv_i$

The so-obtained $N-1$ vectors u_i are eigenvectors of $S = X^TX$, with non-null eigenvalue λ_i.

Prob. PCA

- Rep. data x with lower dimensional latent variables z
- Assume linear relationship DATE

$$x = Wz + \mu$$

- Assume Gaussian dist of latent var.

$$P(z) = \mathcal{N}(z; 0, I)$$

- Assume Linear-Gaussian relationship b/w latent vars & data.

$$P(x|z) = \mathcal{N}(x; Wz + \mu, \sigma^2 I)$$

Maginal dist

$$P(x) = \int P(x|z) P(z) dz = \mathcal{N}(x; \mu, C)$$

$$C = WW^T + \sigma^2 I$$

Posterior dist:

$$P(z|x) = \mathcal{N}(z; M^{-1} W^T (x - \mu), \sigma^2 M^{-1})$$

$$\rightarrow M = W^T W + \sigma^2 I$$

MLE: given data X

$$\Rightarrow \underset{W, \mu, \sigma}{\operatorname{argmax}} \ln P(X|W, \mu, \sigma^2) = \sum_{n=1}^{N} \ln P(x_n|W, \sigma^2, \mu)$$

set derivatives to 0, we get.

$$\mu_{ML} = \bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

$$W_{ML} = \text{---}, \quad \sigma^2_{ML} = \text{---}.$$

ML $\rightarrow$ PCA can be obtained by EM.

AEs for anomaly detect.

- Train an autoencoder (AE) with $\{x_n\}$ (learn a latent rep. for normal samples)

DATE

compute final train loss

- Determine a threshold: δ = mean(loss) + std(loss)
- Given $x' \notin D$, reconstruct x' with AE and compute loss'.
- If loss' < δ return normal otherwise abnormal.

→ Works better with ensemble of AEs.

VAEs → focus on learning latent space structure

GANs → focus on learning a distribution

(CNNs)

$$w_{out} = \frac{w_{in} - w_K + 2p}{s} + 1$$

$$h_{out} = \frac{h_{in} - h_K + 2p}{s} + 1$$

# of trainable params of conv layer.

$$|\theta| = \underbrace{w_K \cdot h_K \cdot d_{in} \cdot d_{out}}_{\text{kernel weights}} + \underbrace{d_{out}}_{\text{bias}}$$