1

Teoria da Informação

Fórmula da surpresa → $\log_2 \frac{1}{p(x)}$

+ surpresa, - probabilidade / é contínua / é aditiva

- Entropia → $H(X) = -\sum p(x) \log_2 p(x)$ (surpresa esperada)

Unidades de entropia - bit (base 2), nat (base e), Hartley (base 10)
- fornece uma medida de incerteza sobre uma variável aleatória
- aproxima-se de zero quando um dos símbolos é mais provável (0 ou 1)

Propriedades: - $H(X) \geq 0$
- $H(X) \leq \log_2 |X|$ → Cardinalidade do alfabeto
- entropia máxima quando tem distribuição uniforme - $p(x) = \frac{1}{|x|}$

- Entropia Conjunta → $H(X,Y) = -\sum p(x,y) \log_2 p(x,y)$

Propriedades: - $H(X,Y) = H(X) + H(Y)$, para v.a. independentes
- $H(X,Y) = H(X) = H(Y)$, para v.a. $X = Y$
- $H(X,Y) = H(X)$, se $Y = f(X)$

- Entropia Condicional → $H(Y|X=x) = -\sum p(y|x) \log_2 p(y|x)$

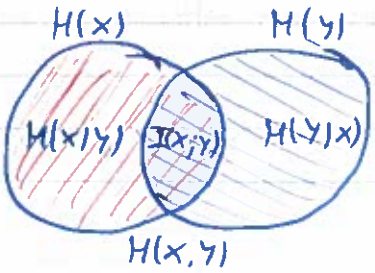
$H(Y|X) = -\sum p(x) \sum p(y|x) \log_2 p(y|x)$

Propriedades: - $H(Y|X) \leq H(Y)$
- $H(X,Y) = H(X) + H(Y|X)$
- $H(X,Y) = H(Y) + H(X|Y)$

- Informação Mútua → $I(X;Y) = \sum p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$

Propriedades: - $p(x,y) = p(x)p(y)$, se v.a. independentes logo
$I(X;Y) = \sum p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} = 1 = 0$
- $I(X;Y) \geq 0$
- $I(X;Y) \leq \min(H(X), H(Y))$

$I(X;Y) = H(X) - H(X|Y)$
$= H(Y) - H(Y|X)$
$= H(X) + H(Y) - H(X,Y)$



Entropia relativa - $D(p(x)\|q(x)) = \sum p(x) \log_2 \frac{p(x)}{q(x)}$
(divergência de Kullback - Leibler)
• mede o desvio entre duas distribuições de probabilidade

Satisfaz - $D(p\|p) = 0$ e $D(p\|q) > 0$, se $p \neq q$
Não satisfaz - $D(p\|q) \neq D(q\|p)$; $D(p\|q) \not\leq D(p\|r) + D(r\|q)$
(por isto não é uma distância)

Relação entre informação mútua e entropia relativa

$$I(X;Y) = D(p(x,y)\|p(x)p(y))$$

Propriedades: - $p(x,y) = p(x)p(y) \rightarrow I(X;Y) = 0$, se X e Y independentes
- $p(x,y) \neq p(x)p(y) \rightarrow I(X;Y) > 0$, se X e Y dependentes

Teorema de Bayes → $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$ / $p(x|y) = \frac{p(y|x)}{\sum p(y|x)p(x)} p(x)$

O aumento ou redução da incerteza de X depende do valor de y

Ganho de informação → $E_Y(D(p(x|y)\|p(x)))$ / $H(X) - H(X|Y)$
(entropia relativa entre a distribuição a posteriori e o prior $D(p(x|y)\|p(x))$

Processo determinístico → futuro determinado (execução algoritmo)
Processo estocástico → futuro incerto (lançamento dado) (variação ao longo do tempo)

Uma variável pode evoluir em tempo contínuo (longo do dia) ou tempo discreto (longo da semana)

Processo estocástico → sequência de v.a., caracterizado por distribuição de probabilidade conjunta ($p(x_1, x_2 \ldots)$)

Processo estocástico estacionário → se a distribuição de probabilidade conjunta se mantem inalterada no tempo

Teoria da Informação

Processo de Markov → cada v.a. depende da precedente e é condicionalmente independente das restantes ($p(x_n | x_{n-1})$)

- É um processo estocástico

$X \rightarrow Y \rightarrow Z = X \xrightarrow{p(y|x)} Y \xrightarrow{p(z|y)} Z$

Ex: Jogo da Glória

Cadeia de Markov → processo de Markov em que a variável X_n toma valores num alfabeto finito

Representam-se por grafos: nós – estados
arcos – probabilidades de transição $p(x_t | x_{t-1})$

Homogénea no tempo → probabilidades de transição constantes ao longo do tempo (diz respeito ao mecanismo de funcionamento)

Estacionário → probabilidade da cadeia se encontrar em certos estados (diz respeito ao estado)

$$p(x_n) = \sum p(x_n, x_{n-1}) = \sum p(x_n | x_{n-1})\, p(x_{n-1})$$

$$\mu_n = P\, \mu_{n-1} \longrightarrow p(x_n) \qquad (\mu_n = P^n \mu_0)$$

Irredutível → possível transitar para todos os estados com probabilidade não nula

Aperiódica → se todos os estados são aperiódicos

Periódica

Aperiódica

Distribuição estacionária → $p(x_n) = p(x_{n-1})$
irredutível e aperiódica
$p(x_n) \rightarrow \mu$ quando $n \rightarrow \infty$

Page Rank → frequência com que as páginas são visitadas é dada pela distribuição estacionária

Ritmo de Entropia → quanto a entropia conjunta cresce ao longo do tempo a cada nova ocorrência

$$H'(\mathcal{X}) = \lim_{n\to\infty} \frac{H(X_1, X_2, \dots, X_n)}{n}$$

Para a cadeia de Markov, - $H(\mathcal{X}) = H(X_n \mid X_{n-1})$
Para v.a. independentes - $H(\mathcal{X}) = H(X)$

Código → $C(x)$ - palavra de código correspondente a x
→ $l(x)$ - comprimento da palavra de código $C(x)$
→ $L(c)$ - comprimento médio do código C $\sum p(x)\, l(x)$

Código não singular → símbolos diferentes tem palavras de código diferentes
→ só garante a descodificação de símbolos isolados

Extensão de um código → $C^*(x_1 x_2 \dots x_n) = C(x_1)\, C(x_2) \dots C(x_n)$

Código Univocamente descodificável → extensão é não singular
(cada string pode apenas ter sido gerada por uma única mensagem

Código instantâneo → (ou código de prefixo) nenhuma palavra de código é prefixo da outra
(pode ser descodificado sem referência às palavras de código futuras)

(Todos os códigos (Não singulares (Univ. Descodificáveis (Instantâneos))))

Exemplo:

x	Singular	Não Singular	Univ. Descodificável	Instantâneo
A	0	0	10	00
B	0	00	00	10
C	1	000	11	111
D	1	0000	110	110

3

Teoria da Informação

Desigualdade de Kraft → é possível construir código instantâneo com palavras de código de comprimento $l(n)$ se e só se $\sum 2^{-l(n)} \leq 1$

$$\sum_{x \in X} 2^{l_{max} - l(n)} \leq 2^{l_{max}} \Leftrightarrow \sum_{n \in \mathbb{Z}} 2^{-l(n)} \leq 1$$

Códigos ótimos → Considera-se o conjunto de todos os códigos instantâneos
Códigos diferentes têm comprimentos médios diferentes
Calcular os comprimentos ótimos $l(n)$ das palavras de código
Os símbolos mais frequentes têm comprimentos menores

Se $p(n)$ é potência negativa de 2:

$$L(C) = \sum p(n) l(n) = \sum p(n) (-\log p(n)) = H(X)$$

Comprimentos superiores:

$$l(n) = \lceil -\log_2 p(n) \rceil$$

Comprimento médio satisfaz:

$$H(X) \leq L(C) < H(X) + 1$$ → Código de Shannon-Fano é subótimo

Para símbolos agrupados:

$$L(C) = \frac{\sum_X p(x) l(x)}{n} \qquad H(X) \leq L(C) < H(X) + \frac{1}{n}$$

$$L(C) - H(X) = \sum_n p(n) \log \frac{p(n)}{q(n)}$$

Código de Huffman → Todos os símbolos são folhas
Juntam-se os 2 nós de menor probabilidade
Palavras de código são caminho da raiz às folhas

Satisfaz: $H(X) \leq L(C) < H(X) + 1$

Assume que os símbolos são v.a. independentes e que as probabilidades não variam no tempo

Só pode ser construída depois de saber as probabilidades da fonte

2 passagens sobre o ficheiro:

- Contar o número de ocorrências de cada símbolo
- Desenhar código e comprimir ficheiro

Código de Huffman Adaptativo

- Não precisa conhecer as probabilidades dos símbolos
- Faz uma passagem pelo ficheiro
- Não precisa transmitir o código

- Constrói-se à medida que são lidos novos símbolos
- ESCAPE - Símbolo que ainda não ocorreram
- Maiores ocorrências de cima para baixo
- Ocorrências ordenadas da esquerda para a direita em cada nível
- Se o símbolo já existe, usa-se o código e incrementa ocorrências
- Se não existe usa o código do escape, acrescenta o símbolo à árvore no sítio do escape e atualiza as ocorrências na árvore

Código Shannon - Fano - Elias

- As palavras de código são obtidas a partir da expansão binária das probabilidades acumuladas

$$\bar{F}(n) = \sum_{x_i < x} p(x_i) + \frac{1}{2} p(n)$$

- Escrevem-se em binário os valores
- Calculam-se os comprimentos das palavras de código $l(n) = \lceil -\log_2 p(n) \rceil + 1$
- As palavras de código são os $l(n)$ bits mais significativos
- O comprimento médio satisfaz $H(X) + 1 \leq L(C) < H(X) + 2$
- Este código tem desvantagem sobre o Huffman, pois usa em média mais 1 bit por símbolo, mas a codificação aritmética mostra que se obtém um código com desempenho superior

Codificação Aritmética

- Agrupam n símbolos. n é o tamanho da mensagem a comprimir e aplica o código SFE
- Só é preciso calcular uma única palavra de código

- A probabilidade de cada caminho desde a raiz até cada uma das bolas representa a probabilidade de todas as strings com esse prefixo

Ex: $\bar{F}(BANANA) = p(A) + p(BAA) + p(BAB) + p(BANAA) + p(BANAB) + \frac{1}{2} p(BANANA)$

$n = l(BANANA) = \lceil -\log_2 p(BANANA) \rceil + 1 = 10$ bits

$C(BANANA) = 1001000011$ → string comprimida

Teoria da Informação

Codificação Aritmética

Pros - Desempenho melhor que o do código de Huffman
Contras - Presume que os cálculos são feitos com precisão infinita
Não pode ser implementado como descrito

Algoritmo de Lempel e Ziv, 1977 (LZ77)

- é universal (não depende da fonte)
- Procura a maior string no lookAhead buffer que existe no buffer da esquerda
- Output gerado: (Ponteiro ①, Comprimento ②, NovoSimbolo ③)

① offset da string no buffer da esquerda
② comprimento da string a copiar do buffer
③ símbolo do lookAhead buffer seguinte à string encontrada

String: AABCBBABC
Codificação: (0,0,A) (1,1,B) (0,0,C) (2,1,B) (5,3, Eof)

- Se a freqência não for uniforme pode aplicar-se um algoritmo de compressão (ex Huffman)

Algoritmo Lempel e Ziv, 1978 (LZ78)

- código universal e código de dicionário
- Output: (W,S), palavra de código (índice no dicionário) e símbolo
- Em cada passo é adicionada uma palavra nova ao dicionário, que é a extensão de uma já existente caracterizada pelo par (W,S)
- Vantagem em relação ao 77: reduzido número de comparações necessárias

- Inicialmente dicionário vazio
- Lê sequência de símbolos até que não esteja no dicionário
- Adiciona sequência ao dicionário, atribuindo um novo índice
- W → índice do prefixo da sequência
- S → último símbolo da sequência, tornar não existente no dicionário

Input: ABBC BCABA

Output: (0, A)
(0, B)
(2, C)
(3, A)
(2, A)

0	""
1	A
2	B
3	BC
4	BCA
5	BA

Algoritmo Lempel - Ziv - Welch (LZW)

- melhoramento do LZ78
- foi patenteado, o que reduziu a sua adoção
- Output tem apenas palavras de código
- O dicionário é inicializado com todo o alfabeto
- Prefixo da nova sequência é o último símbolo da sequência anterior

Compressão:

- Prefixo vazio e dicionário contém todas as palavras
- Lê símbolo, P+S enquanto P+S está no dicionário
- Output - código de P
- Adiciona P+S ao dicionário
- Próximo prefixo é o último símbolo P=S

Input: AABABAABABAB

Output: 1
1
2
4
3
5
8

1	A
2	B
3	AA
4	AB
5	BA
6	ABA
7	AAB
8	BAB

Descompressão:

- Dicionário contém todos os símbolos; Atual é o primeiro número
- Lê número próximo, descodifica e output
- Adiciona ao dicionário Atual + 1º símbolo do próximo
- Atual ← Próximo

5

Teoria da Informação

Input: 1124358
Output: A
A
B
AB
AA
BA
BAB

1 A
2 B
3 AA
4 AB
5 BA
6 ABA
7 AAB
8 BAB

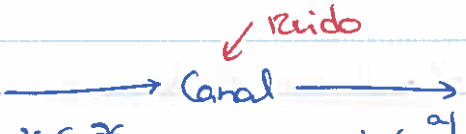
Se o 'próximo' se refere a uma linha não existente, o primeiro símbolo é igual ao primeiro símbolo de 'Atual'

Canais de Comunicação

1 - A fonte gera uma mensagem
2 - O codificador transforma a mensagem para a adaptar ao canal usado
3 - A mensagem codificada é enviada pelo canal, mas este não é fiável e pode comprometer a mensagem
4 - O descodificador recebe a mensagem do canal e tenta detetar erros de transmissão e corrigi-los
5 - A mensagem descodificada é entregue ao recetor

Canal discreto sem memória

alfabeto de entrada - $\mathcal{X}$
alfabeto de saída - $\mathcal{Y}$
probabilidades de transição $p(y|x)$



- A distribuição de probabilidade da saída depende da entrada e é condicionalmente independente das entradas e saídas passadas

Capacidade de um canal discreto sem memória

$$C = \max_{p(x)} I(X;Y)$$

- Um canal em que a saída Y é independente de X não permite transmitir
- Se é possível X e Y mais dependentes, a capacidade do canal aumenta
- Encontrar $p(x)$ que maximiza a informação mútua $I(X;Y)$ (optimização
- $C \geq 0$ porque $I(X;Y) \geq 0$
- $C \leq \log|\mathcal{X}|$ e $C \leq \log|\mathcal{Y}|$ porque $C = \max I(X;Y) \leq \max H(X) \leq \log|\mathcal{X}|$
- $I(X;Y)$ é uma função contínua e concava de $p(x)$

- Canal binário sem erros

$C = \max_{p(x)} I(X;Y) = 1$ bit ← capacidade do canal

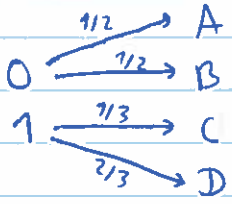
O máximo é atingido com $p(x) = \frac{1}{2}$

- Canal binário inverson

Capacidade do canal → $C = \max_{p(x)} I(X;Y) = 1$ bit

O máximo é atingido com $p(x) = \frac{1}{2}$
É análogo ao canal binário sem erros

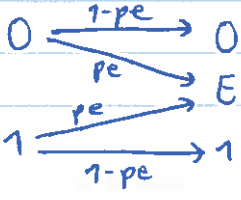
- Canal com saídas não sobrepostas



Capacidade do canal → $C = \max_{p(x)} I(X;Y) = 1$ bit

O máximo é atingido com distribuição uniforme $p(x) = \frac{1}{2}$
É semelhante ao canal binário sem erros

- Canal binário com perdas

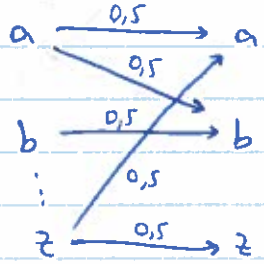


Capacidade do canal → $C = \max_{p(x)} I(X;Y) = 1 - pe$

O máximo é atingido com $p(x) = \frac{1}{2}$

Teoria da Informação

- Máquina de escrever ruidosa



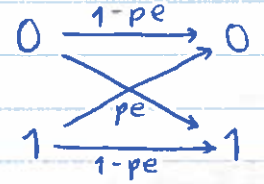
Capacidade do canal

$C = \max_{p(x)} I(X;Y) = \log 13$

O máximo é atingido com distribuição uniforme

$p(x) = \frac{1}{26}$

- Canal binário simétrico



Capacidade do canal → $C = \max_{p(x)} I(X;Y) = 1 - H(pe)$

$H(pe) = -pe \log_2 pe - (1-pe)\log_2(1-pe)$

O máximo é atingido com $p(x) = \frac{1}{2}$

- Canal simétrico

- Canal em que todas as linhas e colunas são permutações umas das outras

$$P = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_2 & p_3 & p_1 \\ p_3 & p_1 & p_2 \end{bmatrix}$$

Capacidade do canal → $C = \log|Y| - H(\text{"linha"})$

O máximo é atingido com distribuição uniforme

- O canal binário simétrico é um caso particular deste

- Canal fracamente simétrico

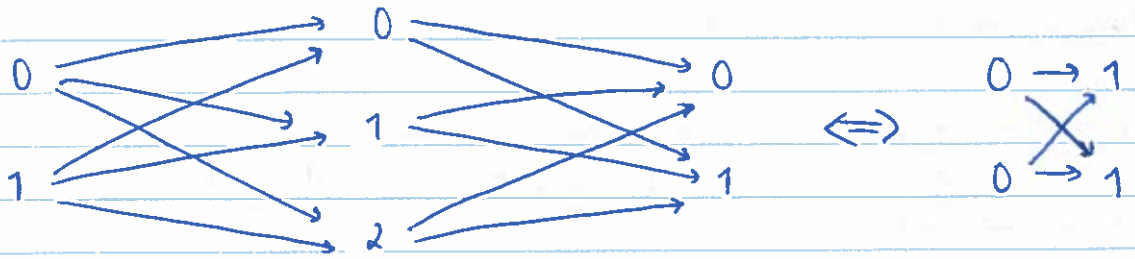
- Todas as linhas são permutações umas das outras e as colunas têm a mesma soma:

$\sum_x p(y|x)$

Capacidade do canal → $C = \log|Y| - H(\text{"linha"})$

O máximo é atingido com distribuição uniforme

- Concatenação de vários canais

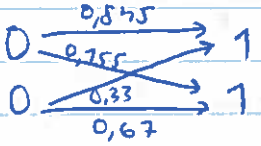


· É necessário calcular as probabilidades de transição do canal equivalente
· Se:

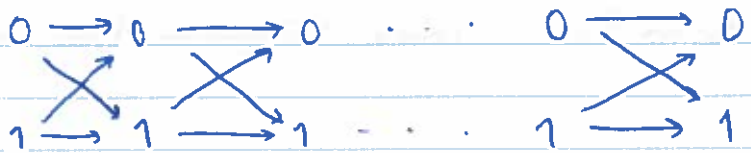
$$P_1 = \begin{bmatrix} 0,9 & 0,1 \\ 0,05 & 0,7 \\ 0,05 & 0,2 \end{bmatrix} \qquad P_2 = \begin{bmatrix} 0,9 & 0,2 & 0,5 \\ 0,1 & 0,8 & 0,5 \end{bmatrix}$$

As probabilidades de transição do canal equivalente são

$$P_{eq} = P_1 P_2 = \begin{bmatrix} 0,845 & 0,37 \\ 0,155 & 0,66 \end{bmatrix}$$



- Concatenação de canais idênticos



$$P_{eq} = P^n$$

· A concatenação de canais forma uma cadeia de Markov

Transmissão de informação por um canal ruidoso

· está sujeita a erros
· é possível construir códigos que permitam a deteção e correção de erros, mas requerem a transmissão de informação adicional
· a redundância baixa a probabilidade de erro, mas baixa o ritmo de transmissão, pois tem que transmitir mais símbolos pelo canal por cada símbolo da fonte
· É possível construir códigos com probabilidade de erro baixa se o ritmo de transmissão estiver abaixo

Teoria da Informação

Código para o canal $(\mathcal{X}, p(y|x), \mathcal{Y})$

- conjunto de índices $\{1, 2, \dots, M\}$, M possíveis mensagens a transmitir
- função de codificação $X^n : \{1, 2, \dots, M\} \to \mathcal{X}^n$ que fornece as palavras de código $x^n(1), x^n(2), \dots, x^n(M)$
- o índice (mensagem a transmitir) é codificado como uma sequência de n símbolos do alfabeto $\mathcal{X}$
- uma função de descodificação $g : \mathcal{Y}^n \to \{1, 2, \dots, M\}$ - adivinha o índice a partir da sequência de símbolos recebida y^n. A sequência y^n formada por n símbolos do alfabeto $\mathcal{Y}$ é descodificada pela função g que devolve um índice de $\{1, \dots, M\}$ correspondente à mensagem que se julga ter sido enviada

Probabilidade de Erro

- probabilidade condicional de erro dado que foi transmitido o símbolo i

$$\lambda_i = Pr\{g(Y^n) \neq i \mid X^n = x^n(i)\}$$

Probabilidade de erro máxima

- para um código de comprimento n é dada por:

$$\lambda^{(n)} = \max_{i \in \{1, \dots, M\}} \lambda_i$$

Ritmo de um código

- comprimento n para M índices

$$R = \frac{\log M}{n}$$

Ritmo atingível

- é possível construir uma sequência de códigos progressivamente maiores tal que a probabilidade de erro máxima $\lambda^{(n)}$ tende para 0 quando $n \to \infty$

Codificação de canal

- todos os ritmos abaixo da capacidade do canal são atingíveis. Para todos os ritmos $R < C$, é possível construir uma sequência de códigos progressivamente maiores tais que a probabilidade de erro máxima $\lambda^n \to 0$

Código de Hamming (7,4)

- permite corrigir até 1 erro numa palavra de código

dados	paridade
$2^m - 1 - m$ bits	m bits

Matriz de Hamming

$$H \triangleq \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

As palavras de código válidas tem que satisfazer $Hc = 0$

0 - 0000000	8 - 1000011
1 - 0001111	9 - 1001100
2 - 0010110	10 - 1010101
3 - 0011001	11 - 1011010
4 - 0100101	12 - 1100110
5 - 0101010	13 - 1101001
6 - 0110011	14 - 1110000
7 - 0111100	15 - 1111111

- A troca de um bit pode ser simulada fazendo um XOR bit a bit entre a palavra de código transmitida e um vetor de zeros onde o bit onde ocorre o erro é colocado a um

$$r = c \oplus e$$

- O descodificador usa esta equação para detetar erros $Hc \neq 0 \rightarrow$ erro
- Caso não ocorra erros na transmissão, a palavra recebida é igual à transmitida, $r = c$ e logo $Hr = 0$
- No caso de ocorrer um erro, a palavra recebida é $r = c \oplus e$

  $Hr = H(c \oplus e) = Hc \oplus He = 0 \oplus He = He$
- O resultado indica a posição onde ocorreu o erro
- A transmissão ocorre sem erros se não houve bits trocados e/ou houve um bit trocado

  $P_r\{\text{não erros}\} = (1-pe)^7 + 7(1-pe)^6 pe$