CS 8501 Advanced Topics in Machine Learning

Lecture 06: Information Theory Basics

Yangfeng Ji

Information and Language Processing Lab

Department of Computer Science

University of Virginia

https://yangfengji.net/

Entropy

Entropy

The entropy of a random variable X is defined as the average Shannon information content

$$H(X) = -\sum_{x} p(X = x) \log p(X = x)$$

Recall the definition of expectation, it can also be written as

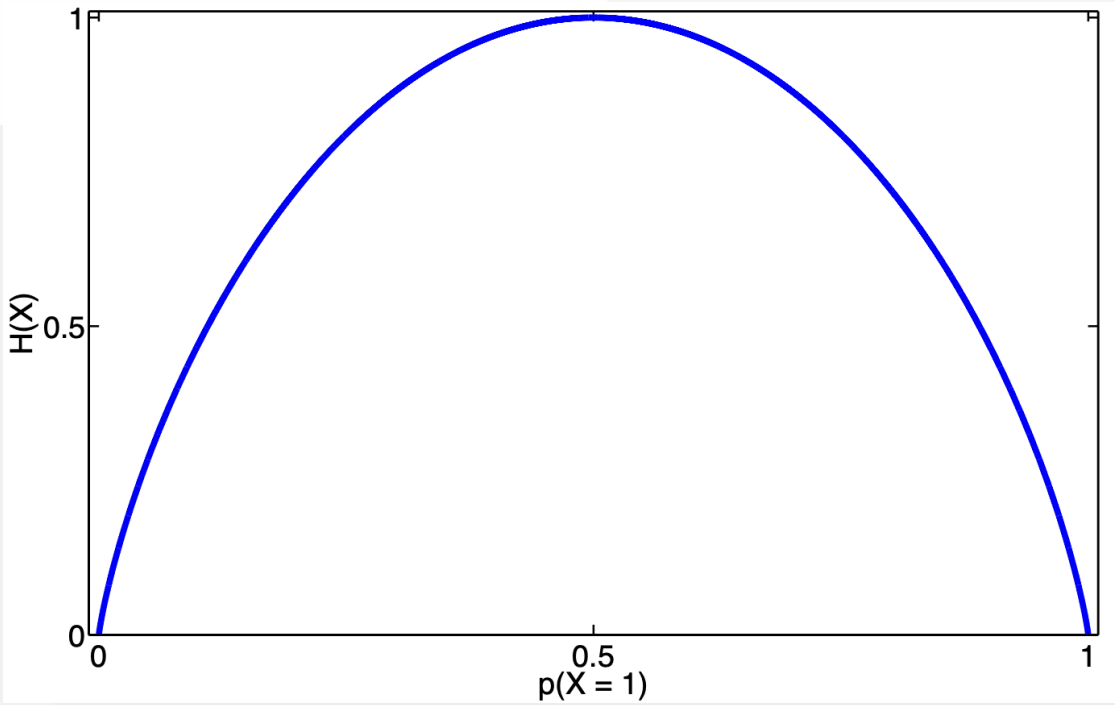
$$H(X) = E_p[-\log p(X)]$$

where $E_p[\cdot]$ is the expectation under the distribution p

Maximum Entropy

The discrete distribution with maximum entropy is the uniform distribution.

For Bernoulli distribution $p(X = x) = \theta^x(1 - \theta)^{1-x}$, the entropy $H(p)$ with respect to different θ



Properties of Entropy

- $H(X) \geq 0$ with equality if and only if $p(x) = 1$ for a specific $X = x$
- For discrete random variable, entropy is maximized if $p(X)$ is uniform

$$H(X) \leq \sum_{x} \frac{1}{K} \log K = \log K$$

  where K is the number of possible values that X can take

- Therefore, we have

$$0 \leq H(X) \leq \log K$$

Cross Entropy

The cross entropy between distribution p and q is defined by

$$H(p, q) = -\sum_{k=1}^{K} p_k \log q_k$$

where $p_k = p(X = k)$ and $q_k = q(X = k)$

- $H(p, q)$ is the expected number of bits needed to compress some data samples
  - from distribution p
  - using a code based on distribution q

Cross Entropy and MLE

Consider the following two distributions

- Empirical distribution $p(Y = y^* \mid x) = 1$, otherwise 0
- Predictive distribution $q(Y = y \mid x)$

The cross entropy of these two distributions is

$$H(p, q) = -\sum_k p(Y = k \mid x) \log q(Y = k \mid x) = -\log q(Y = y^* \mid x)$$

which is equivalent to the negative log likelihood

Joint Entropy

If X and Y follow the joint distribution $p(X, Y)$, then their entropy is defined as

$$H(X, Y) = -\sum p(X, Y) \log p(X, Y)$$

- In general, we have

  $$\max\{H(X), H(Y)\} \leq H(X, Y) = H(X) + H(Y \mid X) = H(Y) + H(Y \mid X) \leq H(X) + H(Y)$$

  which can be verified by the definition
- When $X \perp\!\!\!\perp Y$, we have

  $$H(X, Y) = H(X) + H(Y)$$

Conditional Entropy

The conditional entropy of Y given X is the uncertainty we have in Y after knowing X:

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

Intuitively,

$$H(Y|X) \leq H(Y)$$

with equality if and only if X and Y are independent

Chain Rule

The chain rule for entropy is

$$H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} H(X_i | X_1, \ldots, X_{i-1})$$

Perplexity

The **perplexity** of a discrete probability distribution p is defined as

$$\mathrm{PPLx}(p) = 2^{H(p)}$$

- $H(p)$: the average length of binary code that we need to represent message X
- $2^{H(p)}$: the essential number of messages in X

Perplexity: A Simple Example

Consider two extreme cases of the Bernoulli distribution

Case 1 $p(X = 1) = 1$ and $p(X = 0) = 0$

$$\mathrm{PPLx}(p) = 2^0 = 1$$

Case 2 $p(X = 1) = p(X = 0) = 0.5$

$$\mathrm{PPLx}(p) = 2^1 = 2$$

Perplexity: A Further Example

Consider a categorical distribution with the sample space size K, if

$$p(X = k) = \frac{1}{K}$$

then, the perplexity of this distribution is

$$\mathrm{PPLx}(p) = 2^{H(p)} = 2^{\log K} = K$$

On average, the model is perplexed with $\mathrm{PPLx}(p)$ of outputs

Perplexity in Language Modeling

- A language model is a probabilistic model that can predict the next word based on the preceding context $p(X_t|X_{1:t-1})$
- With a given text, we can evaluate the model performance with

$$p(x_t|x_{1:t-1}) = p(X_t = x_t|X_{1:t-1} = x_{1:t-1})$$

- The cross entropy (NLL) of the given text is $-\frac{1}{T}\sum_t \log p(x_t|x_{1:t-1})$
- The perplexity of a language model on the given text

$$1 \leq 2^{-\frac{1}{T}\sum_t \log p(x_t|x_{1:t-1})} \leq V$$

where V is the size of the sample space of X (aka, the vocabulary size)

Relative Entropy

Relative Entropy

The **relative entropy** or **Kullback-Leibler divergence** is to measure the difference between two distribution $p(X)$ and $q(X)$ defined on the same sample space

$$\mathrm{KL}(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

Similar to the definition of entropy, relative entropy can also be viewed as an expectation of function

$$\mathrm{KL} = E_p[\log \frac{p(x)}{q(x)}]$$

Relative Entropy and Cross Entropy

Relative entropy is the difference between cross entropy $H(p, q)$ and the entropy $H(p)$

$$\mathrm{KL}(p\|q) = E_p[\log \frac{1}{q(x)} - \log \frac{1}{p(x)}]$$

Therefore

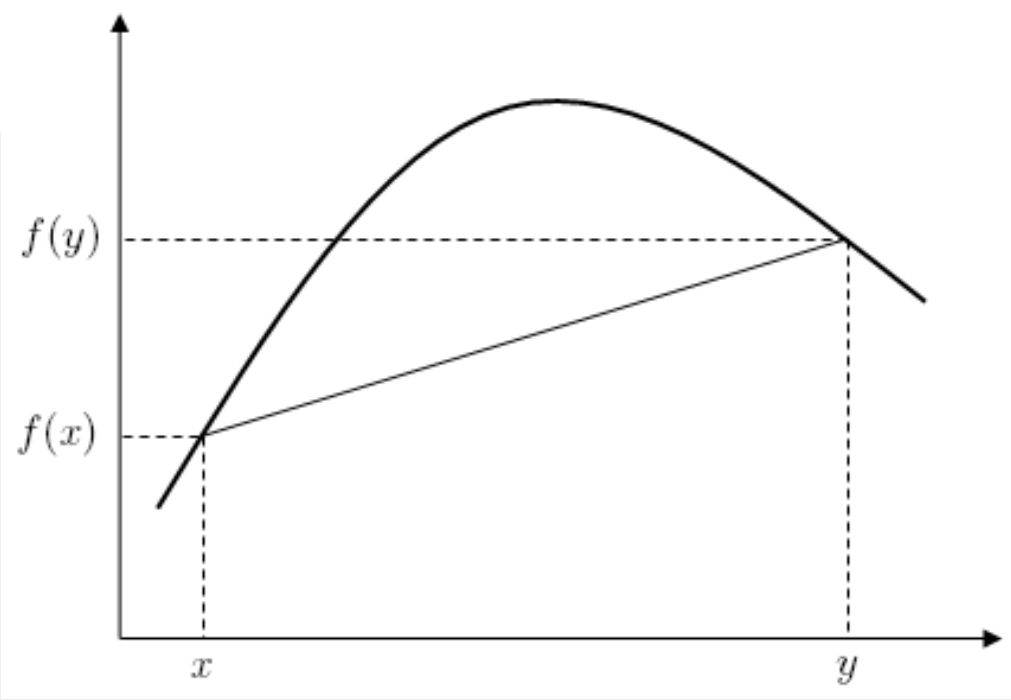
$$\mathrm{KL}(p\|q) = H(p, q) - H(p) \geq 0$$

This equation lays the foundation of **variational inference**, where q is the empirical distribution built over data

Convex Functions

A function $f(x)$ is concave over (a, b) if every chord of the function lies above the function. In other words, for all $x_1, x_2 \in (a, b)$ and $0 \leq \lambda \leq 1$, we have

$$f(\lambda x_1 + (1 - \lambda)x_2) \geq \lambda f(x_1) + (1 - \lambda)f(x_2)$$



Jensen's Inequality

For any concave function f,

$$f(\sum_{i=1}^{n} \lambda_i x_i) \geq \sum_{i=1}^{n} \lambda_i f(x_i)$$

where $\lambda_i \geq 0$ and $\sum_{i=1}^{n} \lambda_i = 1$

If $f(x) = \log(x)$,

$$\log E[f(x)] \geq E[\log f(x)]$$

Therefore,

$$\mathrm{KL}[p\|q] \geq 0$$

Forward vs Reverse KL

Consider the problem of approximating

- a distribution p with
- a *simpler* distribution q with either,

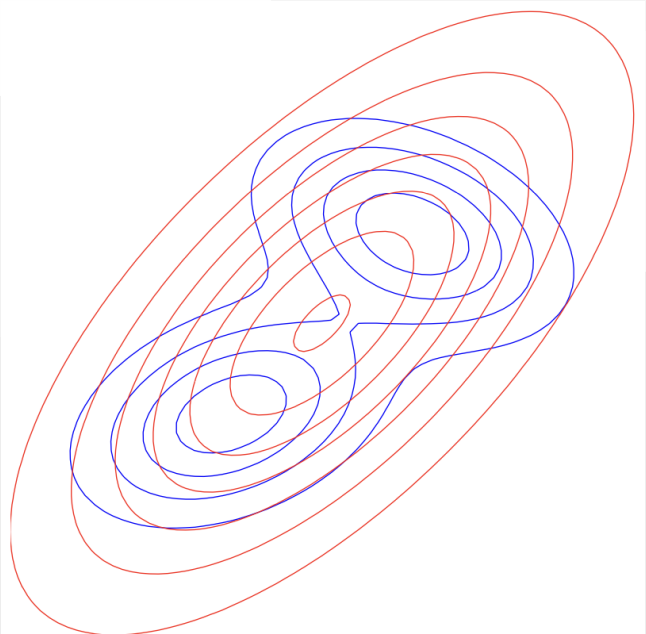
there are two ways to formulate this problem

- Forward KL: $\mathrm{KL}[p\|q] = \int p(x) \log \frac{p(x)}{q(x)} dx$
- Backward KL: $\mathrm{KL}[q\|p] = \int q(x) \log \frac{q(x)}{p(x)} dx$

Forward KL

$$\mathrm{KL}[p\|q] = \int p(x) \log \frac{p(x)}{q(x)} dx$$

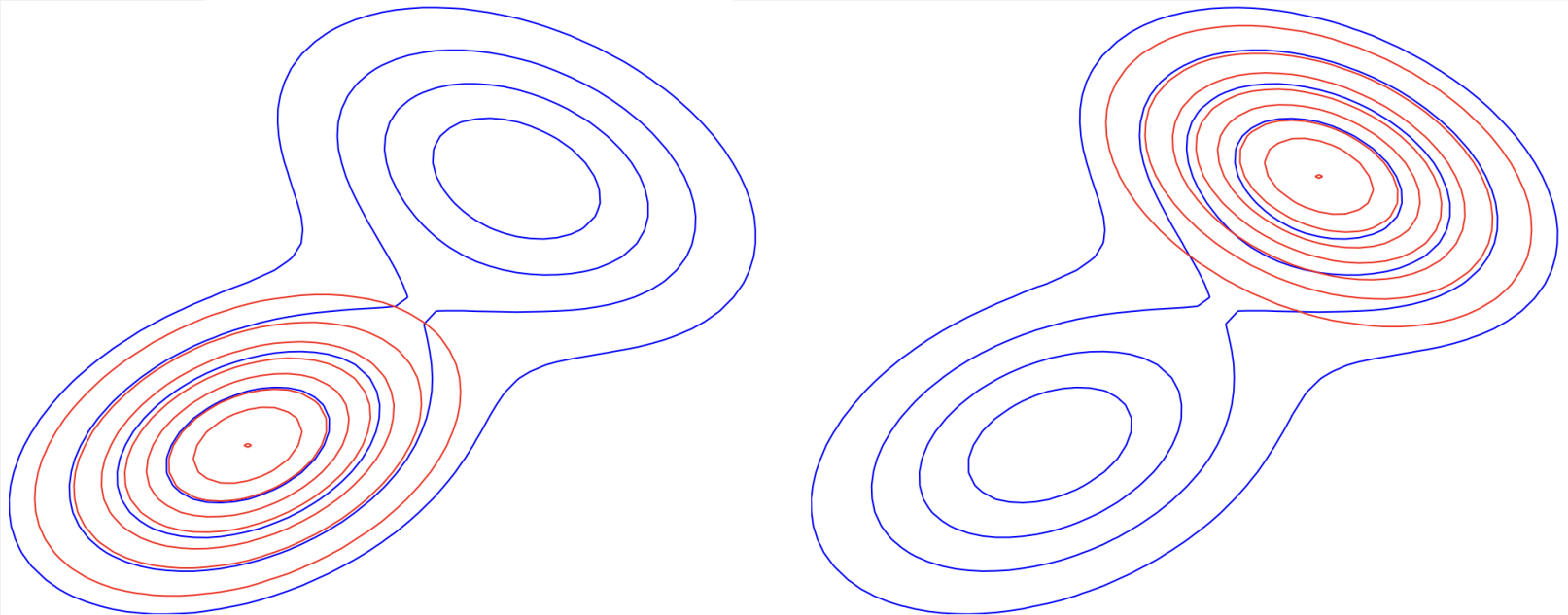
- Blue: true distribution p
- Red: approximation distribution q



Reverse KL

$$\mathrm{KL}[q\|p] = \int q(x) \log \frac{q(x)}{p(x)} dx$$

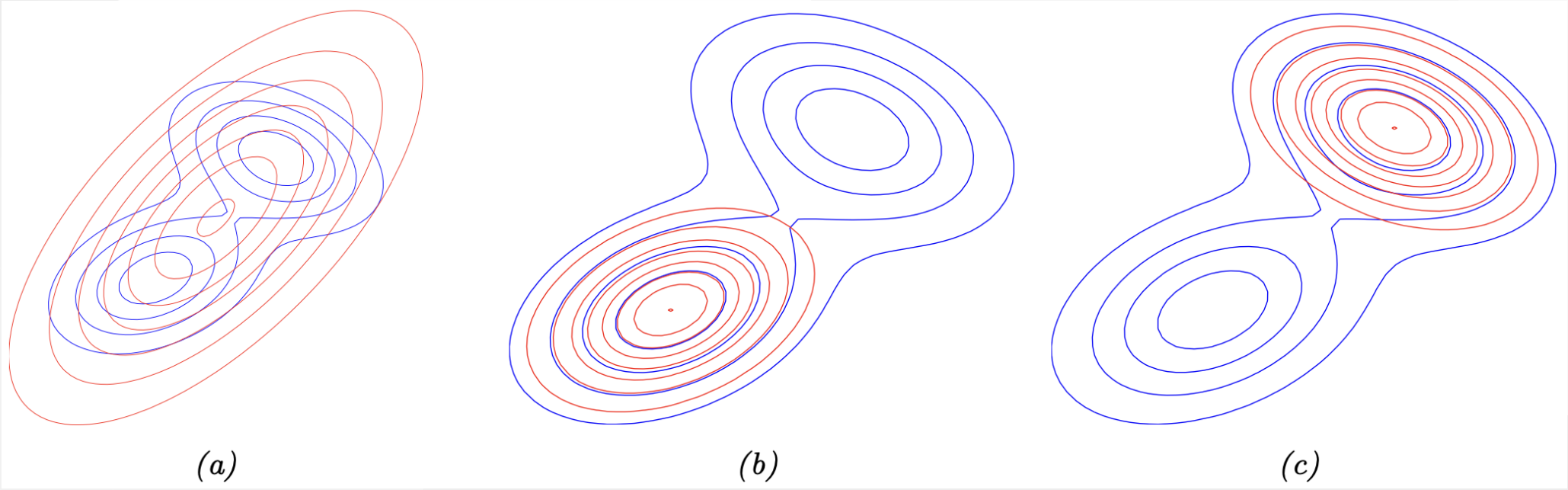
- Blue: true distribution p
- Red: approximation distribution q



Comparison

In practice, reverse KL is more popularly used, as

- it can identify the modes
- the expectation is easier to compute



Mutual Information

Definition

Mutual information between two random variable X and Y is defined as the difference between

- the joint distribution $p(X, Y)$
- the product of two marginal distributions $p(X)$ and $p(Y)$

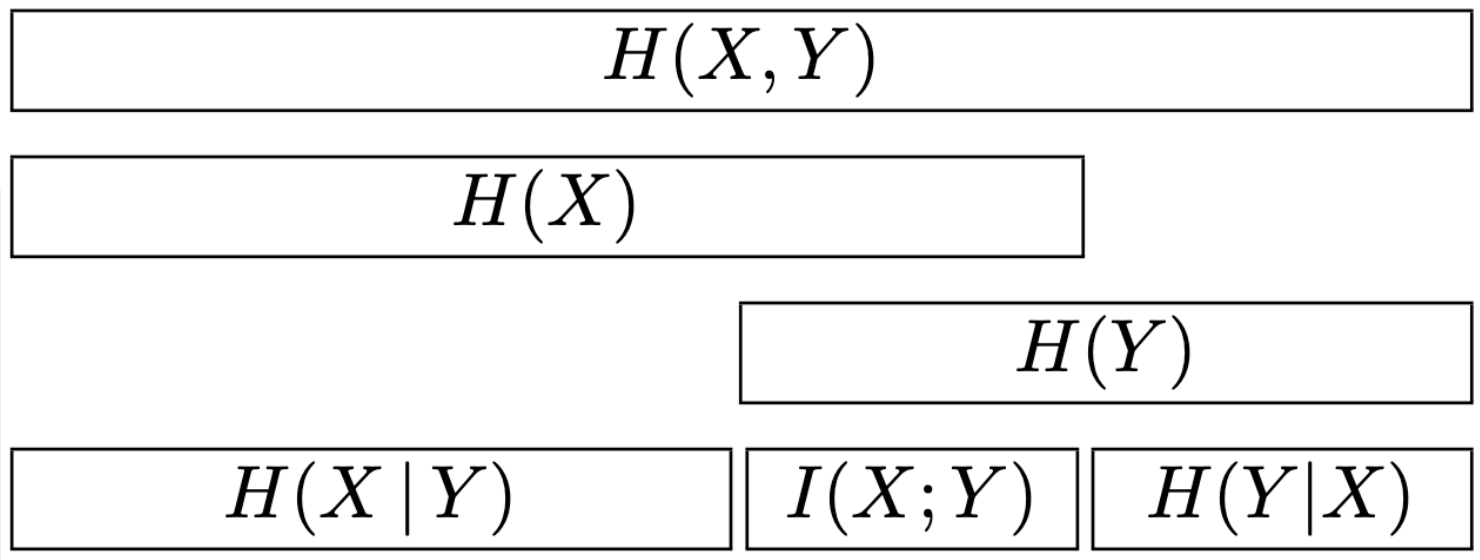
$$I(X;Y) = \mathrm{KL}[p(X,Y) \| p(X)p(Y)] = \sum_{x,y} p(X,Y) \log \frac{p(X,Y)}{p(X)p(Y)}$$

Properties

- $I(X;Y) \geq 0$, the equality holds iff $p(X,Y) = p(X)p(Y)$, in other words $X \perp Y$
- $I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$
- Consider a classification problem, where X is input and Y is output (label)
  - $H(Y)$: the uncertainty of Y
  - $H(Y|X)$: the remaining uncertainty of Y after knowing X
  - $I(X;Y)$: how much we know about Y after knowing X

Relationship

The relationship between joint entropy, marginal entropy, conditional entropy, and mutual information



The Data Processing Theorem

Consider the dependency relation specified by a Markov chain

$$X \to Z \to Y$$

the mutual information satisfies the following inequality

$$I(X, Y) \leq I(X, Z)$$

which is also called the data processing inequality.

The Data Processing Theorem (II)

$$I(X, Y) \leq I(X, Z)$$

- The message conveyed by this inequality implies that data processing can *only* destroy information
  - It is not necessarily a bad thing, if our goal is to predict Z
- For example, consider the following realization of the three random variables
  - X: the original text
  - Z: the bag-of-words representation of text X
  - Y: classification label about X

Information Bottleneck: Problem Definition

Based on (Tishby et al., 2000):

- X: the original signal
- Z: the quantization of X
- Y: the variable of interest

Expectation:

- Z to be as simple as possible
- Z capture as much of the information about Y as possible

Information Bottleneck: Formulation

The mathematical formulation of the information bottleneck is

$$\min I(X;Z) - \beta I(Z;Y)$$

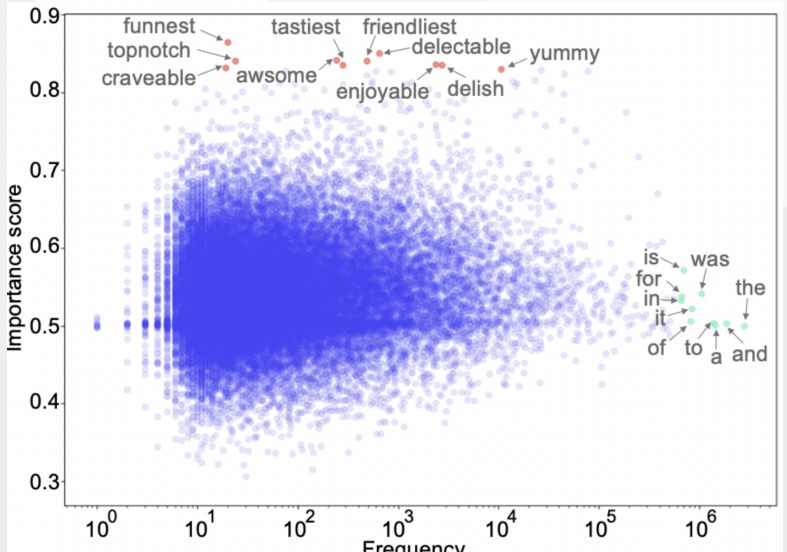
where β is a hyper-parameter

Interpretation: Z behaves as a bottleneck that filters information from X as much as possible, while keep the information useful for Y

- $\beta \to 0$
- $\beta \to \infty$

Information Bottleneck: Applications

- Information bottleneck formulation: $X \rightarrow Z \rightarrow Y$
- For every word x_t in the vocabulary, defined as $Z_t = R_{x_t} v_t$, where $R_{x_t} \in \{0, 1\}$ and v_{x_t} is the corresponding word embedding
- Information bottleneck can identify important words for the task and assign with high weights



Thank You!