Factor Graphs And Unsupervised Joke Generation

Alex Bolton

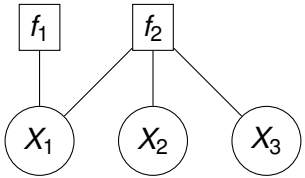
MathsJam Gathering 2019

Factor Graph

Definition

A factor graph is a graph that represents the factorisation of a function.

E.g. let $g(X_1, X_2, X_3) = X_1(X_1 + X_2 + X_3) = f_1(X_1)f_2(X_1, X_2, X_3)$:
Its factor graph is



Factor Graphs In Probability

Factor graphs can represent the factorisation of probabilities.

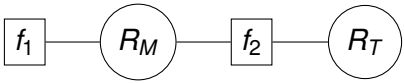
- E.g. joint probability of rain on Monday and on Tuesday (R_M and R_T).

$$P(R_M, R_T) = P(R_M)P(R_T \mid R_M)$$

or

$$P(R_M, R_T) = f_1(R_M)f_2(R_M, R_T)$$

So the factor graph looks like



Factor f_1 determines whether it rains on Monday, and factor f_2 determines if it rains on Tuesday given whether it rained on Monday.

Unsupervised Joke Generation

"Unsupervised Joke Generation From Big Data" (Petrović and Matthews) uses factor graphs to generate jokes of the form

$$\text{I like my } X \text{ like I like my } Y\text{: } Z.$$

Unsupervised Joke Generation

"Unsupervised Joke Generation From Big Data" (Petrović and Matthews) uses factor graphs to generate jokes of the form

$$\text{I like my } X \text{ like I like my } Y\text{: } Z.$$

- I like my women like I like my tea:

Unsupervised Joke Generation

"Unsupervised Joke Generation From Big Data" (Petrović and Matthews) uses factor graphs to generate jokes of the form

$$\text{I like my } X \text{ like I like my } Y\text{: } Z.$$

- I like my women like I like my tea: hot

Unsupervised Joke Generation

"Unsupervised Joke Generation From Big Data" (Petrović and Matthews) uses factor graphs to generate jokes of the form

$$\text{I like my } X \text{ like I like my } Y\text{: } Z.$$

- I like my women like I like my tea: hot
- I like my man like I like my homework:

Unsupervised Joke Generation

"Unsupervised Joke Generation From Big Data" (Petrović and Matthews) uses factor graphs to generate jokes of the form

$$\text{I like my } X \text{ like I like my } Y\text{: } Z.$$

- I like my women like I like my tea: hot
- I like my man like I like my homework: marked.

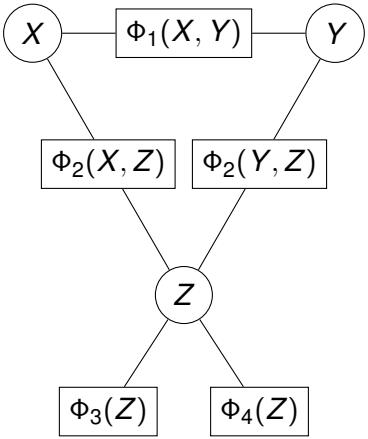
Unsupervised Joke Generation

The probability that a joke of the form

$$\text{I like my } X \text{ like I like my } Y\text{: } Z.$$

is funny is a function $p(X, Y, Z)$.

Unsupervised Joke Generation Factor Graph



The model presented as a factor graph.

$\Phi_1(X, Y)$

First factor encodes that dissimilar nouns lead to funnier jokes:

$$\Phi_1(x, y) = 1/\mathit{similarity}(x, y)$$

Without this assumption we get anti-jokes:

$\Phi_1(X, Y)$

First factor encodes that dissimilar nouns lead to funnier jokes:

$$\Phi_1(x, y) = 1/similarity(x, y)$$

Without this assumption we get anti-jokes:

I like my tea like I like my coffee:

$\Phi_1(X, Y)$

First factor encodes that dissimilar nouns lead to funnier jokes:

$$\Phi_1(x, y) = 1/\mathit{similarity}(x, y)$$

Without this assumption we get anti-jokes:

I like my tea like I like my coffee: hot.

$\Phi_2(X, Z)$

Second factor encodes that the joke is funnier if Z is often used to describe X:

$$\Phi_2(x, z) = \textit{co-occurrence}(x, z)$$

Without this assumption we get anti-jokes:

$\Phi_2(X, Z)$

Second factor encodes that the joke is funnier if Z is often used to describe X:

$$\Phi_2(x, z) = \textit{co-occurrence}(x, z)$$

Without this assumption we get anti-jokes:

I like my teapot like I like my fireplace:

$\Phi_2(X, Z)$

Second factor encodes that the joke is funnier if Z is often used to describe X:

$$\Phi_2(x, z) = \textit{co-occurrence}(x, z)$$

Without this assumption we get anti-jokes:

I like my teapot like I like my fireplace: crazy.

$\Phi_3(Z)$

Third factor encodes that the joke is funnier if Z is not a common adjective:

$$\Phi_3(z) = 1/\mathit{frequency}(z)$$

Without this assumption we get anti-jokes:

$\Phi_3(Z)$

Third factor encodes that the joke is funnier if Z is not a common adjective:

$$\Phi_3(z) = 1/\mathit{frequency}(z)$$

Without this assumption we get anti-jokes:

I like my dogs like I like my quality of life:

$\Phi_3(Z)$

Third factor encodes that the joke is funnier if Z is not a common adjective:

$$\Phi_3(z) = 1/\mathit{frequency}(z)$$

Without this assumption we get anti-jokes:

I like my dogs like I like my quality of life: good.

$\Phi_4(Z)$

Final factor encodes that the joke is funnier if Z has many different senses, because the humour often comes from Z being used on one sense for X and another for Y:

$$\Phi_4(z) = \mathit{senses}(z)$$

Without this assumption we get anti-jokes:

Final factor encodes that the joke is funnier if Z has many different senses, because the humour often comes from Z being used on one sense for X and another for Y:

$$\Phi_4(z) = \mathit{senses}(z)$$

Without this assumption we get anti-jokes:

I like my horror films like I like my murder:

$\Phi_4(Z)$

Final factor encodes that the joke is funnier if Z has many different senses, because the humour often comes from Z being used on one sense for X and another for Y:

$$\Phi_4(z) = \mathit{senses}(z)$$

Without this assumption we get anti-jokes:

I like my horror films like I like my murder: gruesome.

- Use Google n-gram data to estimate co-occurrence, etc.
- Use Monte Carlo to generate potential jokes.
- 16% of the jokes were found funny!

Two of the jokes:

- Use Google n-gram data to estimate co-occurrence, etc.
- Use Monte Carlo to generate potential jokes.
- 16% of the jokes were found funny!

Two of the jokes:

I like my coffee like I like my war:

- Use Google n-gram data to estimate co-occurrence, etc.
- Use Monte Carlo to generate potential jokes.
- 16% of the jokes were found funny!

Two of the jokes:

I like my coffee like I like my war: cold.

- Use Google n-gram data to estimate co-occurrence, etc.
- Use Monte Carlo to generate potential jokes.
- 16% of the jokes were found funny!

Two of the jokes:

I like my coffee like I like my war: cold.
I like my relationships like I like my source:

- Use Google n-gram data to estimate co-occurrence, etc.
- Use Monte Carlo to generate potential jokes.
- 16% of the jokes were found funny!

Two of the jokes:

I like my coffee like I like my war: cold.
I like my relationships like I like my source: open.

Summary

- Factor graphs can represent probabilities.
- They can also be used to help generate jokes!