Computational morphology reading group

Intrinsic Probing through Dimension Selection

Lucas Torroba Hennigen **Adina Williams** **Ryan Cotterell**

Québec Artificial Intelligence Institute (Mila) University of Cambridge

Facebook AI Research ETH Zürich

lucas.torroba-hennigen@mila.quebec, adinawilliams@fb.com,
ryan.cotterell@inf.ethz.ch

June 4, 2021

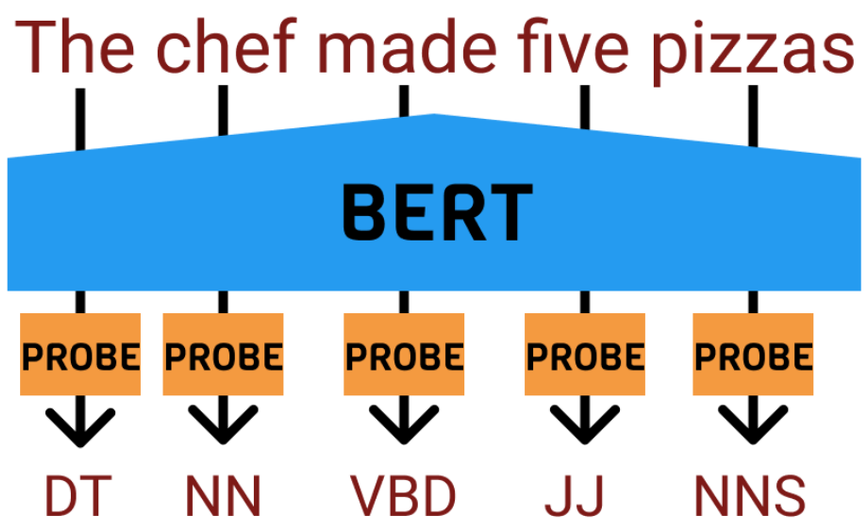
Miikka Silfverberg

Probing representations for linguistic features

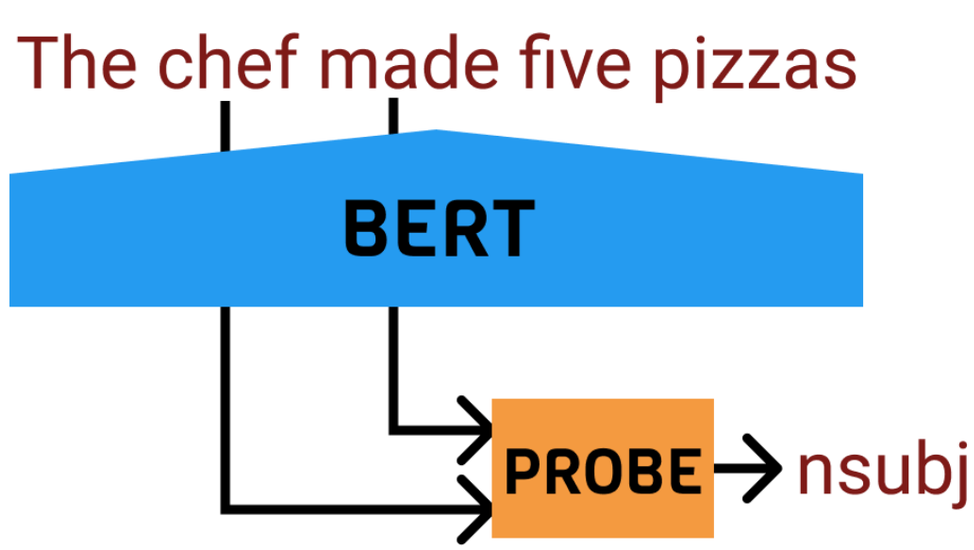
For each word in our sentence, BERT generates a 768-dimensional vector which encodes information about the word

We don't understand how and what information is encoded :(

Part-of-speech!

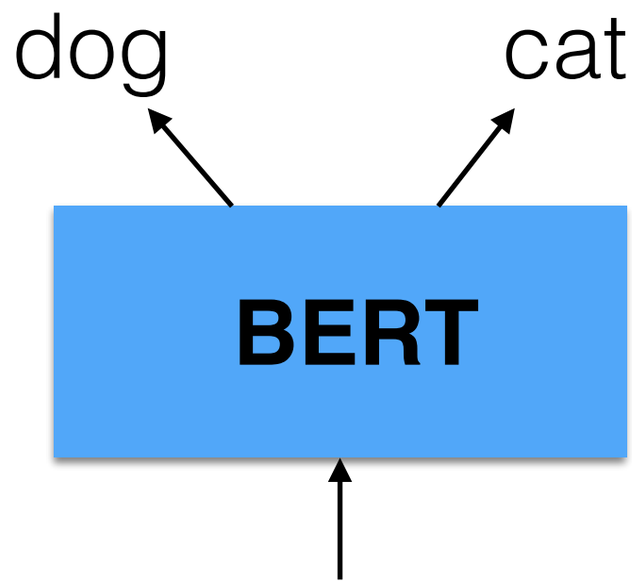


Partial dependency info!



Train a classifier to infer linguistic features from representations

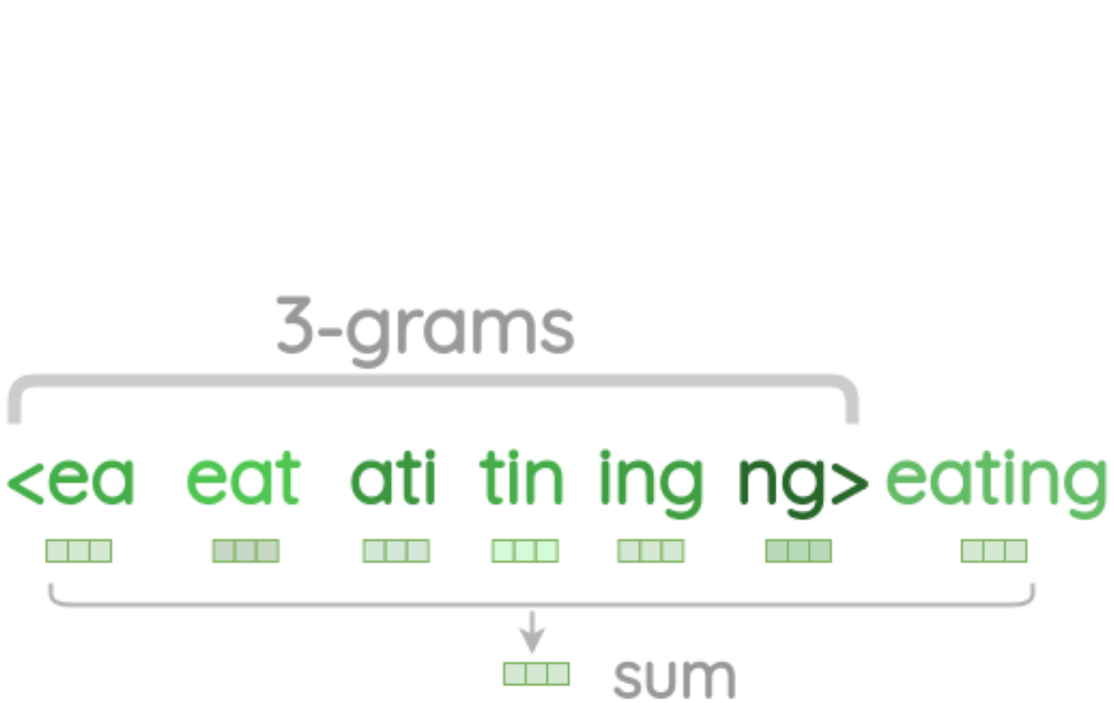
Multilingual BERT embeddings

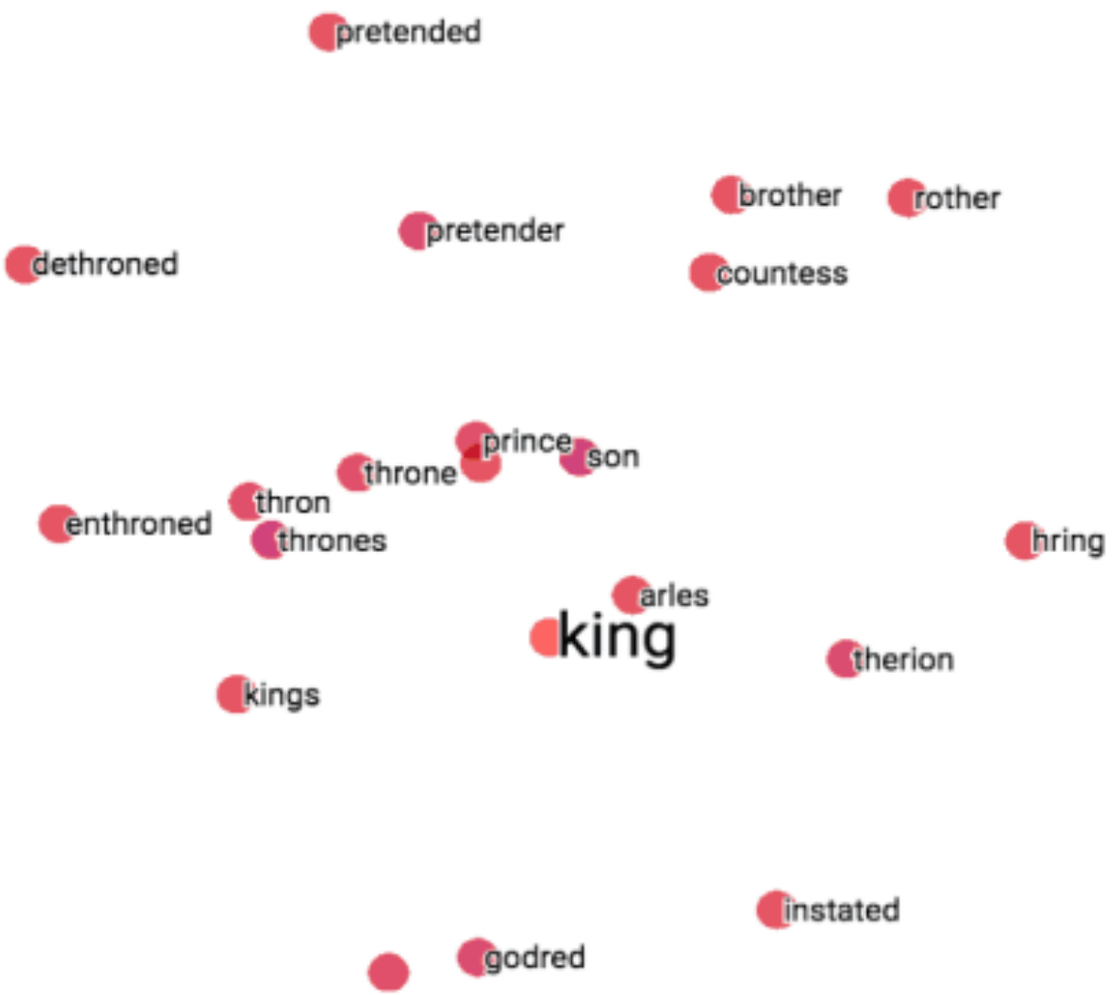


768-dimensional representations jointly trained on 36 languages

They used the final BERT layer

Multilingual fastText embeddings





768-dimensional representations(**?**) jointly trained on 36 languages

Extrinsic probing

We train a classifier which takes representations as input and predicts linguistic features from the representation vectors

How well can the features be predicted from the representations?

Attribute: TENSE, Value: PAST

Intrinsic probing

We train a classifier which takes representations as input and predicts linguistic features from the representation vectors

How well can the features be predicted from the representations?

Which dimensions in the representation encode this information?

How is the information represented?

Attribute: TENSE, Value: PAST

Research question

How do BERT and fastText encode morphosyntactic information?

Which of the models encodes morphosyntactic features using fewer dimensions on average?

Spoiler: fastText does

Probing individual dimensions

The simplest approach is to train a classifier with a single feature: our activation for dimension d

We can then find dimensions which can predict a particular feature

Probing subsets of dimensions is (NP-)hard

Likewise we can easily train a classifier for a fixed subset of features

However, there are prohibitively many subsets to train classifiers for all of them

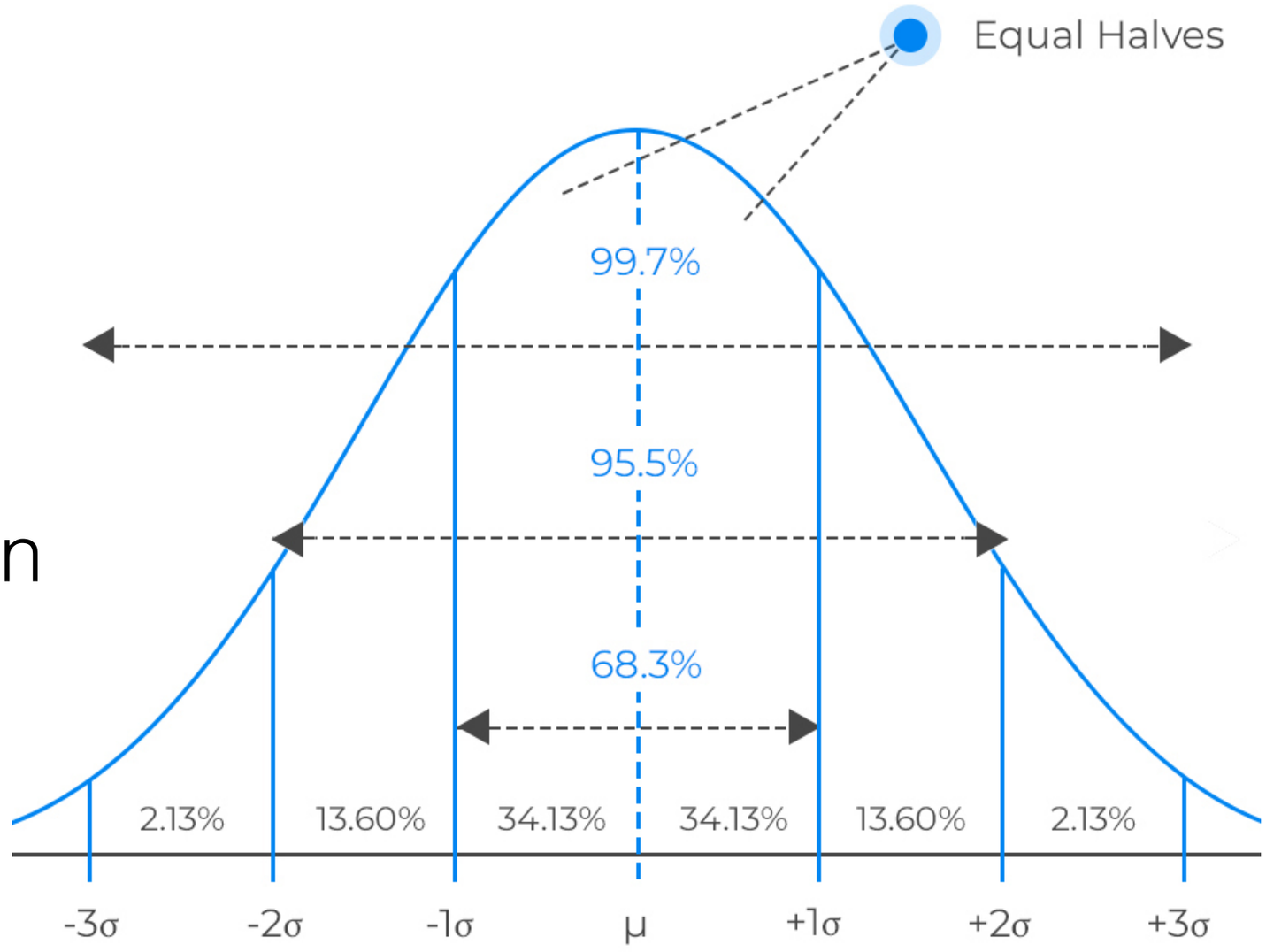
E.g. comb(768,10) ~ $1.85*10^{22}$

Let’s discuss the greedy approach

Gaussian distribution

Parametrized by:

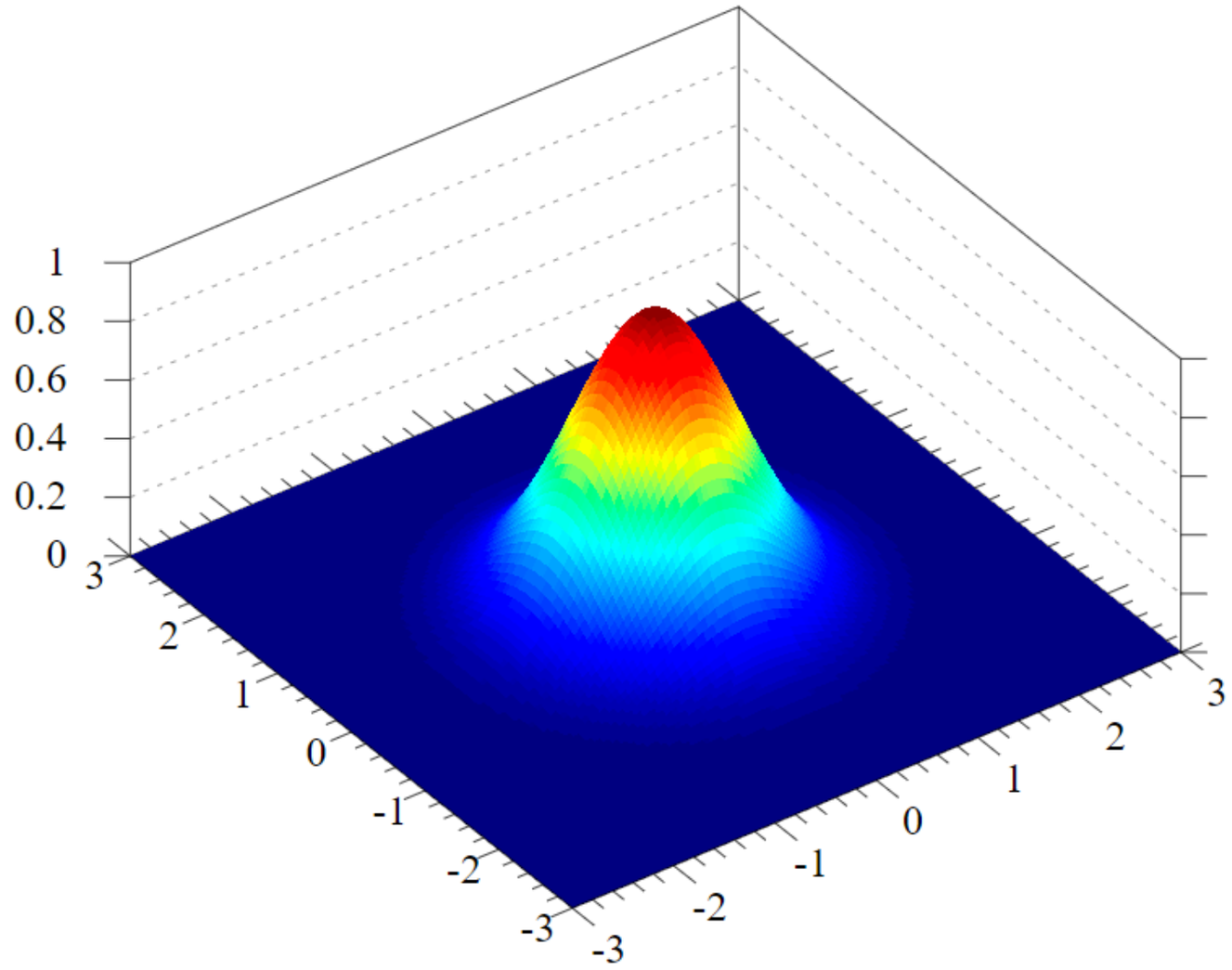
μ - mean
σ - standard deviation



Multivariate Gaussian

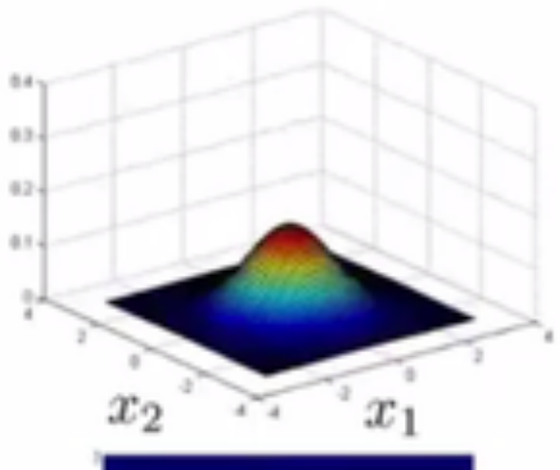
Parametrized by:

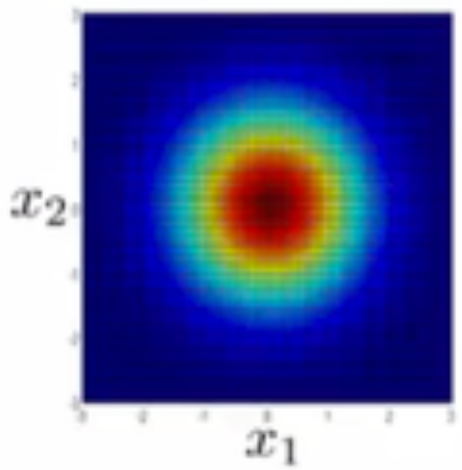
μ - mean vector
Σ - covariance matrix



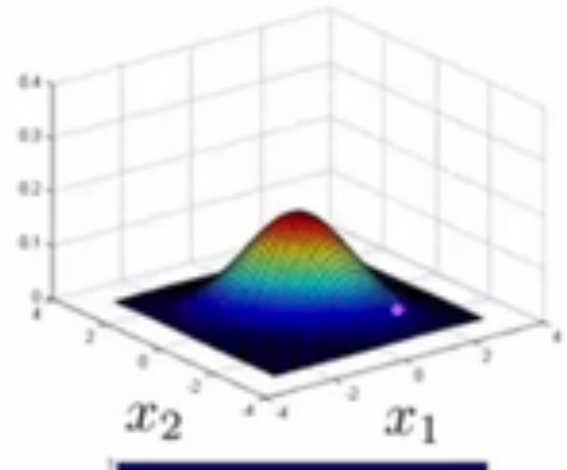
Multivariate Gaussian

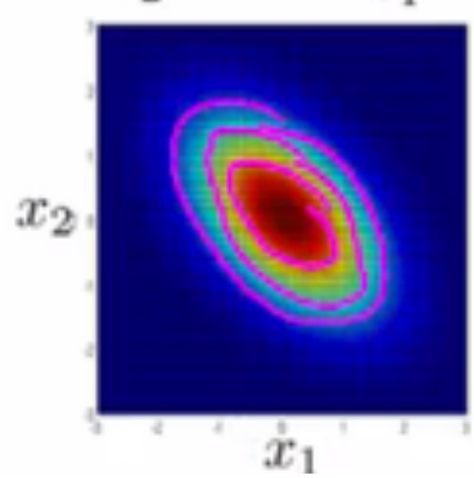
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



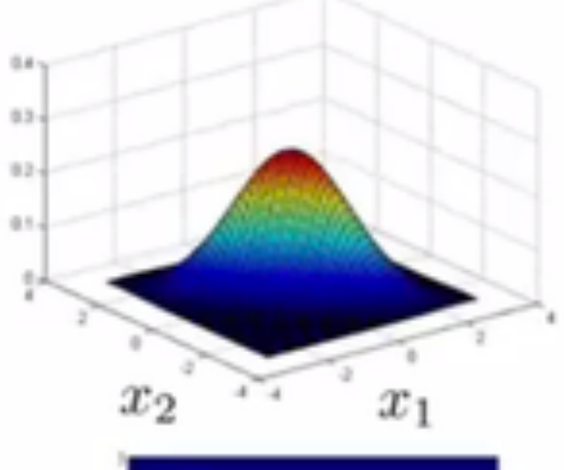


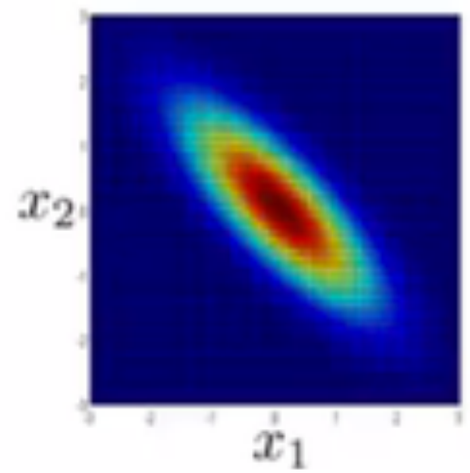
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$





$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$



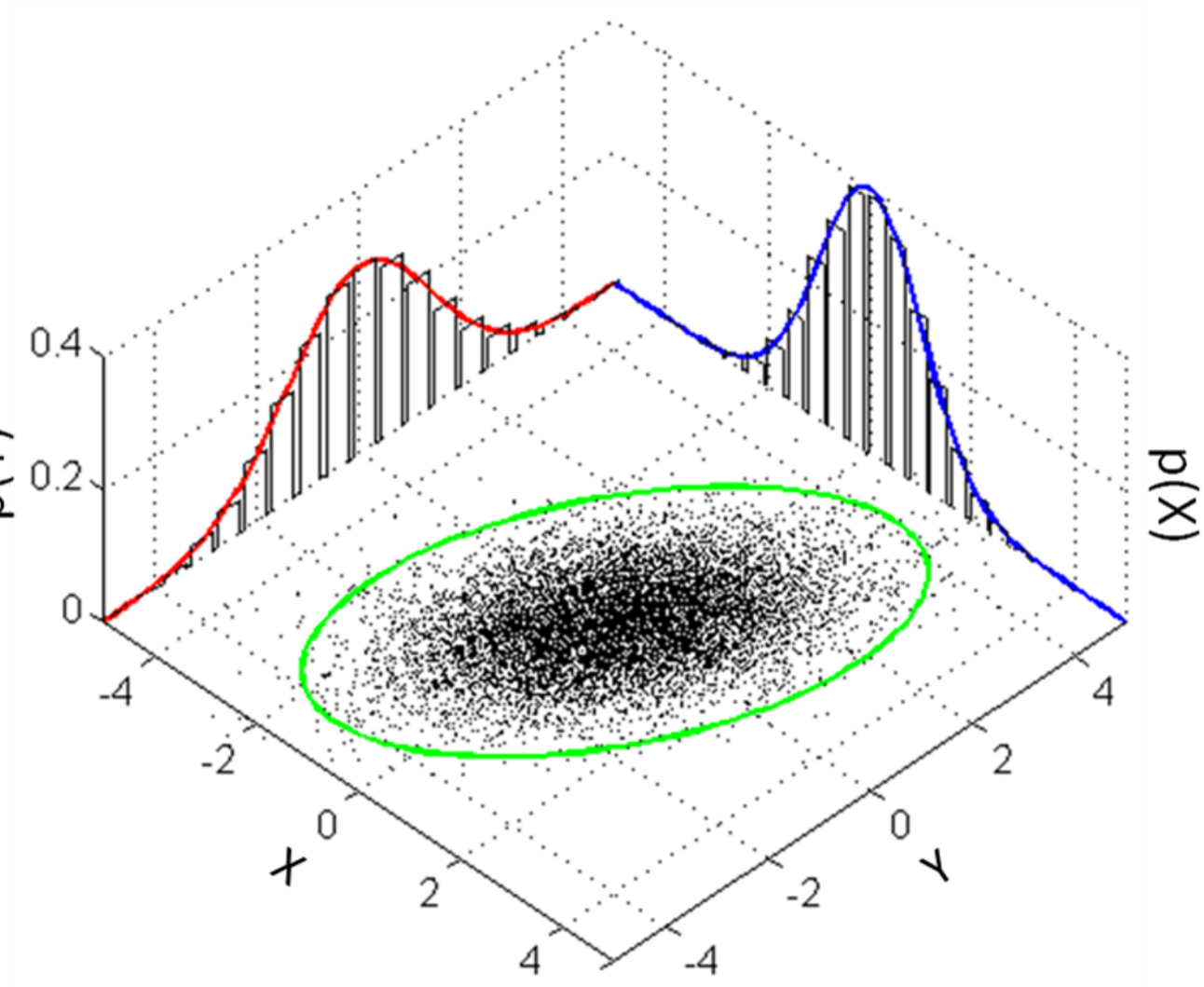


Decomposability

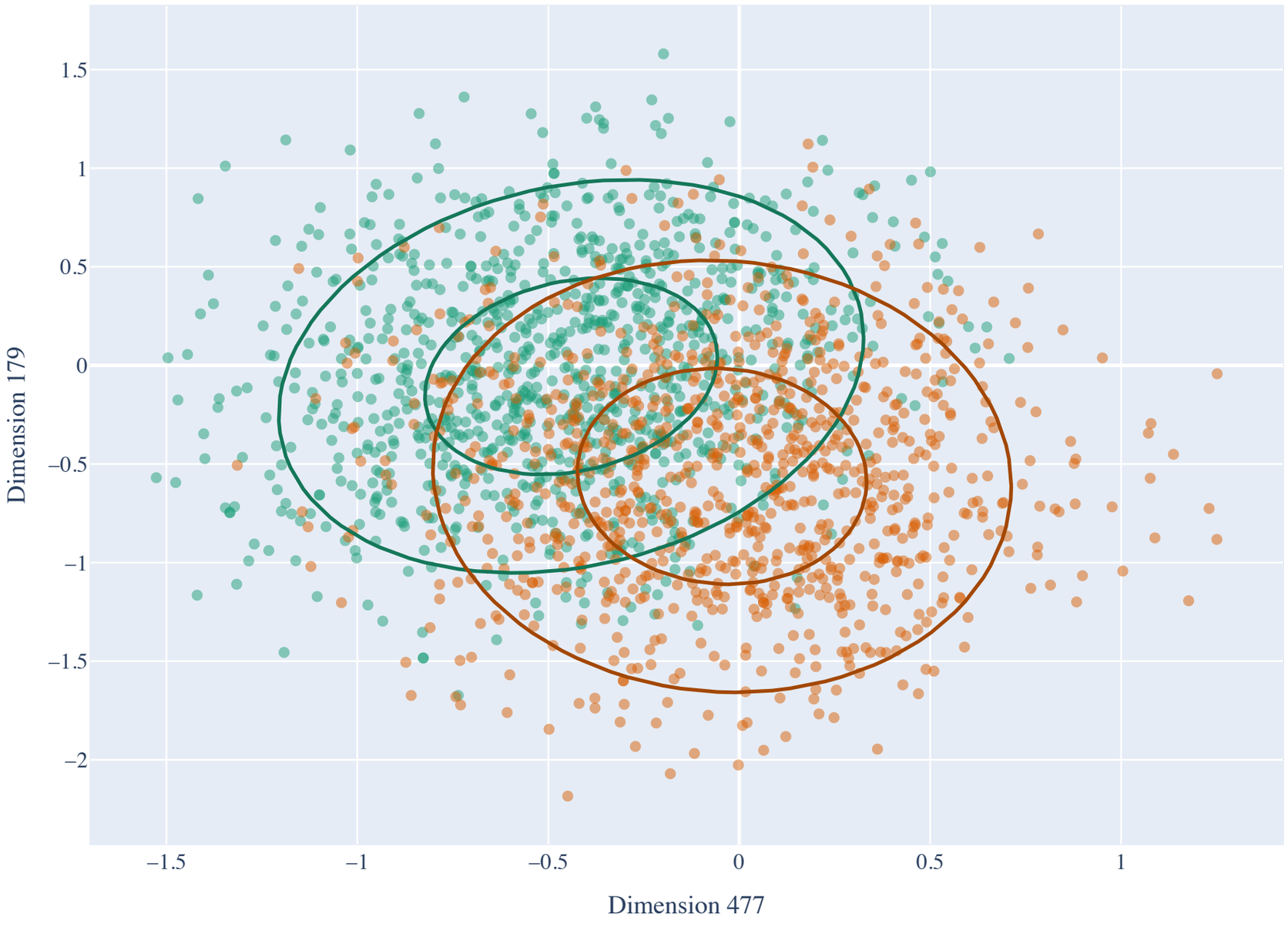
If we know **μ** and **Σ** for our 768-dimensional distribution,

we can very easily generate $\boldsymbol{\mu}_C$ and $\boldsymbol{\Sigma}_C$ for any subset C of those dimensions

By training one probe, we essentially train a whole set of probes



Informative dimensions for PAST and PRES (English)



Gaussian probe

$$p(\boldsymbol{h}, v) = p(\boldsymbol{h} \mid v)\, p(v) \qquad p(\boldsymbol{h} \mid v) = \mathcal{N}(\boldsymbol{h} \mid \boldsymbol{\mu}_v, \Sigma_v)$$

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma) = \tag{2}$$

$$|2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

*"the closer to the mean, the more likely **x** will be"* (+ covariance)

Classification using our probe:

Find value *v* for attribute a which maximizes $p(\boldsymbol{h}, v)$ for given ***h***

(or $p(v|\boldsymbol{h})$)

Gaussian probe

$$p(\boldsymbol{h}, v) = p(\boldsymbol{h} \mid v)\, p(v) \qquad p(\boldsymbol{h} \mid v) = \mathcal{N}(\boldsymbol{h} \mid \boldsymbol{\mu}_v, \Sigma_v)$$

For every attribute-value pair *a=v* (e.g. TENSE=PAST), we need to estimate the mean $\boldsymbol{\mu}_v$ and covariance $\boldsymbol{\Sigma}_v$

p(v) expresses how often attribute *a* gets value *v*

Why use a Gaussian probe?

Among all models with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$,
the Gaussian has the maximal entropy (uncertainty)

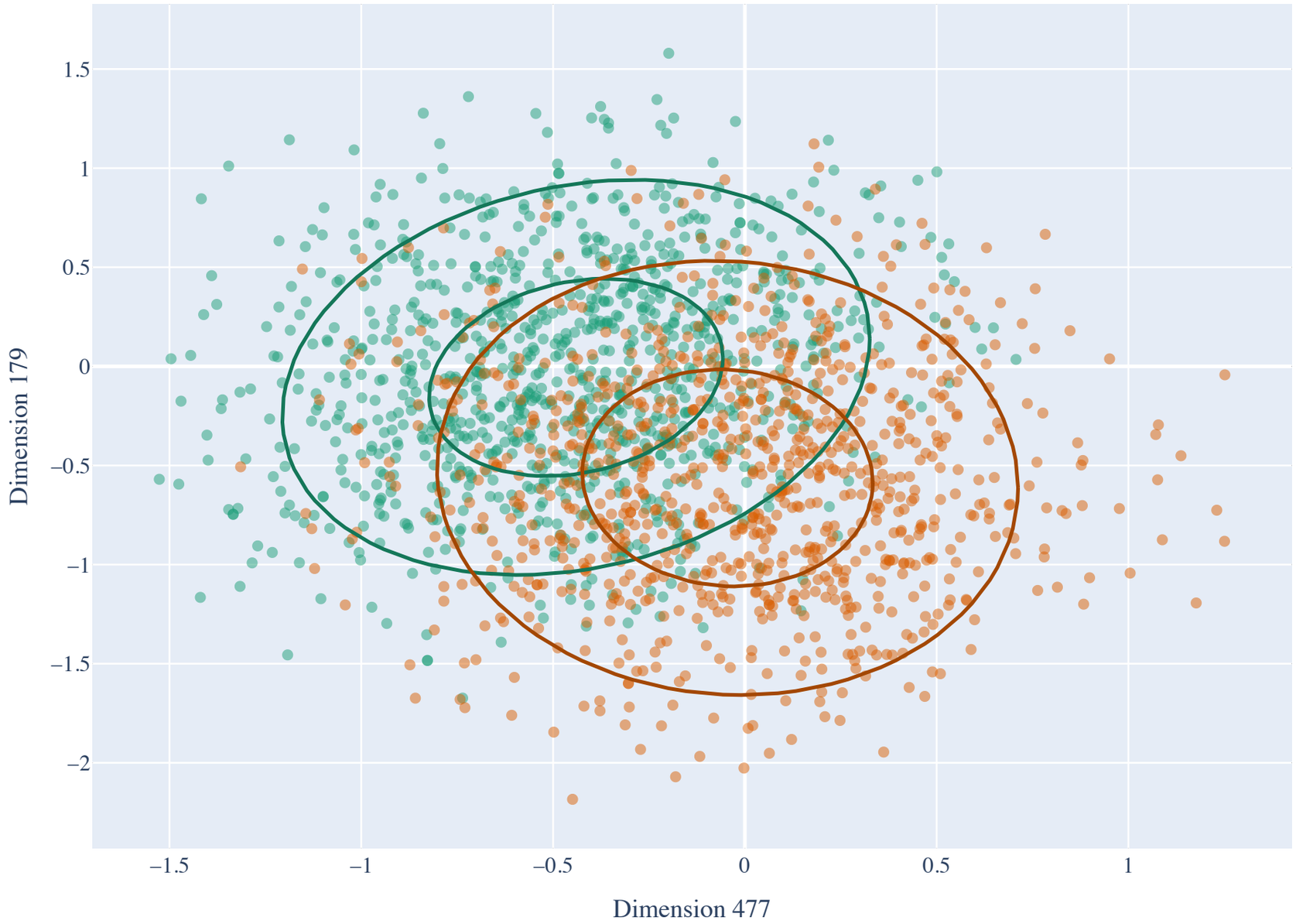
=

It assumes the least amount of additional information about the data

=

It is the least biased model

Gaussian probe



It seems that dimensions 477 and 179 behave like Gaussians but these were found by the Gaussian probe!

Simple way to estimate using empirical counts

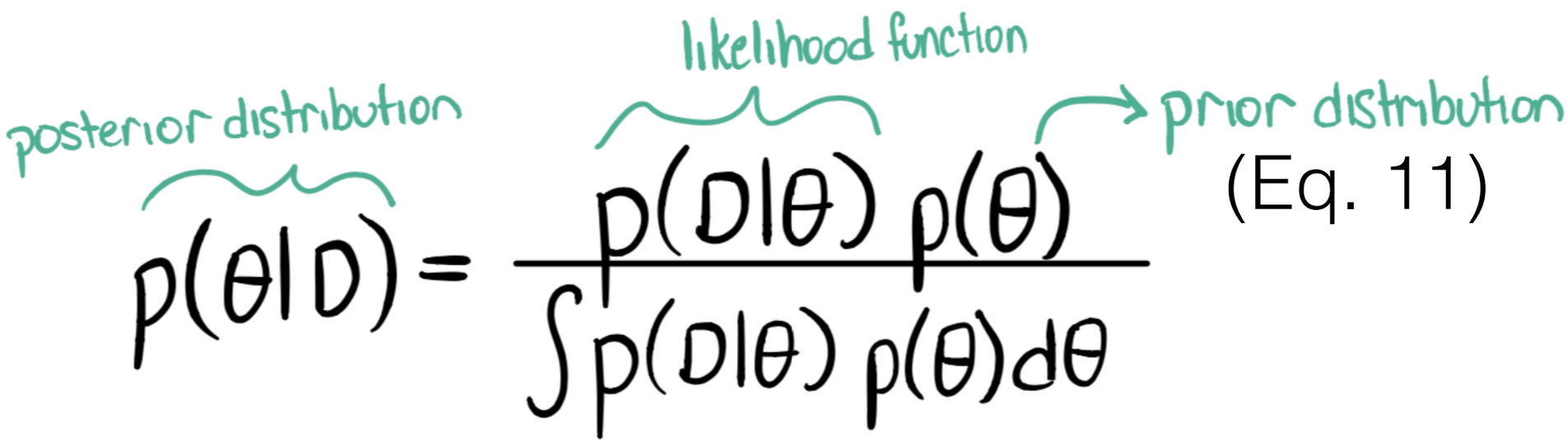
$$\widehat{\mu} = \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i. \qquad \hat{\mathbf{\Sigma}} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(x_i - \overline{x})^{\mathrm{T}},$$

$\boldsymbol{X}_i$ is a *d*-dimensional representation having value *v*

This works poorly when there are less data points than dimensions (degenerate Gaussians)

MAP estimation

(Eq. 6,7,8)



(Eq. 11)

$\theta = \{\boldsymbol{\mu}_V, \boldsymbol{\Sigma}_V\}$

MAP:
Just apply argmax_θ to $p(\theta|D)$.
No need to compute the integral

Using the Gaussian-inverse-Wishart distribution (GIW) as the prior,
we get a nice closed form solution in Eqs. 13 & 14

Evaluation: accuracy

Compare probing accuracy given a subset of dimensions to the majority-class baseline

Majority-class baseline:

Just predict the most common attribute value

Lower-bound accuracy (LBA) of a set of dimensions C:

the highest accuracy achieved by any subset C’ ⊆ C

Evaluation: Mutual information

How much do we know about the attribute value given some subset of embedding dimensions?

$$\mathrm{MI}(V_a; H) = \mathrm{H}(V_a) - \mathrm{H}(V_a \mid H)$$ (always >= 0)

V_a is the set (r.v.) of all values for an attribute *a* (like TENSE)

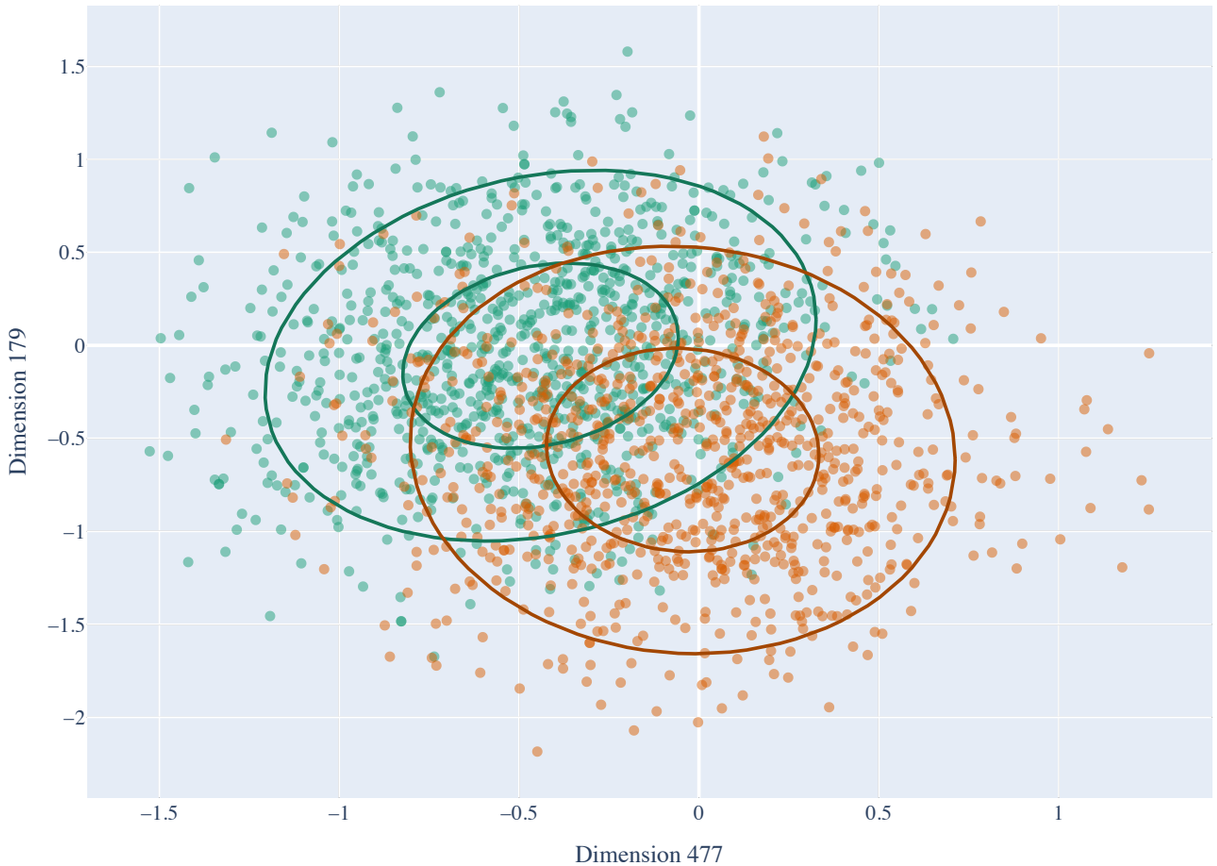
H is a set (r.v.) of embedding dimensions

H(…) is entropy

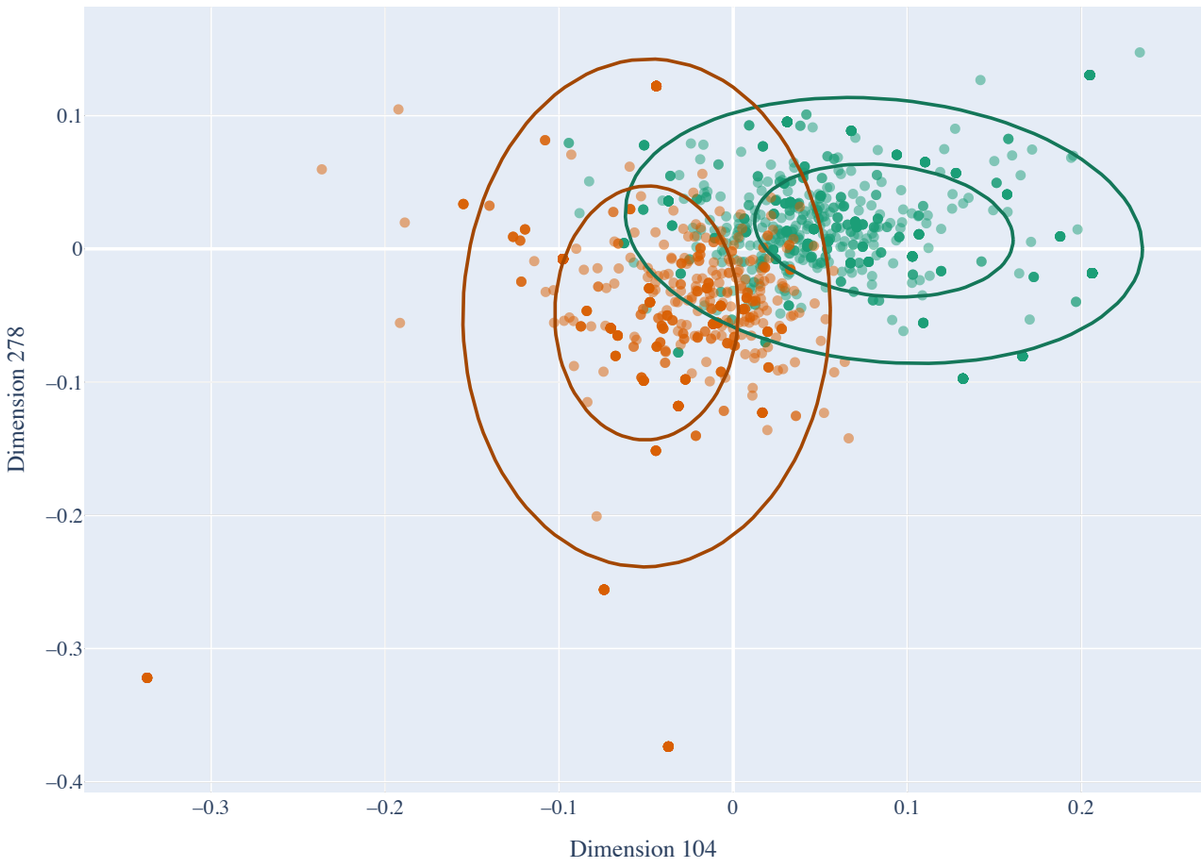
Results

Generally, very few neurons are needed to encode for a given property like TENSE

Results: fastText vs. BERT



BERT



fastText

Results: fastText vs. BERT

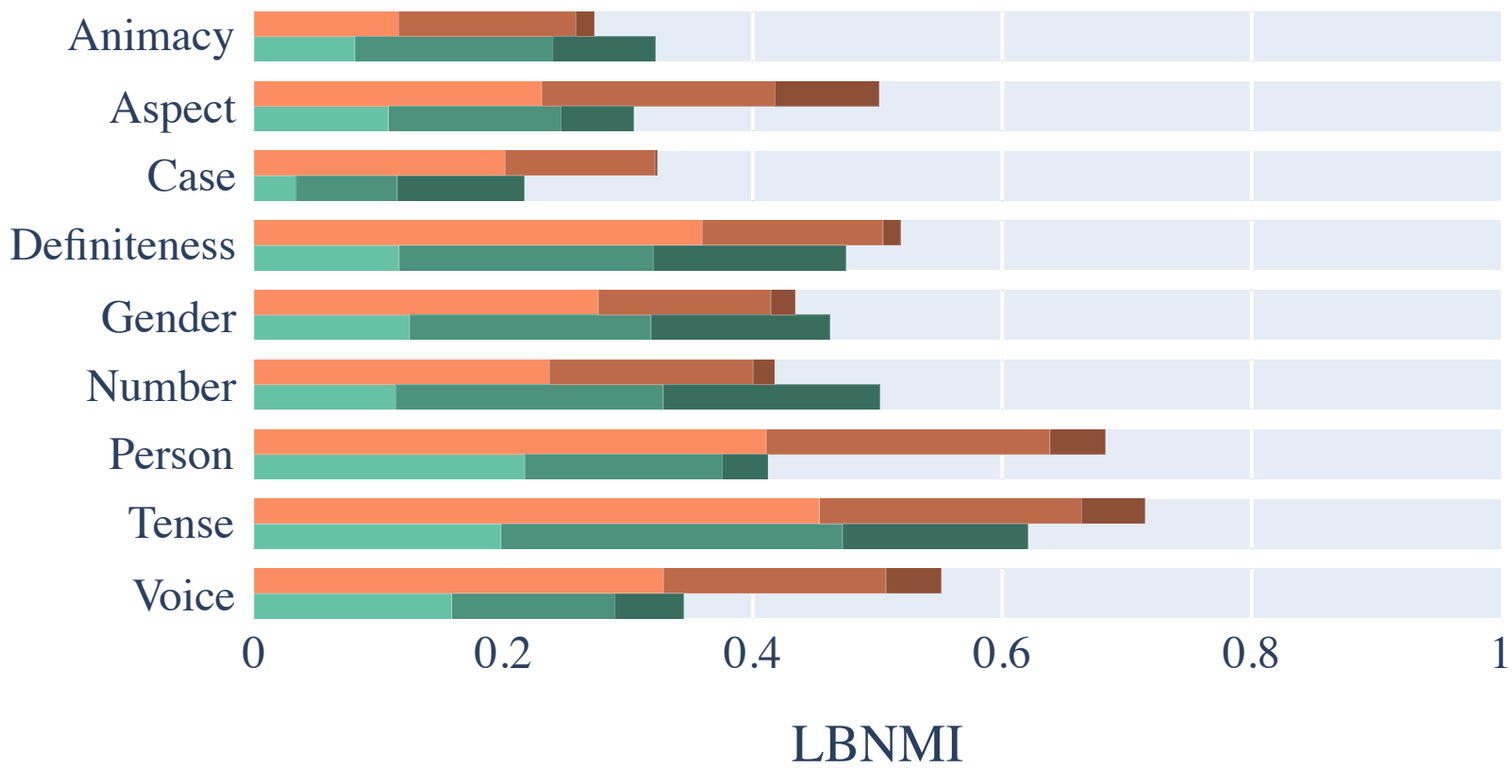
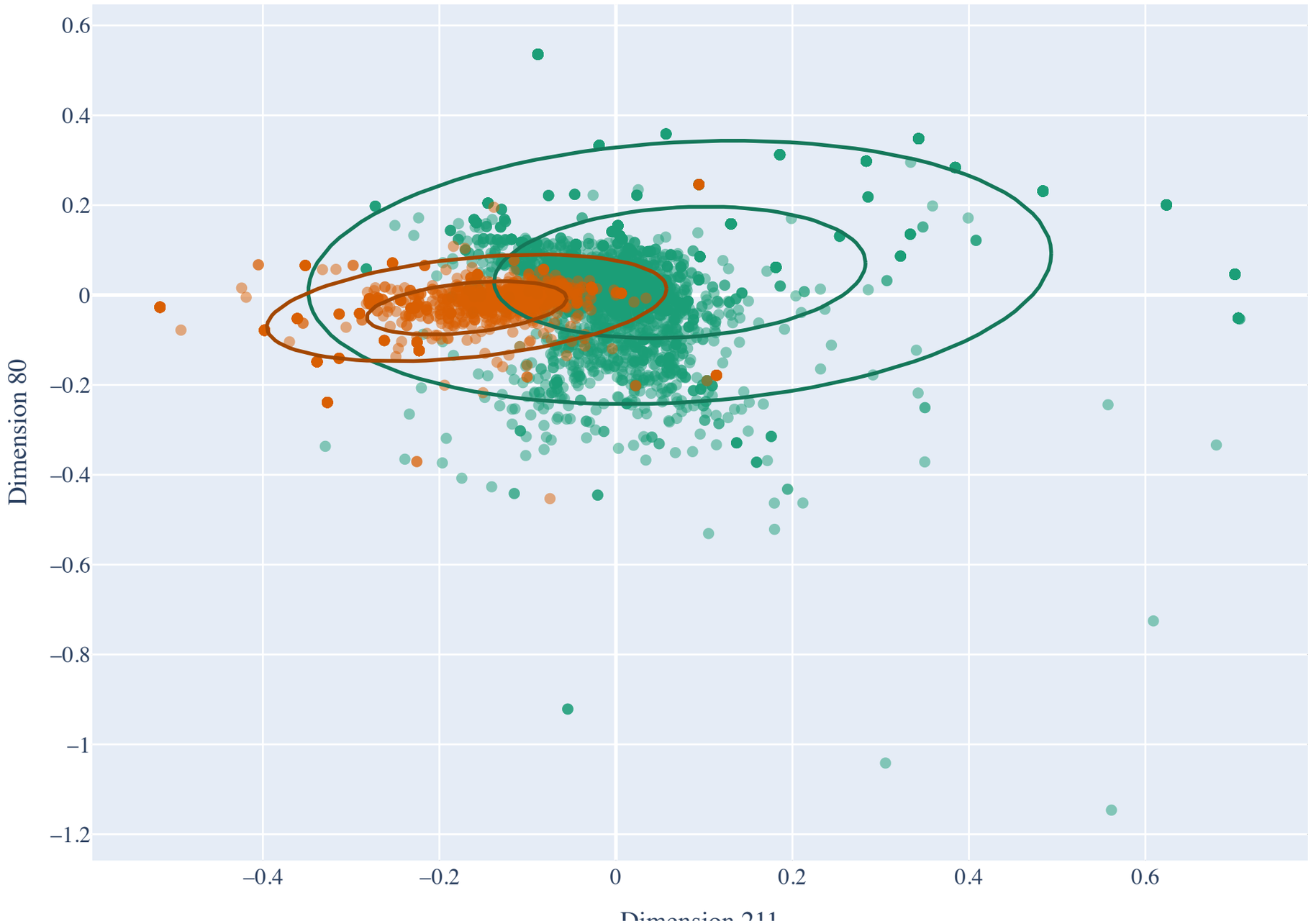


Figure 3: Comparison of per-attribute average lower-bound normalized mutual information (LBNMI) for fastText and BERT. Each bar is broken up into three components, which denote the LBNMI after selecting 2, 10 and 50 dimensions.

Limitations



The structure isn’t always well described by a Gaussian