Quantitative Text Analysis
Meeting 8: Unsupervised Machine Learning

Petro Tolochko

Overview

- From Supervised to Unsupervised Learning
- Measurement and the Quantity of Interest
- Clustering
- Topic Models (LDA and STM)
- Text Scaling (Wordfish)
- Validation and Interpretation

Machine Learning Paradigms

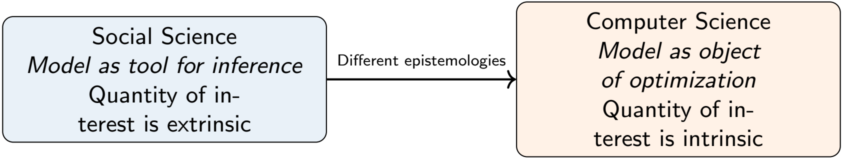
Supervised

- Outcome variable defined
- Focus: prediction and accuracy
- Objective function clear

Unsupervised

- No predefined labels
- Focus: structure discovery
- Objective = quantity of interest

Measurement and Paradigms



What is Measurement?

- Measurement = the process of assigning numbers (or symbols) to phenomena according to rules.
- Involves three linked steps:
  - **Conceptualization:** What do we want to capture?
  - **Operationalization:** How can we represent it empirically?
  - **Quantification:** How do we express it numerically?
- Measurement always implies a mapping between *theoretical constructs* and *observed data.*

Measurement in Social Science

- Measurement is often **theory-driven.**
- The goal is to capture latent constructs (e.g., trust, ideology, polarization).
- Accuracy means: correspondence between the *measure* and the *theoretical concept.*
- Model $\Rightarrow$ means to an end (a tool to understand the construct).

Focus: Construct validity and interpretability.

Measurement in Computer Science

- Measurement is typically **data-driven.**
- The model itself produces measurable quantities (loss, accuracy, error).
- Accuracy means: minimizing difference between $\hat{y}$ and y (or optimizing an objective function).
- Model $\Rightarrow$ end in itself (the metric is *intrinsic* to the model).

Focus: Optimization and predictive performance.

What Is an Objective Function?

- In machine learning, the **objective function** defines what the model tries to achieve.
- It translates a goal into something computable.
- Examples:
  - Linear regression: minimize squared error $(y - \hat{y})^2$
  - K-means: minimize within-cluster variance
  - PCA: minimize reconstruction error
- **Think of it as:** the model's "definition of success."

Objective Function vs. Quantity of Interest

Computer Science

- Objective function = quantity of interest
- "Success" is defined by minimizing loss
- Example: clustering minimizes distance

Social Science

- Quantity of interest is *theoretical*
- "Success" = capturing a latent construct
- Example: clustering may reflect norms, roles, or ideologies

In social science, we care about meaning, not just optimization.

Examples of Objective Functions

Model	**Objective Function**	**Goal**
Linear Regression	$\min(y - \hat{y})^2$	Fit predictions
K-Means	$\min \sum_i \|\|x_i - \mu_{c_i}\|\|^2$	Compact clusters
PCA	$\min \|\|X - X_{approx}\|\|^2$	Reduce dimensions
LDA	$\max P(w, z\|\alpha, \beta)$	Infer latent topics
Wordfish	$\max L(\alpha, \psi, \beta, \omega)$	Position texts

Key insight: each method measures something *different* because its objective differs.

Why the Objective Function Matters

- Defines what the model “cares about”.
- Different objectives $\Rightarrow$ different structures discovered.
- If the objective does not align with theory, results can mislead.

Good measurement = alignment between objective and theory.

Two Ways of Defining the Objective

Computer Science
Objective defines the question

Optimization ↔ Meaning

$\longrightarrow$

Social Science
Question defines the objective

What Does $\hat{\theta}$ Mean?

- Every model estimates parameters $\hat{\theta}$: coefficients, embeddings, topic proportions, etc.
- But the role of $\hat{\theta}$ differs across disciplines:
  - In **computer science**: $\hat{\theta}$ is a *means* — it helps minimize a loss function.
  - In **social science**: $\hat{\theta}$ is the *quantity of interest* — what we want to understand and interpret.

$\hat{\theta}$ in Two Paradigms

Computer Science

$$\min_{\theta} L(\theta, \hat{\theta}) = (\hat{y} - y)^2$$

- Goal: make predictions accurate.
- $\hat{\theta}$ has no meaning beyond performance.
- Once optimized, the parameters can be ignored.

Social Science

$\hat{\theta}$ is evidence about the world.

- Goal: explain.
- $\hat{\theta}$ approximates the theoretical construct θ.
- $\hat{\theta} \approx \theta$ is the research objective.

Two Roles of $\hat{\theta}$

Computer Science
$\hat{\theta}$ helps minimize objective function $L(\theta, \hat{\theta})$

different epistemic roles $\longrightarrow$

Social Science
$\hat{\theta}$ **is** the quantity of interest to be interpreted as θ

Same symbol, opposite direction of reasoning.

Two Views on Learning and Measurement

Two Views on Learning and Measurement

$$L(\theta, \hat{\theta}) = (\hat{\theta} - \theta)^2$$

Two Views on Learning and Measurement

$$L(\theta, \hat{\theta}) = (\hat{\theta} - \theta)^2$$

Means to an end
for social science

Main quantity of interest
for computer science

Two Views on Learning and Measurement

$$L(\theta, \hat{\theta}) = (\hat{\theta} - \theta)^2$$

Means to an end
for social science

Main quantity of interest
for computer science

$$\hat{\theta} \approx \theta$$

What social science wants

Focus on Discovery

Unsupervised methods shift our goal

From: Estimating $\hat{y}$ as close to y as possible

To: Discovering latent structure, clusters, or dimensions

Prediction vs Inference

- **Prediction:** How well does the model reproduce unseen data?
- **Inference:** What do parameters tell us about the world?
- Large parameter spaces $\Rightarrow$ poor inference, good prediction

When There Is No True θ

$$L(X, \hat{S})$$

When There Is No True θ

$$L(X, \hat{S})$$

The model no longer minimizes the distance between
a true parameter θ and its estimate $\hat{\theta}$,

but instead searches for an internal structure $\hat{S}$
that best organizes the data X.

When There Is No True θ

$$L(X, \hat{S})$$

The model no longer minimizes the distance between
a true parameter θ and its estimate $\hat{\theta}$,

but instead searches for an internal structure $\hat{S}$
that best organizes the data X.

K-Means:	minimize within-cluster variance
PCA:	minimize reconstruction error
LDA:	maximize topic coherence (fit to data)

When There Is No True θ

$$L(X, \hat{S})$$

The model no longer minimizes the distance between
a true parameter θ and its estimate $\hat{\theta}$,

but instead searches for an internal structure $\hat{S}$
that best organizes the data X.

K-Means:	minimize within-cluster variance
PCA:	minimize reconstruction error
LDA:	maximize topic coherence (fit to data)

In unsupervised learning, the objective function itself becomes the measure.

Problems

- **Translation of social science concepts** Machine learning measures patterns, not constructs.

Problems

- **Translation of social science concepts** Machine learning measures patterns, not constructs.
- **Connecting methods to theory** Models optimize statistical objectives, not theoretical meanings.

Problems

- **Translation of social science concepts** Machine learning measures patterns, not constructs.
- **Connecting methods to theory** Models optimize statistical objectives, not theoretical meanings.
- **Difficult to understand what is being measured** Without a clear mapping between theory and model, $\hat{S}$ may have no interpretable referent.

Unsupervised Learning Example

- Clustering algorithms are powerful exploratory tools.

Unsupervised Learning Example

- Clustering algorithms are powerful exploratory tools.
- Yet they don't fit well with the *standard* social science paradigm:
    - no ground truth
    - no predefined concept to validate against

Unsupervised Learning Example

- Clustering algorithms are powerful exploratory tools.
- Yet they don't fit well with the *standard* social science paradigm:
  - no ground truth
  - no predefined concept to validate against
- External validation is needed, but there is no single "best" validation criterion.

Unsupervised Learning Example

- Clustering algorithms are powerful exploratory tools.
- Yet they don't fit well with the *standard* social science paradigm:
  - no ground truth
  - no predefined concept to validate against
- External validation is needed, but there is no single "best" validation criterion.
- "Validation" based on theory or expectation can introduce confirmation bias.

The Paradigm

- In traditional modeling, we assume data are generated by some process $f(\theta)$.

The Paradigm

- In traditional modeling, we assume data are generated by some process $f(\theta)$.
- The goal is to **approximate the data-generating process.**

The Paradigm

- In traditional modeling, we assume data are generated by some process $f(\theta)$.
- The goal is to **approximate the data-generating process.**
- *Assumption:* there exists one (and only one) "true" data-generating process.

The Paradigm

- In traditional modeling, we assume data are generated by some process $f(\theta)$.
- The goal is to **approximate the data-generating process.**
- *Assumption:* there exists one (and only one) "true" data-generating process.
- This assumption implies that "truth" exists and can be recovered.

Sticking to the Paradigm

- The “normal” paradigm works only if we assume a single **correct** classification or data-generating process.

Sticking to the Paradigm

- The "normal" paradigm works only if we assume a single **correct** classification or data-generating process.
- Unsupervised methods violate this assumption: they discover **multiple** plausible structures.

Sticking to the Paradigm

- The "normal" paradigm works only if we assume a single **correct** classification or data-generating process.
- Unsupervised methods violate this assumption: they discover **multiple** plausible structures.
- Therefore, unsupervised models are **meaningless** if interpreted under the "true model" assumption.

Sticking to the Paradigm

- The “normal” paradigm works only if we assume a single **correct** classification or data-generating process.
- Unsupervised methods violate this assumption: they discover **multiple** plausible structures.
- Therefore, unsupervised models are **meaningless** if interpreted under the “true model” assumption.
- We need a new epistemology: *discovery-oriented*, not truth-oriented.

From Truth to Discovery

Truth-oriented: $\theta \rightarrow \hat{\theta}$

Model approximates a single, true process.

From Truth to Discovery

Truth-oriented: $\theta \rightarrow \hat{\theta}$

Model approximates a single, true process.

Discovery-oriented: $X \rightarrow \hat{S}$

Model identifies patterns or latent structures in the data.

From Truth to Discovery

Truth-oriented: $\theta \rightarrow \hat{\theta}$

Model approximates a single, true process.

Discovery-oriented: $X \rightarrow \hat{S}$

Model identifies patterns or latent structures in the data.

In unsupervised learning, meaning is not recovered, it is constructed.

Epistemology of Discovery

Measurement as Mapping

$$\theta \longrightarrow X$$

Model represents existing theoretical constructs.

Epistemology of Discovery

Measurement as Mapping

$$\theta \longrightarrow X$$

Model represents existing theoretical constructs.

Measurement as Modeling

$$X \longrightarrow \hat{S}$$

Model induces its own structure from data.

Epistemology of Discovery

Measurement as Mapping

$$\theta \longrightarrow X$$

Model represents existing theoretical constructs.

Measurement as Modeling

$$X \longrightarrow \hat{S}$$

Model induces its own structure from data.

Discovery = modeling as a form of measurement.

Questions?

Families of Unsupervised Methods

- **Clustering:**
  Find groups of similar items.
  Examples: K-Means, Hierarchical Clustering.

- **Topic Modeling:**
  Find latent themes in text.
  Examples: Latent Dirichlet Allocation (LDA), Structural Topic Model (STM).

- **Scaling:**
  Place items along latent dimensions.
  Example: Wordfish (Slapin & Proksch, 2008).

K-Means Clustering

- Partitions data into K non-overlapping clusters
- Simple(ish) algorithmic method

K-Means Clustering

$$C_1, C_2, \ldots, C_K$$

K-Means Clustering

$$C_1, C_2, \ldots, C_K$$

$$C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$$

K-Means Clustering

$$C_1, C_2, \ldots, C_K$$

$$C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$$

$$C_k \cap C_{k'} = \varnothing \quad \text{for all } k \neq k'$$

K-Means Clustering

$$C_1, C_2, \ldots, C_K$$

$$C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$$

$$C_k \cap C_{k'} = \varnothing \quad \text{for all } k \neq k'$$

Each observation belongs to exactly one cluster.
Clusters are non-overlapping and collectively exhaustive.

Objective Function

Goal: Find K clusters that minimize within-cluster variation.

Objective Function

Goal: Find K clusters that minimize within-cluster variation.

$$\min_{C_1,\dots,C_K} \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

Objective Function

Goal: Find K clusters that minimize within-cluster variation.

$$\min_{C_1,\dots,C_K} \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

where μ_k is the centroid (mean) of cluster C_k:

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

Objective Function

Goal: Find K clusters that minimize within-cluster variation.

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

where μ_k is the centroid (mean) of cluster C_k:

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

Intuition: assign points to clusters so that each cluster is as compact as possible.

K-Means Algorithm

Algorithm: Iterative refinement of cluster assignments.

K-Means Algorithm

Algorithm: Iterative refinement of cluster assignments.

1. Choose the number of clusters K.

K-Means Algorithm

Algorithm: Iterative refinement of cluster assignments.

1. Choose the number of clusters K.
2. Randomly assign each observation to one of the K clusters.

K-Means Algorithm

Algorithm: Iterative refinement of cluster assignments.

1. Choose the number of clusters K.
2. Randomly assign each observation to one of the K clusters.
3. Compute the centroid μ_k for each cluster:

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

K-Means Algorithm

Algorithm: Iterative refinement of cluster assignments.

1. Choose the number of clusters K.
2. Randomly assign each observation to one of the K clusters.
3. Compute the centroid μ_k for each cluster:

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

4. Reassign each observation to the cluster with the nearest centroid.

K-Means Algorithm

Algorithm: Iterative refinement of cluster assignments.

1. Choose the number of clusters K.
2. Randomly assign each observation to one of the K clusters.
3. Compute the centroid μ_k for each cluster:

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

4. Reassign each observation to the cluster with the nearest centroid.
5. Repeat steps 3–4 until assignments do not change.

K-Means Algorithm

Algorithm: Iterative refinement of cluster assignments.

1. Choose the number of clusters K.
2. Randomly assign each observation to one of the K clusters.
3. Compute the centroid μ_k for each cluster:

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

4. Reassign each observation to the cluster with the nearest centroid.
5. Repeat steps 3–4 until assignments do not change.

K-Means converges to a local optimum, not necessarily the global one.

Choosing the Number of Clusters

- **Elbow method:** plot WCSS vs. K
- **Silhouette score:** average distance to own cluster vs others
- **Theory-informed:** choose K based on expected structure

Cluster Algorithms Validation

- **Data assumptions:** check if clustering assumptions match data generation.

Cluster Algorithms Validation

- **Data assumptions:** check if clustering assumptions match data generation.
- **Internal validity:** best results for the data (e.g., cohesion, separation).

Cluster Algorithms Validation

- **Data assumptions:** check if clustering assumptions match data generation.
- **Internal validity:** best results for the data (e.g., cohesion, separation).
- **External validity:** alignment with pre-existing understanding or labels.

Cluster Algorithms Validation

- **Data assumptions:** check if clustering assumptions match data generation.
- **Internal validity:** best results for the data (e.g., cohesion, separation).
- **External validity:** alignment with pre-existing understanding or labels.
- **Cross-validity:** stability across similar datasets or time periods.

Cluster Algorithms Validation

- **Data assumptions:** check if clustering assumptions match data generation.
- **Internal validity:** best results for the data (e.g., cohesion, separation).
- **External validity:** alignment with pre-existing understanding or labels.
- **Cross-validity:** stability across similar datasets or time periods.
- **You are the validation method.**
  Interpretation and theory provide the ultimate validation.
- Validation moves from *data–fit* to *meaning–fit*.

Topic Models

Topic Modeling

- **Goal:** discover latent themes or "topics" in a collection of documents.

Topic Modeling

- **Goal:** discover latent themes or “topics” in a collection of documents.
- Topic modeling is a **family of models**, not a single algorithm.

Topic Modeling

- **Goal:** discover latent themes or "topics" in a collection of documents.
- Topic modeling is a **family of models**, not a single algorithm.
- **Latent Dirichlet Allocation (LDA)** is one of several topic models.

Topic Modeling

- **Goal:** discover latent themes or “topics” in a collection of documents.
- Topic modeling is a **family of models**, not a single algorithm.
- **Latent Dirichlet Allocation (LDA)** is one of several topic models.
- Other approaches include:
  - Latent Semantic Analysis (LSA)
  - Singular Value Decomposition (SVD)
  - Clustering-based topic discovery

Latent Dirichlet Allocation (LDA)

- A **Bayesian generative hierarchical model** for discovering topics in text.

Latent Dirichlet Allocation (LDA)

- A **Bayesian generative hierarchical model** for discovering topics in text.
- Originally introduced to model *population structure* in genetics (Pritchard, Stephens & Donnelly, 2000).

Latent Dirichlet Allocation (LDA)

- A **Bayesian generative hierarchical model** for discovering topics in text.
- Originally introduced to model *population structure* in genetics (Pritchard, Stephens & Donnelly, 2000).
- Adapted to text analysis as a probabilistic topic model (Blei, Ng & Jordan, 2003).

Latent Dirichlet Allocation (LDA)

- A **Bayesian generative hierarchical model** for discovering topics in text.
- Originally introduced to model *population structure* in genetics (Pritchard, Stephens & Donnelly, 2000).
- Adapted to text analysis as a probabilistic topic model (Blei, Ng & Jordan, 2003).
- Each document is modeled as a *mixture of topics*, and each topic as a *distribution over words*.

Latent Dirichlet Allocation (LDA)

- Goal: discover latent topics that generate observed words.
- Assumption: documents are mixtures of topics; topics are distributions of words.

Mixture Models

- **Idea:** observations come from a combination of several underlying distributions.

Mixture Models

- **Idea:** observations come from a combination of several underlying distributions.
- Each distribution represents a **latent group or component**.

Mixture Models

- **Idea:** observations come from a combination of several underlying distributions.
- Each distribution represents a **latent group or component**.
- The probability of each component is weighted by a **mixture proportion**.

Mixture Models

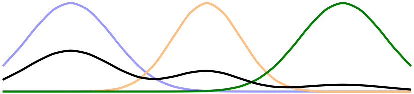
- **Idea:** observations come from a combination of several underlying distributions.
- Each distribution represents a **latent group or component**.
- The probability of each component is weighted by a **mixture proportion**.
- Unlike K-means, mixture models use *soft assignment:*
  each observation belongs to components with certain probabilities.

Mixture Models

- **Idea:** observations come from a combination of several underlying distributions.
- Each distribution represents a **latent group or component**.
- The probability of each component is weighted by a **mixture proportion**.
- Unlike K-means, mixture models use *soft assignment:* each observation belongs to components with certain probabilities.

Example: a document can be 70% about politics, 20% about health, 10% about sports.

Mixture Model



Observed data = weighted combination of latent distributions

Hierarchical Models

$$y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon$$

Hierarchical Models

$$y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon$$

In a standard regression model, coefficients β are fixed unknowns.
We estimate them directly from the data.

Hierarchical Models

$$y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon$$

In a standard regression model, coefficients β are fixed unknowns.
We estimate them directly from the data.

In hierarchical (Bayesian) models, coefficients are random variables with their own probability distributions.

Hierarchical Models

$$y \sim Normal(\mu, \sigma)$$
$$\mu = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$
$$\beta \sim Normal(0, 5)$$
$$\sigma \sim Exponential(1)$$

Hierarchical Models

$$
\begin{aligned}
y &\sim Normal(\mu, \sigma) \\
\mu &= \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k \\
\beta &\sim Normal(0, 5) \\
\sigma &\sim Exponential(1)
\end{aligned}
$$

Likelihood: $y \sim Normal(\mu, \sigma)$: how data are generated.
Priors: β, σ have their own distributions.
The model is hierarchical because parameters depend on other parameters.

Hierarchical Models: Hyperparameters

$$\beta \sim Normal(0, 5)$$
$$\sigma \sim Exponential(1)$$

Hierarchical Models: Hyperparameters

$$\beta \sim Normal(0, 5)$$
$$\sigma \sim Exponential(1)$$

The numbers $(0, 5)$ and (1) are **hyperparameters**:
they define the prior distributions for model parameters.

Hierarchical Models: Hyperparameters

$$\beta \sim Normal(0, 5)$$
$$\sigma \sim Exponential(1)$$

The numbers $(0, 5)$ and (1) are **hyperparameters**:
they define the prior distributions for model parameters.

In LDA, the Dirichlet priors α and β play this same role.
They control how concentrated or diffuse the topic and document distributions are.

LDA Formalized

Generative process:

1. For each topic $k \in \{1, \ldots, K\}$:

$$\theta_k \sim \text{Dirichlet}(\alpha)$$

(distribution of words in topic k)

LDA Formalized

Generative process:

1. For each topic $k \in \{1, \ldots, K\}$:

$$\theta_k \sim \text{Dirichlet}(\alpha)$$

(distribution of words in topic k)

2. For each document $d \in \{1, \ldots, D\}$:

$$\eta_d \sim \text{Dirichlet}(\beta)$$

(distribution of topics in document d)

LDA Formalized

Generative process:

1. For each topic $k \in \{1, \ldots, K\}$:

$$\theta_k \sim \text{Dirichlet}(\alpha)$$

   (distribution of words in topic k)

2. For each document $d \in \{1, \ldots, D\}$:

$$\eta_d \sim \text{Dirichlet}(\beta)$$

   (distribution of topics in document d)

3. For each word w in document d:

$$\begin{aligned} z_{d,w} &\sim \text{Multinomial}(\eta_d) \quad &&\text{(topic assignment)} \\ x_{d,w} &\sim \text{Multinomial}(\theta_{z_{d,w}}) \quad &&\text{(observed word)} \end{aligned}$$

LDA Formalized

Generative process:

1. For each topic $k \in \{1, \ldots, K\}$:

$$\theta_k \sim \text{Dirichlet}(\alpha)$$

   (distribution of words in topic k)

2. For each document $d \in \{1, \ldots, D\}$:

$$\eta_d \sim \text{Dirichlet}(\beta)$$

   (distribution of topics in document d)

3. For each word w in document d:

$$\begin{aligned} z_{d,w} &\sim \text{Multinomial}(\eta_d) && \text{(topic assignment)} \\ x_{d,w} &\sim \text{Multinomial}(\theta_{z_{d,w}}) && \text{(observed word)} \end{aligned}$$

Goal: infer the latent variables θ, η, and z given observed words x.

LDA Formalized

Generative process:

1. For each topic $k \in \{1, \ldots, K\}$:

$$\theta_k \sim \text{Dirichlet}(\alpha)$$

(distribution of words in topic k)

2. For each document $d \in \{1, \ldots, D\}$:

$$\eta_d \sim \text{Dirichlet}(\beta)$$

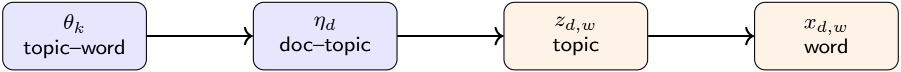
(distribution of topics in document d)

3. For each word w in document d:

$$\begin{aligned} z_{d,w} &\sim \text{Multinomial}(\eta_d) && \text{(topic assignment)} \\ x_{d,w} &\sim \text{Multinomial}(\theta_{z_{d,w}}) && \text{(observed word)} \end{aligned}$$

Goal: infer the latent variables θ, η, and z given observed words x.
Formally: $P(\theta, \eta, z \mid x, \alpha, \beta)$

LDA Generative Process



Model first draws topic–word distributions, then document–topic mixtures, assigns topics to words, and generates observed words.

LDA in Words

1. For each topic k, draw word distribution $\theta_k \sim Dir(\alpha)$
2. For each document d, draw topic mixture $\eta_d \sim Dir(\beta)$
3. For each word:
   - Choose topic $z_{d,w} \sim Mult(\eta_d)$
   - Choose word $x_{d,w} \sim Mult(\theta_{z_{d,w}})$

Dirichlet Distribution

- A **distribution over distributions.**

Dirichlet Distribution

- A **distribution over distributions.**
- It describes random vectors that sum to 1:

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K), \quad \text{where } \sum_{k=1}^{K} \theta_k = 1.$$

Dirichlet Distribution

- A **distribution over distributions.**
- It describes random vectors that sum to 1:

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_K), \quad \text{where } \sum_{k=1}^{K} \theta_k = 1.$$

- Parameterized by $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$, which control how concentrated or spread out the probabilities are.

Dirichlet Distribution

- A **distribution over distributions.**
- It describes random vectors that sum to 1:

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K), \quad \text{where } \sum_{k=1}^{K} \theta_k = 1.$$

- Parameterized by $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$, which control how concentrated or spread out the probabilities are.
- **High** α**:** all θ_k similar (uniform).
  Low α**:** few θ_k dominate (sparse distribution).

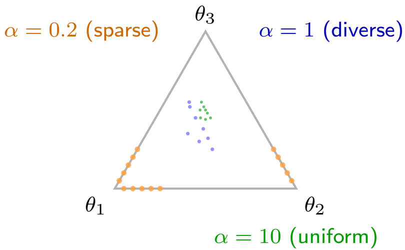
Dirichlet Distribution

- A **distribution over distributions.**
- It describes random vectors that sum to 1:

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_K), \quad \text{where } \sum_{k=1}^{K} \theta_k = 1.$$

- Parameterized by $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$, which control how concentrated or spread out the probabilities are.
- **High α:** all θ_k similar (uniform).
  Low α: few θ_k dominate (sparse distribution).
- In LDA, Dirichlet priors control how concentrated topics or words are.

Visualizing the Dirichlet Distribution



Smaller $\alpha \rightarrow$ *more peaked; larger* $\alpha \rightarrow$ *more uniform.*
Each dot is a draw $(\theta_1, \theta_2, \theta_3)$ lying on the simplex.

Dirichlet in LDA

Variable	Drawn from	Meaning
η_d	Dirichlet(β)	Topic distribution for document d
θ_k	Dirichlet(α)	Word distribution for topic k

Dirichlet priors define the diversity or sparsity of topics and words.

From LDA to STM

- **LDA:** all documents share the same prior over topics.

From LDA to STM

- **LDA:** all documents share the same prior over topics.
- **STM (Structural Topic Model):** extends LDA by allowing topic prevalence and content to vary with **document covariates**.

From LDA to STM

- **LDA:** all documents share the same prior over topics.
- **STM (Structural Topic Model):** extends LDA by allowing topic prevalence and content to vary with **document covariates**.
- Covariates can affect:
  - **Topic prevalence —** how much each topic appears in a document.
  - **Topic content —** which words are used to discuss that topic.

From LDA to STM

- **LDA:** all documents share the same prior over topics.
- **STM (Structural Topic Model):** extends LDA by allowing topic prevalence and content to vary with **document covariates**.
- Covariates can affect:
    - **Topic prevalence —** how much each topic appears in a document.
    - **Topic content —** which words are used to discuss that topic.
- Uses a **Logistic-Normal** prior instead of a Dirichlet:

$$\eta_d \sim \text{LogisticNormal}(\mu_d, \Sigma), \quad \mu_d = X_d\gamma$$

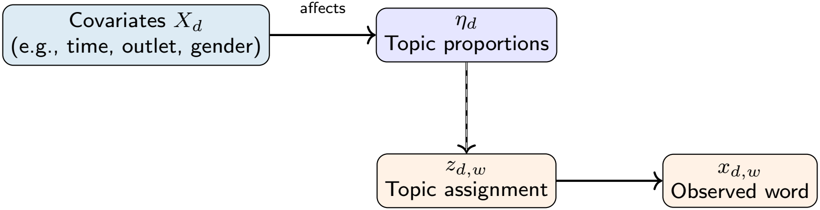
From LDA to STM

- **LDA:** all documents share the same prior over topics.
- **STM (Structural Topic Model):** extends LDA by allowing topic prevalence and content to vary with **document covariates**.
- Covariates can affect:
  - **Topic prevalence —** how much each topic appears in a document.
  - **Topic content —** which words are used to discuss that topic.
- Uses a **Logistic-Normal** prior instead of a Dirichlet:

$$\eta_d \sim \text{LogisticNormal}(\mu_d, \Sigma), \quad \mu_d = X_d\gamma$$

- Enables inference: how topics vary by time, source, gender, ideology, etc.

How STM Extends LDA



STM extends LDA by adding document-level structure:
*Covariates influence which topics appear (***prevalence***) and how topics are expressed (***content***).*

Why Validate Unsupervised Models?

- No ground truth $\Rightarrow$ model can always find patterns.

Why Validate Unsupervised Models?

- No ground truth $\Rightarrow$ model can always find patterns.
- Validation checks whether discovered structure reflects:
  - the *data* (internal coherence),
  - the *world* (external correspondence),
  - or the *theory* (interpretive meaning).

Why Validate Unsupervised Models?

- No ground truth $\Rightarrow$ model can always find patterns.
- Validation checks whether discovered structure reflects:
  - the *data* (internal coherence),
  - the *world* (external correspondence),
  - or the *theory* (interpretive meaning).
- Without validation, unsupervised results are **artifacts of the algorithm**.

Validation of Unsupervised Models

- **Data assumptions:** check that model assumptions match the data structure.
  Example: K-means assumes spherical clusters, LDA assumes word exchangeability.

Validation of Unsupervised Models

- **Data assumptions:** check that model assumptions match the data structure.
  Example: K-means assumes spherical clusters, LDA assumes word exchangeability.
- **Internal validity:** evaluate fit within the data (e.g., cohesion, silhouette, WCSS, topic coherence).

Validation of Unsupervised Models

- **Data assumptions:** check that model assumptions match the data structure.
  Example: K-means assumes spherical clusters, LDA assumes word exchangeability.
- **Internal validity:** evaluate fit within the data (e.g., cohesion, silhouette, WCSS, topic coherence).
- **External validity:** compare with known or manually coded classifications, human judgments, or metadata.

Validation of Unsupervised Models

- **Data assumptions:** check that model assumptions match the data structure.
  Example: K-means assumes spherical clusters, LDA assumes word exchangeability.
- **Internal validity:** evaluate fit within the data (e.g., cohesion, silhouette, WCSS, topic coherence).
- **External validity:** compare with known or manually coded classifications, human judgments, or metadata.
- **Cross-validity:** check stability across samples, time periods, or model runs.

Validation of Unsupervised Models

- **Data assumptions:** check that model assumptions match the data structure.
  Example: K-means assumes spherical clusters, LDA assumes word exchangeability.
- **Internal validity:** evaluate fit within the data (e.g., cohesion, silhouette, WCSS, topic coherence).
- **External validity:** compare with known or manually coded classifications, human judgments, or metadata.
- **Cross-validity:** check stability across samples, time periods, or model runs.
- **Interpretive validity:** the researcher's qualitative assessment of meaning — *You are the validation method.*

Validating Topic Models

- **Semantic validity:** do top words within each topic form a coherent theme?
  (often checked with human evaluation)

Validating Topic Models

- **Semantic validity:** do top words within each topic form a coherent theme?
  (often checked with human evaluation)
- **Convergent validity:** do topic proportions correlate with external variables
  or manual codings for the same documents?
  (e.g., Guo et al., 2016)

Validating Topic Models

- **Semantic validity:** do top words within each topic form a coherent theme?
  (often checked with human evaluation)
- **Convergent validity:** do topic proportions correlate with external variables
  or manual codings for the same documents?
  (e.g., Guo et al., 2016)
- **Quantitative metrics:**
  - *Topic coherence* (e.g., NPMI, UMass, UCI)
  - *Held-out likelihood* or *perplexity*

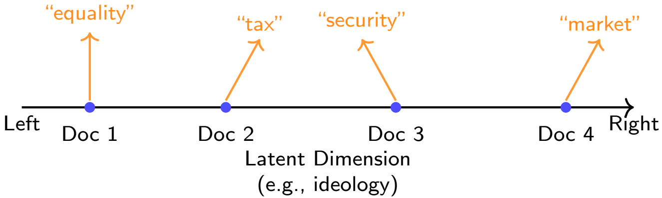
Validating Topic Models

- **Semantic validity:** do top words within each topic form a coherent theme?
  (often checked with human evaluation)
- **Convergent validity:** do topic proportions correlate with external variables
  or manual codings for the same documents?
  (e.g., Guo et al., 2016)
- **Quantitative metrics:**
  - *Topic coherence* (e.g., NPMI, UMass, UCI)
  - *Held-out likelihood* or *perplexity*
- **Human-in-the-loop validation:**
  Experts or crowdsourced coders rate interpretability and label consistency.

Text Scaling Methods

- Place documents on latent dimensions (e.g., ideology).
- Use discriminating word frequencies as anchors.

Text Scaling



Documents are positioned along a latent dimension according to word usage.
Discriminating words push documents toward one end or the other.

Assumptions of Text Scaling Methods

- **Unidimensional structure:** texts vary along one latent continuum (e.g., ideology).

Assumptions of Text Scaling Methods

- **Unidimensional structure:** texts vary along one latent continuum (e.g., ideology).
- **Conditional independence:** words are independent given a document's latent position.

Assumptions of Text Scaling Methods

- **Unidimensional structure:** texts vary along one latent continuum (e.g., ideology).
- **Conditional independence:** words are independent given a document's latent position.
- **Stable word meanings:** each word has the same directional association across texts.

Assumptions of Text Scaling Methods

- **Unidimensional structure:** texts vary along one latent continuum (e.g., ideology).
- **Conditional independence:** words are independent given a document's latent position.
- **Stable word meanings:** each word has the same directional association across texts.
- **Comparability:** all documents are drawn from the same domain or context.

Assumptions of Text Scaling Methods

- **Unidimensional structure:** texts vary along one latent continuum (e.g., ideology).
- **Conditional independence:** words are independent given a document's latent position.
- **Stable word meanings:** each word has the same directional association across texts.
- **Comparability:** all documents are drawn from the same domain or context.
- **Signal vs. topic:** variation in word use reflects position, not topic differences.

Assumptions of Text Scaling Methods

- **Unidimensional structure:** texts vary along one latent continuum (e.g., ideology).
- **Conditional independence:** words are independent given a document's latent position.
- **Stable word meanings:** each word has the same directional association across texts.
- **Comparability:** all documents are drawn from the same domain or context.
- **Signal vs. topic:** variation in word use reflects position, not topic differences.
- **Scale invariance:** only relative positions matter, the scale's origin is arbitrary.

Wordfish as a Hierarchical Model (Slapin & Proksch, 2008)

Likelihood:

$$y_{ik} \sim \mathsf{Poisson}(\lambda_{ik})$$

$$\log \lambda_{ik} = \alpha_i + \psi_k + \beta_k \, \omega_i$$

Wordfish as a Hierarchical Model (Slapin & Proksch, 2008)

Likelihood:

$$y_{ik} \sim \text{Poisson}(\lambda_{ik})$$

$$\log \lambda_{ik} = \alpha_i + \psi_k + \beta_k \, \omega_i$$

Parameters:

- α_i — document-specific intercept (captures length / verbosity)
- ψ_k — word-specific intercept (baseline frequency)
- β_k — word-specific discrimination (strength of association with dimension)
- ω_i — document position on the latent scale

Wordfish as a Hierarchical Model (Slapin & Proksch, 2008)

Likelihood:

$$y_{ik} \sim \text{Poisson}(\lambda_{ik})$$

$$\log \lambda_{ik} = \alpha_i + \psi_k + \beta_k \, \omega_i$$

Parameters:

- α_i — document-specific intercept (captures length / verbosity)
- ψ_k — word-specific intercept (baseline frequency)
- β_k — word-specific discrimination (strength of association with dimension)
- ω_i — document position on the latent scale

Hierarchical structure:

$$\alpha_i \sim \text{Normal}(0, \sigma_\alpha)$$
$$\psi_k \sim \text{Normal}(0, \sigma_\psi)$$
$$\beta_k \sim \text{Normal}(0, \sigma_\beta)$$
$$\omega_i \sim \text{Normal}(0, 1)$$

Wordfish as a Hierarchical Model (Slapin & Proksch, 2008)

Likelihood:

$$y_{ik} \sim \text{Poisson}(\lambda_{ik})$$

$$\log \lambda_{ik} = \alpha_i + \psi_k + \beta_k \, \omega_i$$

Parameters:

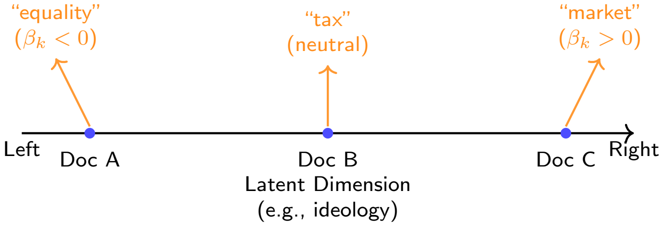
- α_i — document-specific intercept (captures length / verbosity)
- ψ_k — word-specific intercept (baseline frequency)
- β_k — word-specific discrimination (strength of association with dimension)
- ω_i — document position on the latent scale

Hierarchical structure:

$$\alpha_i \sim \text{Normal}(0, \sigma_\alpha)$$
$$\psi_k \sim \text{Normal}(0, \sigma_\psi)$$
$$\beta_k \sim \text{Normal}(0, \sigma_\beta)$$
$$\omega_i \sim \text{Normal}(0, 1)$$

Wordfish estimates document positions ω_i and word discriminations β_k jointly.

How Wordfish Links Words and Documents



Words with positive β_k pull documents to the right; words with negative β_k pull them to the left.
Estimated ω_i summarize these directional tendencies.

Summary: Comparing Unsupervised Methods

	Goal	Output	Example
Clustering	Grouping	Cluster labels	K-means
Topic Modeling	Thematic discovery	Topic-word probs	LDA / STM
Scaling	Latent dimension	Position scores	Wordfish

What Unsupervised Learning Does (and Doesn't Do)

- **Discovers structure**, it does not define it.
- **Measures relationships**, not "truth".
- Depends on **assumptions about meaning, similarity, and latent space.**
- **Useful for exploration and theory building**, not just prediction.

Unsupervised learning helps us see structure we didn't know was there, but we still have to interpret what that structure means.

Each method is a different kind of measurement, and every measurement is a theoretical exercise.

Questions?