Hierarchical Method of Moments

Matteo Ruffini; Guillaume Rabusseau; Borja Balle

-Presented By: Shishir Sharma

Learning Latent Variable Models

- Unsupervised learning of latent variable models is a fundamental machine learning problem
- Such algorithms allow for learning a variety of latent variable models such as Topic models, Hidden Markov Models, having varied practical applications

Outline

- What is the misspecified setting?
- Why work with Method of Moments ?
- Why existing Method of Moments algorithms fail?
- SIDIWO
- Behavior in the misspecified setting
- Hierarchical Topic Modeling application

Misspecified Setting

- Refers to the setting where data generating model and learnt model differ in class
- Examples include learning a model with few number of latent variables
- Another example arises from noisy moment estimates

Model Misspecification

- Important, thus, to design algorithms with meaningful estimation in case $l < k$ latent variables
- This setting is referred to as the misspecified setting
- The setting where $l = k$ is referred to as the realizable setting

Expectation Maximization

- Expectation Maximization (EM) is preferred despite having theoretical limitations due to robustness of the maximum-likelihood principle to model misspecification
- Usually requires tuning a single parameter i.e. the dimension of the latent variables
- Yields models which are easy to interpret and are useful for data visualization in the misspecified setting

Method of Moment

- Various Method of moments (MoM) algorithms for learning latent variable models exist
- Involve solving non-linear system of equations via tensor factorization algorithms
- Suffer from lack of robustness to model misspecification

Why work with MoM algorithms then?

- Provide a stronger theoretical foundation for learning latent variable models
- Converges to true parameters, as the data increases in the realizable setting
- Require only a single pass over the training data
- Highly parallelizable
- Always terminate in polynomial time

Single Topic Model

- We consider a single topic model with *k* topics over a vocabulary with *d* words.
- Model defines a generative process for text documents
- Model defines a discrete distribution over the topics ($Y \in [k]$) as well as the words ($X_t \in [d], 1 \leq t \leq T$) from a given topic.

$$\mathbb{P}[Y = i] = w_i$$
$$\mathbb{P}[X_t = j | Y = i] = \mu_{i,j}$$

Single Topic Model

- The moment equations for such a model are:

$$M_1 = \mathbb{E}[X_t] = \sum_i \omega_i \mu_i = M\omega$$

$$M_2 = \mathbb{E}[X_s \otimes X_t] = \sum \omega_i \mu_i \otimes \mu_i \in \mathbb{R}^{d \times d}$$

$$M_3 = \mathbb{E}[X_r \otimes X_s \otimes X_t] = \sum_i \omega_i \mu_i \otimes \mu_i \otimes \mu_i \in \mathbb{R}^{d \times d \times d}$$

where $\bigotimes$ denotes the tensor (Kronecker) product between vectors, and

$$M = [\mu_1 \cdots \mu_k] \in \mathbb{R}^{d \times k}$$

Method of Moment Algorithm

Applying MoM for learning single topic models involves:

- Computing empirical estimates $\hat{M}_1, \hat{M}_2, \hat{M}_3$ of the moments using a collection of n documents
- Factoring the empirical moments using Matrix and Tensor decomposition algorithms
- Solving the system of non-linear equation for model parameters

Whitening Transformation

- For random variable X ~ N(μ, Σ), whitening transformation results in another random vector whose covariance is the identity matrix
- The components of the new random variable are uncorrelated
- For whitening the covariance matrix Σ, we perform the linear transformation using matrix W:

$$W^T \Sigma W = I$$

Example MoM algorithm

- Matrix form for M_2:

$$M_2 = \sum_{1=i}^{k} w_i \mu_i \otimes \mu_i = M \Omega M^T = U S U^T$$

where, $\Omega = \mathrm{diag}(w)$

- Whitening Matrix $E = U S^{1/2}$ is used
- There exists a unique orthonormal matrix *O*, such that

$$M \Omega^{1/2} = E O$$

Example MoM algorithm

- Denoting the r^{th} row of M as m_r, the r^{th} slice of M_3 across its second mode can then be written as :

$$M_{3,r} = M\Omega^{1/2} diag(m_r) \Omega^{1/2} M^\top$$

- Substituting $M\Omega^{1/2} = EO$

$$M_{3,r} = M\Omega^{1/2} diag(m_r) \Omega^{1/2} M^\top = EO diag(m_r) O^\top E^\top$$

Example MoM algorithm

- Whitened slices of M_3 are simultaneously diagonalized by its pseudoinverse

$$H_r = E^{\dagger} M_{3,r} E^{\dagger\top} = O diag(m_r) O^{\top}$$

- Thus, the problem can be reduced to searching for the common diagonalizer *O* of the whitened slices of M_3

Misspecified Setting

- In the misspecified setting, $E_l^{\dagger}$ derived from the low-rank SVD truncated at rank *l*, will be used instead

$$M_2 \approx U_l S_l U_l^{\top} = E_l E_l^{\top}$$

- The issue here is that the whitened slices that we get i.e. $H_{l,r}$ may not be jointly diagonizable

$$H_{l,r} = E_l^{\dagger} M_{3,r} E_l^{\dagger\top}$$

Simultaneous Diagonalization Based on Whitening and Optimization

- The idea is to cast the problem as on optimization problem constrained to produce meaningful results incase $l < k$ and optimal results otherwise

Lemma: *Let $M_{3,r}$ be the r-th slice across the second mode of the tensor M_3 from (2) with parameters (M, ω). Suppose $rank(M) = k$ and let $\Omega = \mathrm{diag}(\omega)$. Then the matrix $(M\Omega^{1/2})^\dagger$ is the unique optimum (up to column rescaling) of the optimization problem*

$$\min_{D \in \mathcal{D}_k} \sum_{i \neq j} \left(\sum_{r=1}^{d} (D M_{3,r} D^\top)_{i,j}^2 \right)^{1/2},$$

where $\mathcal{D}_k = \{D : D = (EO_k)^\dagger$ for some O_k s.t. $O_k O_k^\top = \mathbb{I}_k\}$ and E is the whitening matrix

Proof of Lemma

$$M_{3,r} = M\,\Omega^{1/2} diag(m_r)\,\Omega^{1/2}\,M^\top = EOdiag(m_r)\,(EO)^\top.$$

Take now a generic matrix $D \in \mathcal{D}_k$; it can be written as

$$D = O_k^\top E^\dagger$$

for a certain orthonormal matrix O_k. So we have

$$DM_{3,r}D^\top = O_k^\top E^\dagger EOdiag(m_r)\,(EO)^\top (O_k^\top E^\dagger)^\top = O_k^\top Odiag(m_r)\,OO_k$$

This matrix is diagonal if and only if $O = O_k$, so the problem

$$\min_{D\in\mathcal{D}_k} \sum_{i\neq j} \Big(\sum_{r=1}^{d} (DM_{3,r}D^\top)_{i,j}^2\Big)^{1/2}$$

is optimized by $D = (M\Omega^{1/2})^\dagger = (EO)^\dagger$, which is the unique (up to a columns rescaling) feasible optimum.

Constraining the optimization solution

- The constraint $D = (EO_k)^\dagger$ is the trivial zero matrix solution.
- Also guarantees the feasible solutions lay in the column space of M as:

$$D^\dagger = M\Omega^{1/2} = EO = US^{1/2}O$$

- The columns of *U* are the left singular vectors of $M\Omega^{1/2}$

SIDIWO

Algorithm 1 SIDIWO: Simultaneous Diagonalization based on Whitening and Optimization

Require: M_1, M_2, M_3, the number of latent states l

1: Compute a SVD of M_2 truncated at the l-th singular vector: $M_2 \approx U_l S_l U_l^\top$.

2: Define the matrix $E_l = U_l S_l^{1/2} \in \mathbb{R}^{d \times l}$.

3: Find the matrix $D \in \mathcal{D}_l$ optimizing Problem (5).

4: Find $(\tilde{M}, \tilde{\omega})$ solving $\begin{cases} \tilde{M}\tilde{\Omega}^{1/2} = D^\dagger \\ \tilde{M}\tilde{\omega}^\top = M_1 \end{cases}$

5: **return** $(\tilde{M}, \tilde{\omega})$

SIDIWO in the misspecified setting

- In the misspecified setting, the optimization solved is :

$$\min_{D \in \mathcal{D}_l} \sum_{i \neq j} \left(\sum_{r=1}^{d} (D M_{3,r} D^\top)_{i,j}^2 \right)^{1/2}$$

- Solve using the relation $D = (\tilde{M}\tilde{\Omega}^{1/2})^\dagger$ for the parameter pair $(\tilde{M}, \tilde{\omega}) \in \mathbb{R}^{d \times l} \times \mathbb{R}^l$

Theorem for misspecified setting

Theorem 3.1 *Let $D \in \mathcal{D}_l$ with rows $d_1, ..., d_l$, and let $\mathbb{I}_{r,s}$ denote the $r \times s$ identity matrix. The following facts hold under the hypotheses of Lemma 3.1:*

1. *For any row d_i, there exists at least one column of M such that $\langle d_i, \mu_j \rangle \neq 0$.*
2. *The columns of any $\tilde{M}$ satisfying $\tilde{M}\tilde{\Omega}^{1/2} = D^\dagger$ are a linear combination of those of M, laying in the best-fit l-dimensional subspace of the space spanned by the columns of M.*
3. *Let π be any permutation of $\{1, ..., d\}$, and let M_π and Ω_π be obtained by permuting the columns of M and Ω according to π. If $\langle \mu_i, \mu_j \rangle \neq 0$ for any i, j, then $((M_\pi \Omega_\pi^{1/2})\mathbb{I}_{k,l})^\dagger \notin \mathcal{D}_l$, and similarly $\mathbb{I}_{l,k}(M_\pi \Omega_\pi^{1/2})^\dagger \notin \mathcal{D}_l$.*

Theorem implications

- The constraint guarantees that the feasible solutions will lie in the best *l*-dimensional subspace approximating column space of *M* as:

$$D^{\dagger} = \tilde{M}\tilde{\Omega}^{1/2} = (E_l O_l) = U_l S_l^{1/2} O_l$$

- The columns of U_l are the left singular vectors of $M\Omega^{1/2}$ that fit the best *l*-dimensional subspace of the space generated by columns of *M*

Theorem implications

- From 2 and 3, non-orthogonal columns of *M* ensure $\tilde{M}$ cannot be sub-block of original *M*
- Instead, $\tilde{M}$ is non-trivial linear combination of its columns laying in the best *l*-dimensional subspace approximating column space of *M*
- Orthogonal columns of M results in the original space and the *l* column space coinciding for *l* largest ω_i, recovering top *l* topics

Hierarchical Topic Modeling

- SIDIWO recovers parameters where the *l* columns of $\tilde{M}$ offer a synthetic representation of the k original centers
- Each of the *l* vector (referred as pseudo-centers) is representative of a group of original centers
- A dataset can be iteratively split into 2 smaller subsets ($l = 2$) according to their similarity to the pseudo-centers

Hierarchical Topic Modeling

- Assignment is done using Maximum A Posteriori (MAP) to find the pseudo-center giving maximum conditional likelihood to each sample
- The method is generalizable to any latent variable model learned with the tensor method of moments (e.g. Latent Dirichlet Allocation)

Corpus Splitting Algorithm

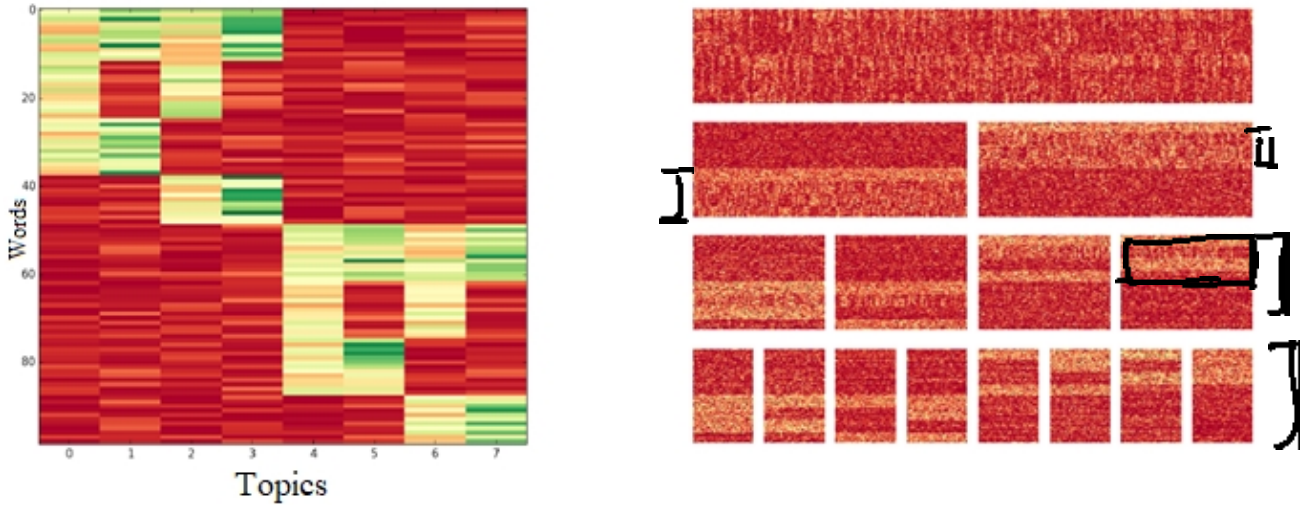
Algorithm 2 Splitting a corpus into two parts

Require: A corpus of texts $\mathcal{C} = (x^{(1)}, ..., x^{(n)})$.

1: Estimate M_1, M_2 and M_3.
2: Recover $l = 2$ pseudo-center with Algorithm 1 .
3: Project the Pseudo-center to the simplex
4: **for** $i \in [n]$ **do**
5: Assign the text $x^{(i)}$ to the cluster $Cluster(i) = \arg\max_j \mathbb{P}[X = x^{(i)}|Y = j, \tilde{\omega}, \tilde{M}]$, where $\mathbb{P}[X|Y = j, \tilde{\omega}, \tilde{M}]$ is the multinomial distr. associated to the j-th projected pseudo-center.
6: **end for**
7: **return** The cluster assignments $Cluster$.

- Recursively iterating this process produces a binary tree with higher depth nodes having distribution more concentrated on fewer topics

Experiment on Synthetic Data



Method	Adj. Rand Idx Mean	St. dev.	Run. Time
ALS	0.70	0.17	1.9 sec.
TPM	0.93	0.06	1.2 sec.
SVD	0.52	0.13	0.1 sec.
Rand. Proj.	0.72	0.06	16 min.
SIDIWO	0.98	0.01	0.4 sec.

Figure 1: Left: Visualization of the topics used to generate samples. Center: Hierarchy recovered from SIDIWO. Right: Mean and standard deviation over 10 runs for various methods along with run times.

- A dataset distributed as a single topic model is created with 100 vocabulary words. The 8 topics have an intrinsic hierarchical structure depicted in Figure 1: Left
- In this figure, topics are on the x-axis, words on the y-axis, and green (resp. red) points represents high (resp low) probability.

Experiment on Synthetic Data

- 400 samples are generated according to this model
- Corpus splitting algorithm is iteratively run to create a hierarchical binary tree with 8 leaves
- Results are displayed in Figure 1: center, where each chart represents a node of the tree and contains the heatmap of the samples clustered in that node

Experiment on Synthetic Data

- Each leaf contains samples from one of the topics and internal nodes group similar topics together
- The experiment is repeated 10 times with different random samples, with the averaged results in the table (Figure 1: Right)
- SIDIWO always recovers the original topic almost perfectly, unlike competing methods

Conclusion

- A MoM based Latent variable model learning algorithm is proposed
- SIDIWO proposes a constrained optimization formulation to produce meaningful results even in the misspecified setting
- This allows hierarchical clustering by running the algorithm recursively with $l = 2$, resulting in a divisive binary tree
- The algorithm recovers pseudo-centers that are representative of groups of original centers and are assigned data points using MAP