Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования «Московский физико-технический институт
(государственный университет)»

Факультет инноваций и высоких технологий
Кафедра «Анализ данных»

Ивашковский Иван Александрович

Методы инициализации в вероятностном тематическом моделировании

Выпускная квалификационная работа бакалавра по направлению
010600 Прикладные математика и физика

Научный руководитель:
д.ф.-м.н., профессор
К. В. Воронцов

Москва, 2016

Содержание

1 Введение

Одной из важных задач в области автоматической обработки текстов является задача выявления тем, присутствующих в документе. Это позволяет эффективно решать задачи информационного поиска, классификации документов, построения рекомендательных систем.

Тематическое моделирование - один из способов ее решения, в рамках которого строится модель текстовой коллекции, которая определяет, к каким темам относятся документы и какие слова образуют каждую тему. Наибольшее применение в тематическом моделировании находят вероятностные модели, осуществляющие «мягкую» кластеризацию документов по темам.

Вероятностное тематическое моделирование основано на нескольких базовых предположениях о коллекции документов. Во-первых, предполагается, что порядок слов в документе не важен для определения его тематики. Во-вторых, сама коллекция рассматривается как простая выборка пар «документ-слово», порожденная некоторым распределением, содержащим латентные переменные - темы. Задача состоит в нахождение этого распределения.

Одним из первых в качестве решения был предложен метод вероятностного латентного семантического анализа (PLSA) [5]. Дальнейшее его развитие привело к появлению метода латентного размещения Дирихле (LDA) [2] и к аддитивным регуляризационным моделям (ARTM) [10]. Все эти методы сходны тем, что они решают некоторую задачу матричного разложения итеративными оптимизационными методами.

Типичные проблемы итеративных методов - остановка в локальных оптимумах и зависимость конечного результата от начального приближения.

Целью данной работы является анализ влияния начальной инициализации на сходимость итеративных оптимизационных методов в задаче тематического моделирования.

Рассматриваемые инициализации сравниваются на различных наборах входных данных. После сравнения инициализаций между собой предлагаются методы их улучшения, повышающие скорость сходимости и улучшающие итоговый результат. В итоге даны практические рекомендации по использованию алгоритмов инициализации.

2 Вероятностное тематическое моделирование

2.1 Основные понятия и обозначения

D — множество (коллекция) текстовых документов, W — множество (словарь) всех употребляемых в них терминов.

Каждый документ $d \in D$ представляет собой последовательность n_d терминов $w_1, \ldots, w_{n_d}$ из словаря W.

Вероятностное пространство. Предполагается, что существует конечное множество тем T, и каждое вхождение термина w в документ d связано с некоторой темой $t \in T$.

Коллекция документов рассматривается как случайная и независимая выборка троек $(w_i, d_i, t_i),\ i = 1, \ldots, n$ из дискретного распределения $p(w, d, t)$ на конечном вероятностном пространстве $W \times D \times T$.

Термины w и документы d являются наблюдаемыми переменными, тема $t \in T$ является *латентной* (скрытой) переменной.

Гипотеза мешка слов. Предполагается, что порядок терминов в документах не важен для выявления тематики. Это предположение называют гипотезой «мешка слов» (bag of words).

Порядок документов в коллекции также не имеет значения; это предположение называют гипотезой «мешка документов».

Гипотеза «мешка слов» позволяет перейти к компактному представлению документа как подмножества $d \subset W$, каждому элементу которого, $w \in d$, поставлено в соответствие число n_{dw} вхождений токена w в документ d.

Гипотеза условной независимости. Предполагается, что появление слов в документе d по теме t зависит от темы, но не зависит от документа d.

Это предположение, называемое *гипотезой условной независимости*, допускает три эквивалентных представления:

$$\begin{aligned} p(w|d,t) &= p(w|t); \\ p(d|w,t) &= p(d|t); \\ p(d,w|t) &= p(d|t)p(w|t). \end{aligned} \tag{1}$$

В соответствии с формулой полной вероятности и гипотезой условной независимости верно следующее:

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d)$$

Это выражение описывает процесс порождения коллекции документов по известной тематической модели (распределениям $p(w|t)$ и $p(t|d)$).

Матричные обозначения Упорядочим слова, темы и документы каким-либо образом.

Тогда гипотеза «мешка слов» позволяет полностью описать коллекцию матрицей F размера $|W| \times |D|$, где $F_{w,d} = n_{d,w}$. При этом условные вероятности $p(w|t)$ и $p(t|d)$ образуют стохастические матрицы Φ размера $|W| \times |T|$ и Θ размера $|T| \times |D|$, где $\phi_{w,t} = p(w|t)$, $\theta_{t,d} = p(t|d)$.

2.2 Вероятностный латентно-семантический анализ

Первые шаги в тематическом моделировании были сделаны в работе Хоффмана Probabilistic latent semantic indexing, [6], где был предложен метод решения задачи, основанный на индексировании числа встречаемых в текстах слов. Предложенный метод был модернизирован и представлен как метод вероятностного латентного семантического анализа (Probabilistic Latent Semantic Analysis, PLSA, [5]).

Последующее развитие методов можно разделить на два подхода: первый заключался в добавлении априорных распределений в модель (например, априорное распределение на слова, как в работе, предлагающей метод LDA [2]). Второй добавляет регуляризацию на оптимизируемые параметры.

Однако в последующем в работах по ARTM - аддитивным регуляризиционным тематическим моделям [10], было показано, что большинство изменений, связанных с внесением априорного знания, можно описать как некоторые регуляризаторы.

Как уже говорилось, большинство методов решения задачи тематического моделирования сводятся к итеративным методам оптимизации - на каждой итерации матрицы Φ, Θ обновляются в соответствии с текущими их значениями. Оптимизация продолжается либо заранее фиксированное число итераций, либо пока не выполнится необходимый критерий останова.

Общий вид алгоритма приведён ниже:

Input: матрица F, число тем T, $\#$ итераций $iter_{\max}$;
Output: матрицы Φ и Θ;

Инициализировать Φ, Θ;
forall the $iter = 1, \ldots, iter_{\max}$ **do**
  $\Phi^{new} = G(F, \Phi^{old}, \Theta^{old})$;
  $\Theta^{new} = F(F, \Phi^{old}, \Theta^{old})$;
end

Algorithm 1: Общий вид итеративного алгоритма решения задачи тематического моделирования

В данной работе исследуются инициализации матриц в рамках наиболее популярной и простой тематической модели - вероятностного латентного семантического анализа (PLSA). Рассмотрим более подробно эту модель.

PLSA Для оценивания матриц Φ, Θ тематической модели по коллекции документов D с матрицей F максимизируется правдоподобие (плотность распределения) коллекции документов:

$$p(F, \Phi, \Theta) = \prod_{i=1}^{n} p(d_i, w_i) = \prod_{d \in D} \prod_{w \in d} p(w|d)^{n_{dw}} p(d)^{n_{dw}} \to max_{\Phi,\Theta}$$

Так как $\forall d \in D\ p(d) = const$, то применив гипотезу условной независимости, из которой следует что $p(w|d) = \sum_{t \in T} p(w|t)p(t|d)$, для логарифма правдоподобия получим выражение:

$$L(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt}\theta_{td} \to max_{\Phi,\Theta}$$

Задача решается при некоторых естественных ограничениях на матрицы Φ, Θ. В силу их стохастичности сумма значений в столбце должна быть равна 1, а значения неотрицательны:

$$\sum_{winW} \phi_{wt} = 1, \phi_{wt} \geq 0$$

$$\sum_{tinT} \theta_{td} = 1, \theta_{td} \geq 0$$

EM-алгоритм Для максимизации правдоподобия в модели PLSA используют итеративный алгоритм - EM-алгоритм, который представляет собой метод простых итерация применительно к данной задаче.

Каждая итерация алгоритма состоит из двух шагов.

1. На E-шаге (Expectation) по текущим матрицам Φ и Θ высчитываются некоторые вспомогательные значения:

$$p_{tdw} = \frac{\phi_{wt}\theta_{td}}{\sum_{s \in T} \phi_{wt}\theta_{sd}} \tag{2}$$

2. На M-шаге (Maximization) посчитанные значения используются для обновления матриц Φ и Θ:

$$\phi_{wt} = \frac{n_{wt}}{\sum_{v \in W} n_{vt}},\ n_{wt} = \sum_{d \in D} n_{dw} p_{tdw}$$
$$\theta_{td} = \frac{n_{td}}{\sum_{s \in T} n_{ds}},\ n_{td} = \sum_{w \in W} n_{dw} p_{tdw}$$

2.3 Проблема инициализации и постановка задачи

Важный этап EM-алгоритма - выбор начального приближения. Оно влияет как на скорость сходимости, так и на конечный результат EM-алгоритма, который может остановиться в одном из локальным оптимумов.

На данный момент на практике наиболее распространенным вариантом инициализации является случайная инициализация. Но она, как будет показано в моей работе, не является оптимальной.

В рамках моей работы были рассмотрены и сравнены между собой несколько алгоритмов инициализаций матриц Φ и Θ. Цель данных исследований - провести сравнительный анализ инициализаций по времени работы и влиянию на сходимость EM-алгоритма, рассмотреть возможные способы повышения их качества и выработать практические рекомендации по их применения.

3 Методы инициализации

К настоящему времени работ, посвященных именно вопросам инициализации матриц в задаче тематического моделирования, довольно мало.

Существуют такие статьи по сходной тематики - неотрицательным матричным разложениям (nonnegative matrix factorization). В такой постановке задачи искомые матрицы не являются стохастическими, а оптимизируемая метрика - не правдоподобие, а норма Фробениуса.

Примерами таких работ являются:

- Выпускная работа [Шадриков А., 2015] - исследование различных методов NMF (в том числе PLSA), инициализация алгоритмом Ароры
- SVD based initialization [3] - применение SVD для инициализации матриц
- Algorithms, Initializations, and Convergence for the NMF [7] - рассмотрены несколько методов инициализации матриц для NMF, в том числе простейшая кластеризация исходных данных

Применительно к задаче тематического моделирования есть работы, предлагающие конкретную инициализацию, например:

- Topic Models Regularization and Initialization for Regression Problems [9]
- Robust Initialization for Learning Latent Dirichlet Allocation[8]

Ряд работ предлагает альтернативный способ решения задачи тематического моделирования, основанный на введении дополнительных предположений о коллекции. Такое решение можно использовать как начальное приближение для EM-алгоритма. Ярчайшим примером такой работы является A Practical Algorithm for Topic Modeling with Provable Guarantees [1].

В моей работе были рассмотрены следующие инициализации:

3.1 Случайная инициализация

Столбцы матриц Φ, Θ генерируются из симметричного распределения Дирихле:

$$f(x_1, \ldots, x_K; \alpha) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=1}^{K} x_i^{\alpha-1}$$

$$x_i \geq 0, \sum_{i=1}^{K} x_i = 1, \alpha > 0$$

В текущей работе рассматривались два способа заполнения матриц - разреженный и равномерный. Параметры распределений Дирихле для матриц Φ, Θ $\alpha = 0.05$, 0.1 в случае разреженных матриц и $\alpha = 1.0$, 1.0 в случае равномерного заполнения.

На текущий момент это наиболее распространенная инциализация на практике.

3.2 Алгоритм Ароры

Алгоритм Ароры основан на дополнительном предположении о темах, которые существуют в коллекции документов. Более подробное описание можно найти в работе [1].

Допустим, в каждой теме найдется такое слово, что оно не встречается ни в какой другой теме - якорное слово.

Определение 1. Если в теме t можно выделить слово w такое, что $\phi_{wt} = p(w|t) > 0$ и $\forall s \neq t \; \phi_{ws} = p(w|s) = 0$, то такое слово называется якорным.

Предположении о существовании якорного слова для каждой темы эквивалентно существованию диагональной подматрицы размера $T \times T$ в матрице Φ.

Алгоритм имеет две стадии - на первой стадии находятся все якорные слова, на второй стадии по ним происходит восстановление матрицы Φ, а затем Θ. Рассмотрим подробнее эти стадии:

1. Для нахождения якорных слов используется матрица $Q_{w_2w_1} = p(w_2|w_1)$, где $w_1, w_2 \in W$, которая получается нормировкой столбцов из матрицы соупотреблений слов $p(w_2, w_1)$. Полагая вероятность документа пропорциональной его длине $p(d) \sim n_d$, а употребление слов в контексте одного документа независимым $p(w_1, w_2, d) = p(w_1, d)p(w_2, d)$, матрицу соупотреблений слов можно оценить как $p(w_1, w_2) = \sum_{d \in D} p(w_1, w_2, d) = \sum_{d \in D} p(w_1, d)p(w_2, d) = \sum_{d \in D} p(w_1|d)p(d)p(w_2|d)p(d) \approx norm(FF^T)$, так как $F_{w,d} \approx p(w|d)n_d \sim p(w|d)p(d)$

   Якорные слова обладают свойством, которое позволяет быстро их находить. А именно если w_t - якорное слово для темы t, то в силу определения якорного слова и формулы Байеса $p(t|w_t) = p(w_t|t)p(t)/p(w_t)$ будет верно $Q_{w_2w_t} = \sum_{s \in T} p(w_2|s)p(s|w_t) = p(w_2|t)$

С учетом этого произвольный элемент матрицы Q можно записать как: $Q_{w_2w_1} = \sum_{t\in T} p(w_2|t)p(t|w_1) = \sum_t C_{tw_1}Q_{w_2w_t}$ где $C_{tw_1} = p(t|w_1)$. Это значит, что столбцы матрицы Q являются выпуклой комбинацией столбцов, соответствующих якорным словам.

Получается, что для их нахождения достаточно найти выпуклую оболочку столбцов матрицы Q - ее вершины и будут столбцами, соответствующими якорным словам. Эта задача не имеет точного решения за полиномиальное время, но с некоторой потерей точности ее можно решить жадным алгоритмом:

Input: матрица Q - содержит столбцы размерности W, точность ϵ, количество вершин симплекса K

Output: K точек, близких к вершинам симплекса

Спроецировать точки на пространство меньшей размерности $4\log W/\epsilon^2$;
Выбрать самую удаленную от начала координат точку $S = \{d_0\}$;
for $i = 1, \ldots, K-1$ **do**
    Выбрать d_i - наиболее удаленную точку от подпространства, порожденного выбранными точками $span(S)$;
    $S += \{d_i\}$;
end

for $i = 1, \ldots, K-1$ **do**
    Выбрать v_i - наиболее удаленную точку от подпространства, порожденного точками $span(S\backslash\{d_i\})$;
    Обновить $d_i = v_i$;
end
Вернуть S;

Algorithm 2: Алгоритм поиска приблизительной выпуклой оболочки

Сложность этого алгоритма $O(W^2 + WK/\epsilon^2)$, при этом мы получим его вершины с точностью до расстояния порядка $O(\epsilon)$

2. Теперь, зная якорные слова, можно получить коэффициенты разложения каждого столбца по якорным столбцам $C_{tw_1} = p(t|w_1)$ - в оригинальной работе предлагается решить задачу оптимизации с использованием экспоненциального градиентного спуска. Затем по формуле Байеса $p(w|t) = norm(p(t|w)p(w))$ восстанавливается матрица Φ. Вероятности $p(w)$ можно подсчитать по матрице Q.

   В моей работе матрица Θ получается как МНК решение задачи $F_{norm} = \Phi\Theta$ с последующим занулением отрицательных значений и нормировкой столбцов на сумму значений в нем, чтобы получить стохастическую матрицу. Здесь F_{norm} - частотная матрица коллекции, ее столбцы получаются из столбцов матрицы F делением на сумму элементов в нем.

В данной работе был реализован стандартный однопоточный алгоритм Ароры. При этом существует параллельная реализация алгоритма Ароры, которая приведена приведена в этой статье [Ding, Efficient Distributed Topic Modeling with Provable Guarantees][4]

3.3 Кластеризация слов

Алгоритм Ароры предполагает наличие диагональной подматрицы в матрице Φ. Конечно, на практике это свойство выполняется не всегда. Но существуют способы искусственно добиться этого свойства.

Наиболее простым способом достижения этого является кластеризация слов (каждое из которых представляет из себя строку нормированной матрицы F_{norm}) на T кластеров. Если найденные центроиды интерпретировать как темы - то есть заполнить ими строки матрицы Θ (с последующей нормировкой в силу стохастичности), то из приблизительного разложения $F_{norm} \approx \Phi\Theta$ следует, что матрица Φ в первом приближении будет иметь

следующий вид:

$$\Phi \sim \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Используемая метрика - L2.

Матрица Φ восстанавливается по заполненной матрице Θ как МНК решение задачи $F_{norm} = \Phi\Theta$ с занулением отрицательных значений и нормировкой столбцов.

3.4 Кластеризация tf-idf слов

Более разумный выбор пространства, в котором происходит кластеризация, может улучшить качество инициализации.

Многие слова могут оказаться незначащими, поэтому для их отброса предлагается преобразовывать матрицу F не к частотному, а к TF-IDF виду. Дальнейший алгоритм повторяет алгоритм предыдущего пункта.

3.5 Сингулярное разложение

В следующей работе [3] был предложен способ инициализации матриц на основе сингулярного разложения матрицы $F = USV$.

В задаче неотрицательного матричного разложения сингулярное разложение дает точный ответ на задачу оптимизации. Но в задаче тематического моделирования оптимизируется правдоподобие, а не норма Фробениуса,

и матрицы должны быть не только неотрицательны, но и стохастичны. Поэтому после нахождения сингулярного разложения предлагается применить следующие шаги:

1. Для заполнения матрицы матрицы Φ используются столбцы u матрицы U, для заполнения матрицы Θ - строки v матрицы V, соответсвующие наибольшим сингулярным значениям

2. Для каждого такого вектор-столбца матрицы U занулением отрицательны значений выделяется его положительная часть $u_+ = uI[u > 0]$ и аналогично отрицательная часть $u_- = -uI[u < 0]$. Аналогичное действие производится для строк матрицы V

3. Для заполнения используются либо оба положительных вектора, либо оба отрицательных. Критерий выбора таков - выбираются тот вектор-столбец и вектор-строка, у которых произведение норм больше. Обозначим их за $u_{(0)}, v_{(0)}$

4. Заполняется очередной столбец матрицы Φ:
   $\Phi_{.i} = \sqrt{S_{ii} * norm(u_{(0)}) * norm(v_{(0)})} * u_{(0)}/norm(u_{(0)})$

5. Заполняется очередная строка матрицы Θ:
   $\Theta_{i.} = \sqrt{S_{ii} * norm(u_{(0)}) * norm(v_{(0)})} * v_{(0)}/norm(v_{(0)})$

Завершающим шагом является нормировка столбцов матриц Φ, Θ

3.6 Модификации инициализаций

В моей работе было рассмотрено несколько различных модификаций предложенных выше инициализаций.

Кластеризация документов Логичным продолжением идеи кластеризации слов является идея кластеризации документов. Алгоритм инициализации практически не изменяется - только кластеризуются не строки, а столбцы матрицы F_{norm} и центроидами заполняется матрица Φ.

Матрица Θ в алгоритме Ароры Эта модификация предполагает заполнение матрицы Θ значениями $1/\mathrm{T}$ в алгоритме Ароры вместо решение задачи $F_{norm} = \Phi\Theta$. Такой способ заполнения имеет право на существование, потому что обычно правдоподобие сильнее зависит от матрицы Φ, чем от Θ. Гипотеза, стоящая за такой инициализацией, состоит в том, что достаточно инициализировать только левую матрицу - матрицу Φ, а всю работу по нахождению Θ сбросить на EM-алгоритм.

Такой метод инициализации Θ гораздо более эффективен с точки зрения вычислительных затрат.

Комбинирование случайной и неслучайной инициализации Проведенные в дальнейшем эксперименты указывают на то, что случайные инициализации могут сойтись к лучшим значениям правдоподобия, но при этом оптимизация занимает большее количество итераций, чем в случае других инициализаций.

Один из методов объединить достоинства случайного и неслучайного подхода заключается в привнесении дополнительной случайности в неслучайные инициализация. Добавления шума к исходным данным - стандартный метод в машинном обучении, который препятствует переобучению модели.

Применительно к задаче тематического моделирования можно использовать следующий алгоритм - взять в качестве инициализации выпуклую комбинацию матриц, выданных алгоритмом инициализации, и случайных

матриц в соответствии со следующими формулами:

$$\Phi = \alpha\Phi_{random} + (1 - \alpha)\Phi_{init}$$

$$\Theta = \alpha\Theta_{random} + (1 - \alpha)\Theta_{init}$$

Встряхивание весов - jogging of weights Логичным развитием такого подхода является метод встряхивания весов. Его идея состоит в использовании указанной выше выпуклой комбинации матриц на каждой итерации EM-алгоритма.

А именно - на каждой итерации EM-алгоритма будем строить выпуклую комбинация текущих матриц Φ,Θ и случайных матриц:

$$\Phi = \alpha\Phi_{random} + (1 - \alpha)\Phi_{current}$$

$$\Theta = \alpha\Theta_{random} + (1 - \alpha)\Theta_{current}$$

. При этом параметр α должен убывать с увеличением номера итерации.

В качестве примесей я предлагаю использовать сильно разреженные матрицы, а не равномерно заполненные, из тех соображения, что в таком случае их добавление будет изменять меньшее число значений в текущих матрицах Φ, Θ. Это должно не сильно влиять на сходимость, если EM-алгоритм попадет в глобальный оптимум.

4 Вычислительные эксперименты

Во всех экспериментах исследовалась модель PLSA, рассматривались первые 100 итераций EM-алгоритма

Для определения качества разложения были использованы метрики:

Перплексия $exp(-L/N)$, где L - логарифм правдоподобия $L = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt} \theta_{td}$, N - размер коллекции (общее число слов в ней)

Это основная метрика качества - фактически она является оптимизируемым функционалом.

Качество восстановления F Расстояние Хеллингера между нормализованной матрицей F_{norm} (сумма элементов в столбце равна 1) и произведением $\Phi\Theta$ показывает качество восстановления вероятностей $p(w|d)$. Расстояние Хеллингера между матрицами определим как среднее расстояние между столбцами этих матриц.

Расстояние Хеллингера между вероятностными распределениями определено следующей формулой:

$$Hellinger(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{P_i} - \sqrt{Q_i})^2}$$

Качество восстановления Φ, Θ Если были известны матрицы $\Phi_{real}, \Theta_{real}$, из которых порождалась коллекция (случай синтетических или полусинтетических данных) то можно измерить, насколько хорошо мы их восстановили.

Для этого ищется соответствие столбцов в двух матрицах (известной и восстановленной EM-алгоритмом) венгерским алгоритмом. Используемая для поиска соответствия норма - L_2. Затем вычисляется среднее расстояние Хеллингера между соответствующими друг другу столбцами матриц.

Интерпретируемость и различность тем из Φ Для конечного результата матрицы «слова-темы» Φ вычислялись следующие метрики, опи-

сывающие найденные темы:

- *Mean PMI* - в каждой теме выбираются топ-10 самых вероятных слов. Для каждой пары слов из топ-10 подсчитывается PMI (pointwise mutual information), затем эта величина усредняется по всем парам. Затем полученная величина PMI темы усредняется по всем темам.

$$PMI = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

  Вычисление PMI происходит аналогично вычислению матрицы Q в алгоритме Ароры. Полагая вероятность документа пропорциональной его длине $p(d) \sim n_d$, а употребление слов в контексте одного документа независимым $p(w_1, w_2, d) = p(w_1, d)p(w_2, d)$, матрицу соупотреблений слов $p(w_i, w_j)$ при большом количестве документов можно оценить как $norm(FF^T)$ - сумма всех элементов должна быть равна 1.

  PMI является метрикой, которая коррелирует с человеческими оценками интерпретируемости тем. Чем она выше, тем чаще слова из одной темы встречаются в одном контексте.

- *Mean NearHellinger* - для каждой темы рассчитываются расстояния Хеллинегра до остальных тем (темы представляют собой столбцы матрицы Φ), затем выбирается наиболее близкая по этой метрике тема к текущей. Эта величина усредняется по всем темам.

  Такая метрика позволяет видеть, насколько различны выделенные темы. Чем выше это значение, тем качественнее найденное разложение.

Визуальное представление Для наглядного представления результаты были визуализированы в виде графиков. На них показаны зависимости метрик от номера итерации EM-алгоритма. На полусинтетических данных представлена зависимость метрик от параметра модели α.

Для всех методов проводилось 10 запусков - полупрозрачная область отображает разброс качества в этих 10 запусках (кривые, соответсвующие max и min конечному значению метрики).

На всех графиках продемонстрированы наиболее интересные участки зависимостей, на которых наиболее заметно различие алгоритмов.

Конфигурация оборудования Все эксперименты проводились на MacBook Air, процессор 2 GHz Intel Core i7, память 8 ГБ 1600 МГц DDR3

4.1 Синтетические данные

Столбцы матриц Φ, Θ генерируются из симметричного распределения Дирихле. Так же, как и в одном из методов инициализации, рассматривались два возможных способа заполнения матриц - разреженный и близкий к равномерному.

При перемножении этих матриц получается частотная матрица термины-документы $F_{norm} = p(w|d)$. Чтобы перейти от частот терминов в документах к количеству упоминаний терминов, каждый столбец матрицы F_{norm} умножается на случайное число от 100 до 8000 - длину документа.

В таком методе известно точное разложении матрицы F, поэтому можно сравнивать получающиеся в результате работы EM-алгоритма матрицы с матрицами, используемыми для генерации.

Для генерации использовались следующие параметры:

- F имеет размер 2000x500 - 2000 слов с словаре и 500 документов в коллекции
- Число тем - 70

- Параметр распр. Дирихле Φ - 0.05 в случае разреженных матриц и 0.5 во втором случае
- Параметр распр. Дирихле Θ - 0.1 в случае разреженных матриц и 0.6 во втором случае

Разреженные синтетические данные Качество восстановления коллекции и порождающих матриц:

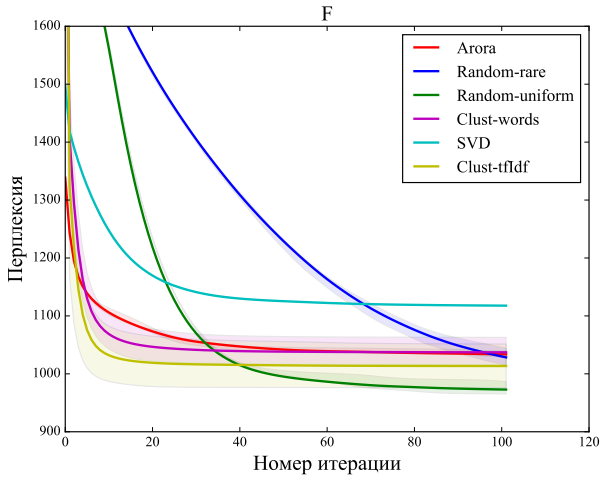

Рис. 1: Перплексия

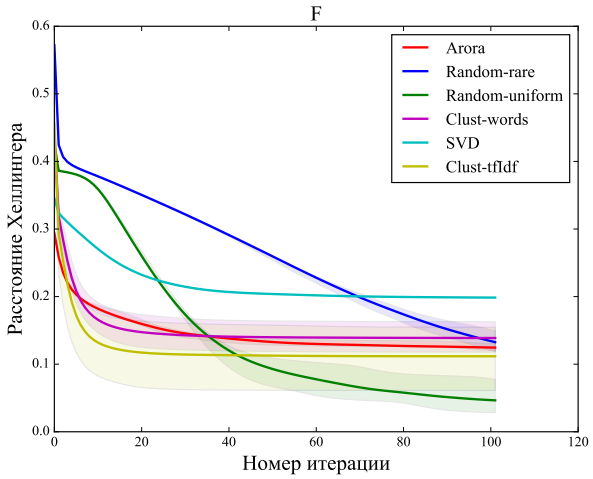

Рис. 2: Расстояние Хеллингера

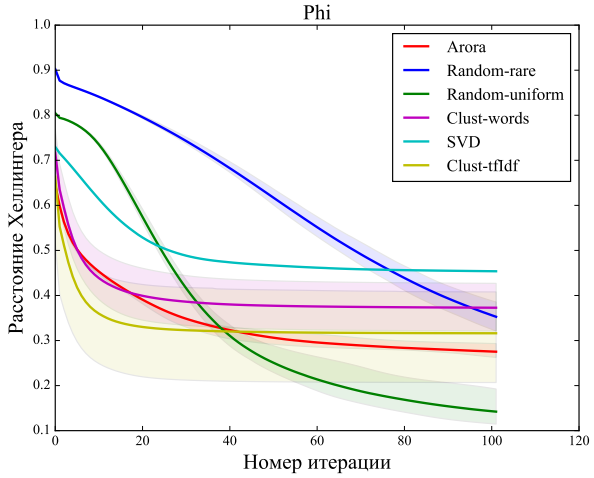

Рис. 3: Расстояния Хеллингера, матрица Φ

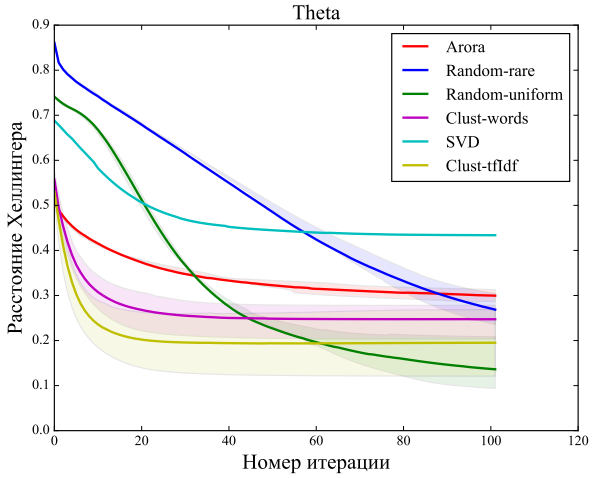

Рис. 4: Расстояния Хеллингера, матрица Θ

Наилучшим образом проявляют себя случайные инициализации, но при этом для сходимости им нужно гораздо большее число итераций (100 против 20). Из неслучайных инициализаций матрица Φ лучше всего восстанавливается алгоритмом Ароры.

Метрики качества матрицы слова-темы Φ и время работы			
	Mean PMI	Mean NearHellinger	Init time, сек.
Arora	0.845	0.76	8.52
Random-rare	0.818	0.86	0.027
Random-uniform	0.867	0.82	0.016
Clust-words	0.841	0.85	0.38
SVD	0.826	0.79	0.97
Clust-tfIdf	0.853	0.86	0.41

100 итераций EM-алгоритма в среднем занимали 10 секунд.

По метрикам матрицы Φ лучшие результаты демонстрируют обе кластеризации.

Время работы алгоритма Ароры сравнимо со временем оптимизации, но стоит отметить, что само нахождения якорных слов занимает около 1 секунды, остальные время тратится на поиск коэффициентов разложения столбцов Q по якорным столбцам.

Эта процедура - решение оптимизационной задачи по нахождению коэффициентов разложения для каждого столбца - может быть эффективно распараллелена и ускорена подбором шага градиентного спуска.

Равномерные синтетические данные Качество восстановления коллекции:

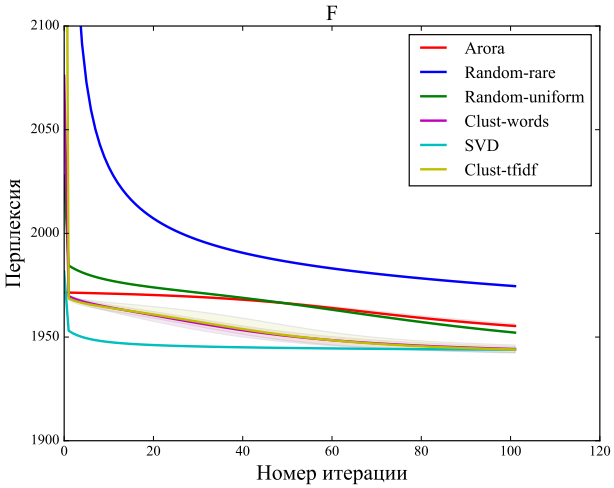

Рис. 5: Перплексия

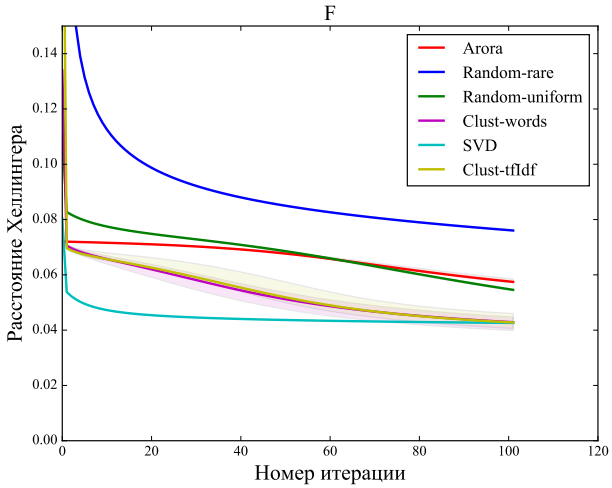

Рис. 6: Расстояния Хеллингера

Качество восстановления порождающих матриц:

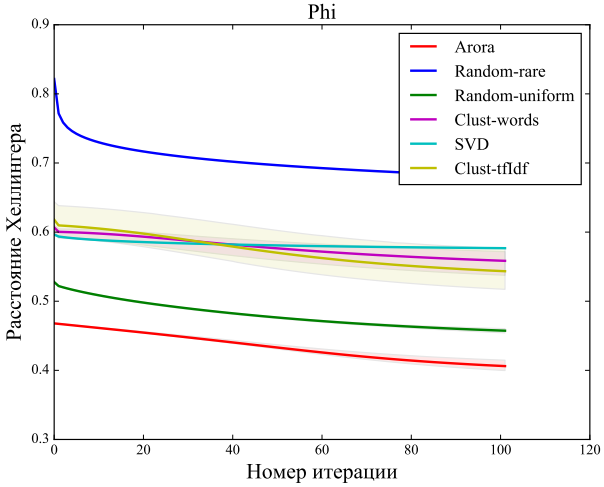

Рис. 7: Расстояния Хеллингера, матрица Φ

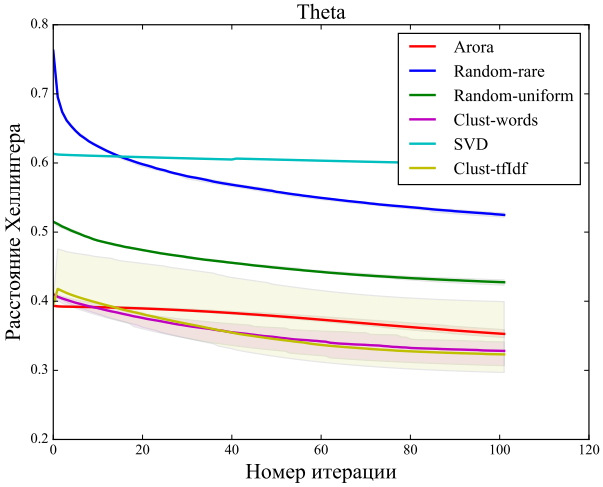

Рис. 8: Расстояния Хеллингера, матрица Θ

Видно, что лучшими являются инициализации методом SVD и кластеризацией. Алгоритм Ароры похож поведением на равномерную случайную инициализации. При этом он опять лучше всего восстанавливается матрицу Φ.

Метрики качества матрицы слова-темы Φ и время работы			
	Mean PMI	Mean NearHellinger	Init time, сек.
Arora	0.020	0.34	8.40
Random-rare	0.007	0.74	0.027
Random-uniform	0.014	0.41	0.016
Clust-words	0.019	0.60	0.38
SVD	0.011	0.58	0.97
Clust-tfIdf	0.021	0.58	0.551

100 итераций EM-алгоритма в среднем занимали 9 секунд.

Малые значения PMI обусловлены тем, что распределение близко к равномерному и слову употребляются практически независимо. Кластеризации показали наилучший результат по этим метрикам. Про быстродействие выводы остаются такими же, как в предыдущем пункте.

4.2 Реальные коллекции

Для сравнения результатов на реальных коллекциях были выбраны коллекции из репозитория UCI:

1. Коллекция документов с конференции nips (12419 терминов, 1500 документов)

2. Статьи блога kos (6906 терминов, 3430 документов)

Количество тем для восстановления - 100.

Результаты на Nips Качество восстановления коллекции:

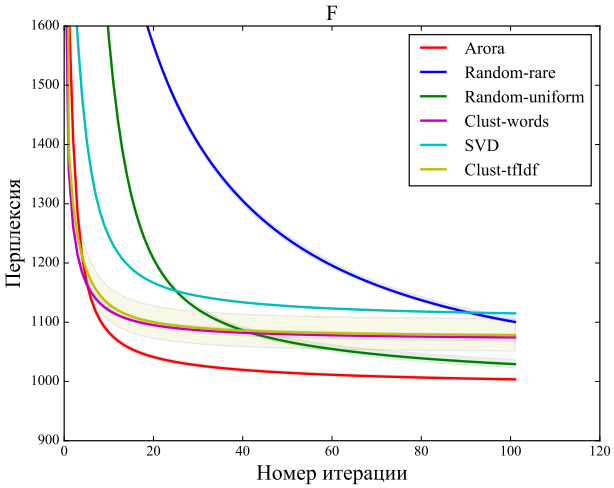

Рис. 9: Перплексия

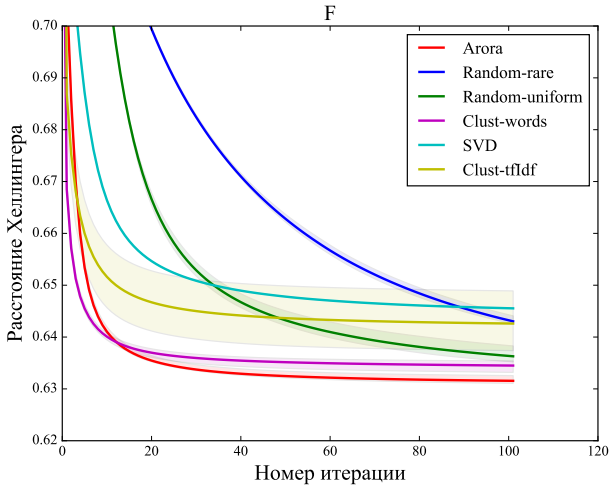

Рис. 10: Расстояния Хеллингера

Метрики качества матрицы слова-темы Φ и время работы			
	Mean PMI	Mean NearHellinger	Init time, сек.
Arora	0.68	0.64	419.25
Random-rare	0.68	0.79	0.22
Random-uniform	0.65	0.68	0.14
Clust-words	0.87	0.82	7.68
SVD	0.79	0.79	71.56
Clust-tfIdf	0.68	0.74	13.90

100 итераций EM-алгоритма в среднем занимали 270 секунд.

По основным метрикам качества наилучшие результаты в экспериментах показал алгоритм Ароры. Он сходится к лучшим значениям перплексии за малое число итераций, при этом показывая сходные результаты со случайными инициализациями по метрикам матрицы Φ.

Алгоритм кластеризации слов демонстрирует схожие результаты по метрике «расстояние Хеллингера», но перплексия не достигает таких же низких значений, как у алгоритма Ароры. При этом выделенные при такой инициализации темы лучше по PMI и межтемному расстоянию Хеллингера.

Нахождение якорных слов в алгоритме Ароры происходило в среднем за 30 секунд, остальное время занимает восстановление по ним матрицы Φ.

Результаты на Kos Качество восстановления коллекции:

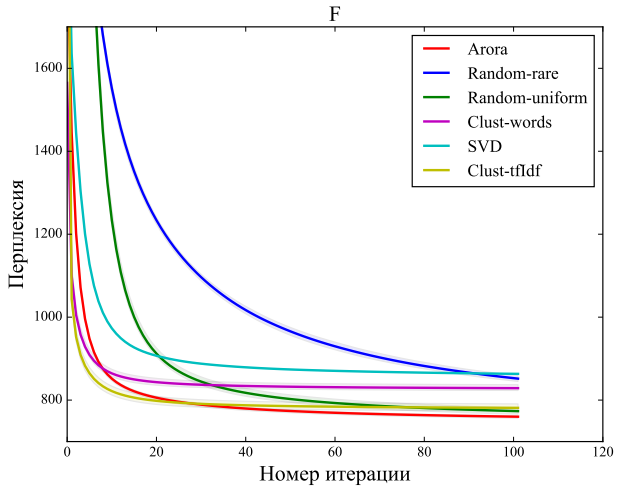

Рис. 11: Перплексия

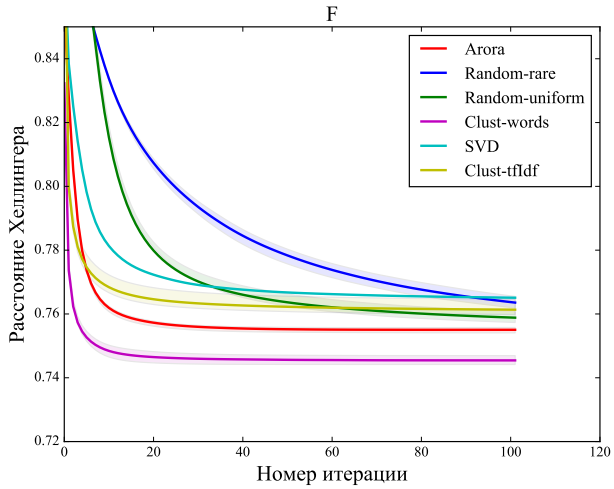

Рис. 12: Расстояние Хеллингера

Метрики качества матрицы слова-темы Φ и время работы			
	Mean PMI	Mean NearHellinger	Init time, сек.
Arora	0.86	0.75	134.10
Random-rare	0.65	0.85	0.19
Random-uniform	0.55	0.78	0.11
Clust-words	0.80	0.86	8.71
SVD	0.87	0.85	65.87
Clust-tfIdf	0.85	0.82	10.21

100 итераций EM-алгоритма в среднем занимали 340 секунд.

В целом результаты совпадают с результатами на nips. Стоит отметить, что по расстоянию Хеллингера кластеризация слов лучше, чем алгоритм Ароры. Также интересно, что по метрикам матрицы Φ случайные инициализации показывают худшие результаты.

Среднее время нахождения якорных слов было около 15 секунд, что опять на порядок меньше общего времени работы алгоритма Ароры.

По результатам эксперимента можно сделать вывод, что для минимизации перплексии наилучшим выбором будет алгоритм Ароры. В случае, когда мы хотим наилучшим образом приблизить частоты встречаемости слов, больше подходит кластеризация слов.

Таким случаем, например, может быть коллекция довольно длинных документов. Тогда частоты встречаемости слов уже довольно хорошо приближают вероятности распределения $p(w|d)$, и стремление приблизить матрицу коллекции по расстоянию Хеллингера выглядит более логичным, чем стремление учитывать степень доверия документу в соответствии с его длиной (правдоподобие коллекции).

4.3 Полусинтетические данные

Эксперименты на сгенерированных коллекциях имеют преимущество перед экспериментами на реальных, потому что мы знаем искомое разложение матрицы. Для того, чтобы воспользоваться таким преимуществом, при этом используя реальную коллекцию для экспериментов, был предложен метод построения полусинтетических данных. Он заключается в следующем - строится семейство задач:

1. Берется матрица «термины-документы» F_1 какой-нибудь реальной коллекции

2. На ней решается задача тематического моделирования, затем полученные матрицы Φ,Θ перемножаются. Столбцы полученной частотной матрицы $p(w|d)$ умножаются на общее количество слов в документе, который соответствует данному столбцу. Таким образом получается матри-

ца F_0

3. Ставится новая задача с матрицей «термины-документы» $F_\alpha = \alpha F_1 + (1-\alpha)F_0$.

   Таким образом, меняя параметр α, можно плавно менять матрицу, непрерывно переходя от синтетических данных (для которых известно разложение Φ,Θ) к реальным. При этом важным является тот факт, что синтетическая матрица получена на основе реальной. Таким образом матрица реальной коллекции выступает в качестве шумовой к синтетической при малых параметрах α и наоборот при больших.

В данной работе эксперименты проводились на коллекциях nips, kos. Задача тематического моделирования для генерации F_0 решалась при случайной разреженной инициализации методом PLSA.

Число тем, используемых для восстановления - 100.

Результаты на Nips Качество восстановления в зависимости от α:

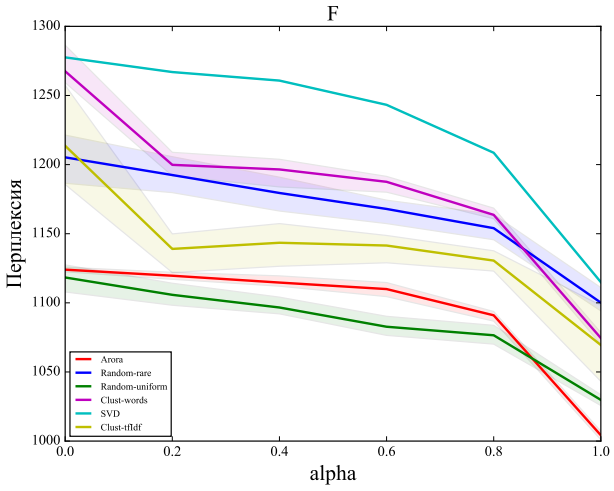

Рис. 13: Перплексия

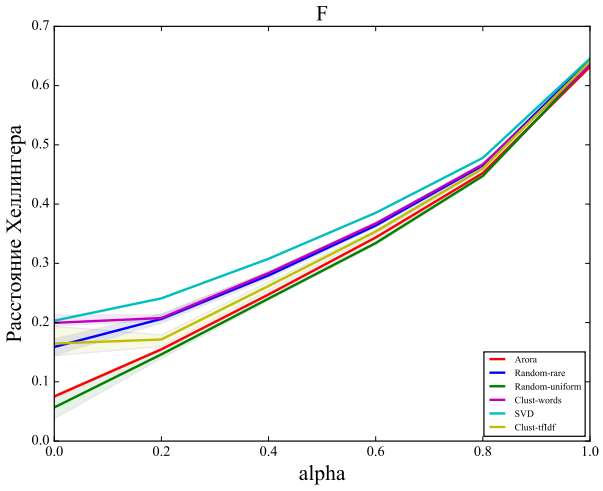

Рис. 14: Расстояние Хеллингера

Качество восстановления порождающих матриц в зависимости от α:

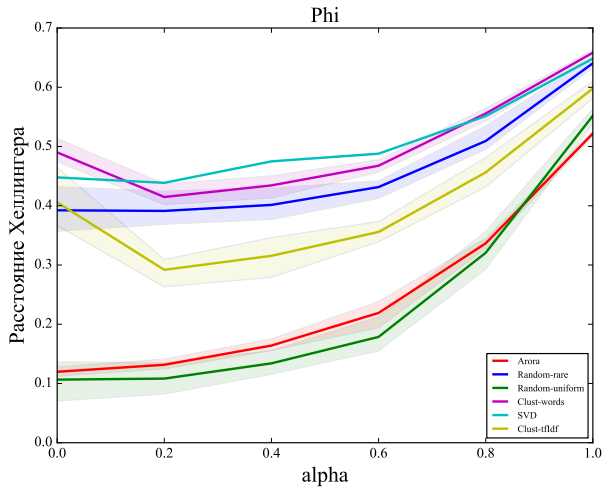

Рис. 15: Расстояние Хеллингера, матрица Φ

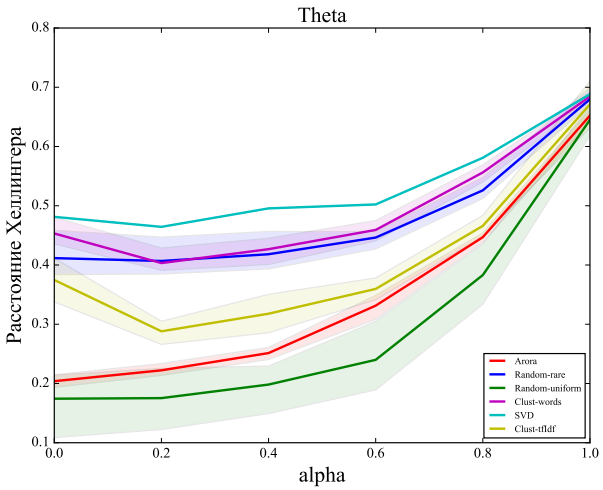

Рис. 16: Расстояние Хеллингера, матрица Θ

Результаты на Kos Качество восстановления в зависимости от α:

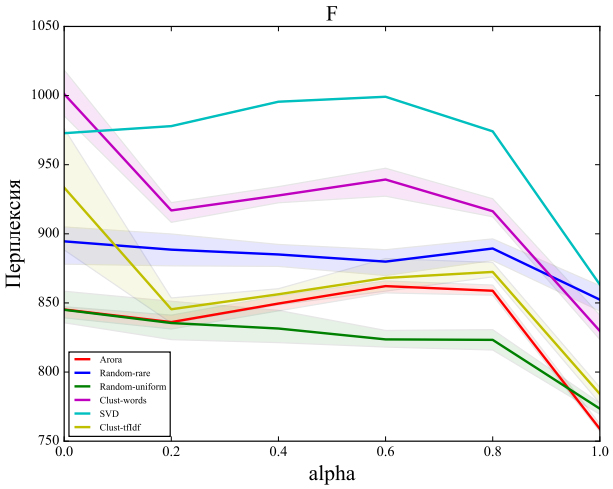

Рис. 17: Перплексия

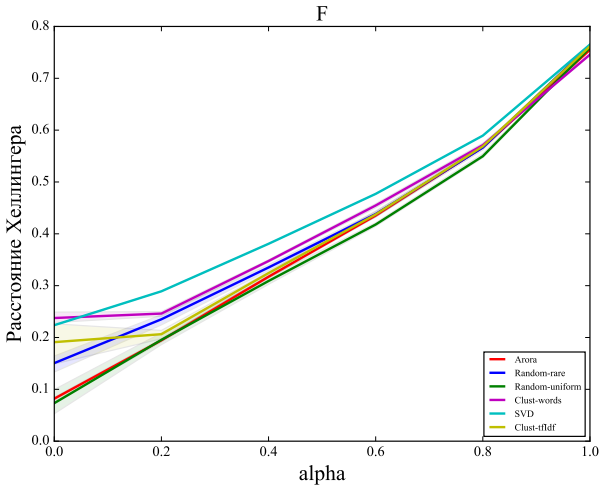

Рис. 18: Расстояние Хеллингера

Качество восстановления порождающих матриц в зависимости от α:

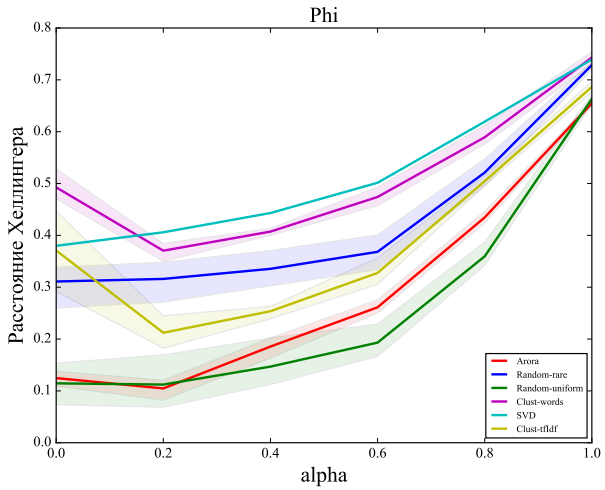

Рис. 19: Расстояние Хеллингера, матрица Φ

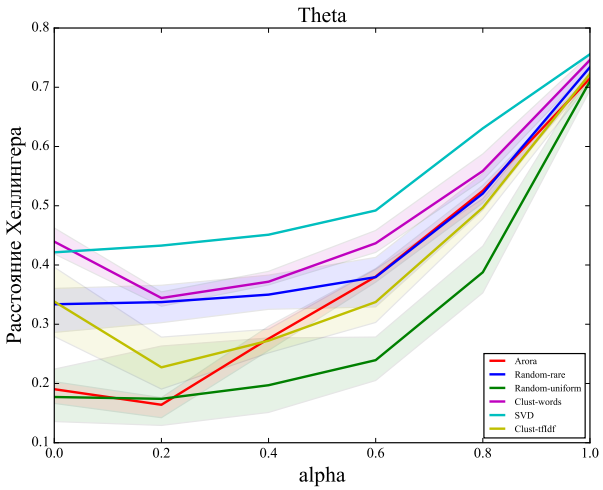

Рис. 20: Расстояние Хеллингера, матрица Θ

Графики перплексии и расстояние Хеллингера в целом показывают те же результаты, что и ранее на синтетических и реальных коллекциях.

Интерес представляет качество нахождения матриц Φ, Θ. Видно, что кластеризация в целом превосходит случайную разреженную инициализацию, а алгоритм Ароры ведет себя примерно как случайная равномерная инициализация.

Один из важных выводов, который можно сделать - результаты на реальных коллекциях и на синтетических данных, сгенерированных каким-либо образом, могут быть несравнимы между собой. Видно, как сильно изменяется картина при переходе от параметра $\alpha = 0$ до 1. Именно поэтому в моей работе были проделаны эксперименты как на реальных, так и на синтетических данных.

4.4 Единичная матрица

Интересно рассмотреть случаи, в которых существует и известно точное разложение. В моей работе рассматривался один такой случай - единичная матрица. Для нее известно разложение $E = EE$, именно оно дает максимум правдоподобия.

Качество восстановления коллекции для единичной матрицы 70x70:

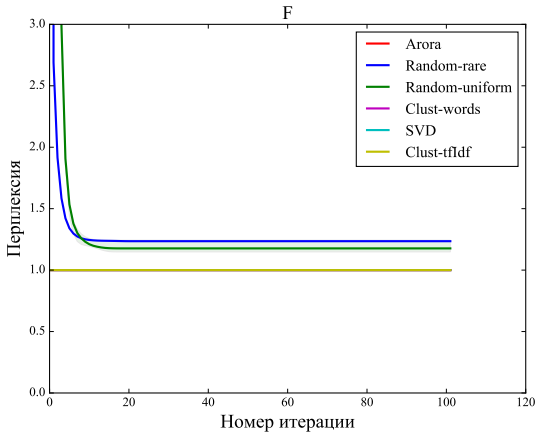

Рис. 21: Перплексия

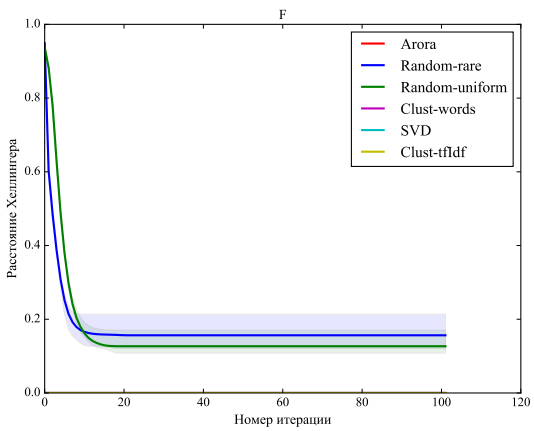

Рис. 22: Расстояние Хеллингера

Качество восстановления порождающих матриц:

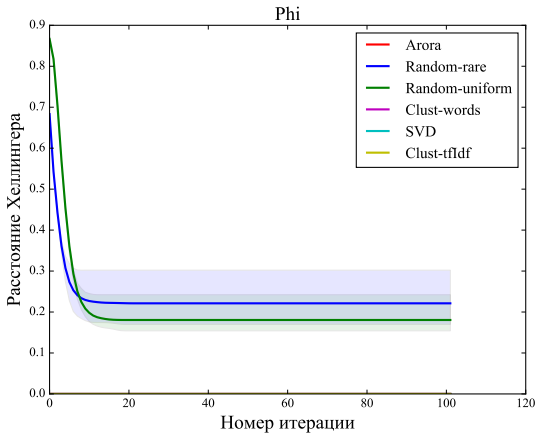

Рис. 23: Расстояние Хеллингера, матрица Φ

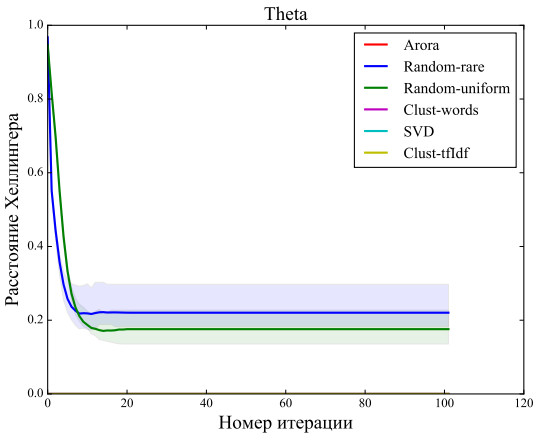

Рис. 24: Расстояние Хеллингера, матрица Θ

Примечательно, что все алгоритмы инициализации, кроме случайных, сразу нашли искомое разложение $E = EE$.

Похожие матрицы могут возникать в реальных задачах - именно благодаря возникшему прецеденту появилась идея проверить разложение $E = EE$.

Случайная инициализация попадает в локальный оптимум, из которого не может выбраться. Это указывает на большие недостатки у случайных инициализаций в вопросе последующего застревания EM-алгоритма в локальном оптимуме.

4.5 Сравнение модификаций

Вначале рассмотрим результаты сравнение рандомизированных и нерандомизированных инициализаций.

Комбинация случайной и неслучайной инициализации Как было описано выше, алгоритм заключается в построении выпуклой комбинации неслучайной инициализации со случайной.

От такой смеси ожидается быстрая сходимость к наименьшему значению перплексии, что подтверждается экспериментами.

В моих экспериментах по сравнению рандомизированных и нерандомизированных инициализаций использовался параметр смешивания $\alpha = 0.3$. В качестве случайных матриц использовались матрицы со столбцами, порожденными из равномерного распределения (нормированными так, чтобы сумма вероятностей в столбце равна 1).

Эксперименты проводились для двух инициализаций - кластеризации слов и алгоритма Ароры. В качестве коллекций использовались коллекции

nips и kos, при этом число тем было уменьшено до 70.

NIPS

Качество восстановления коллекции:

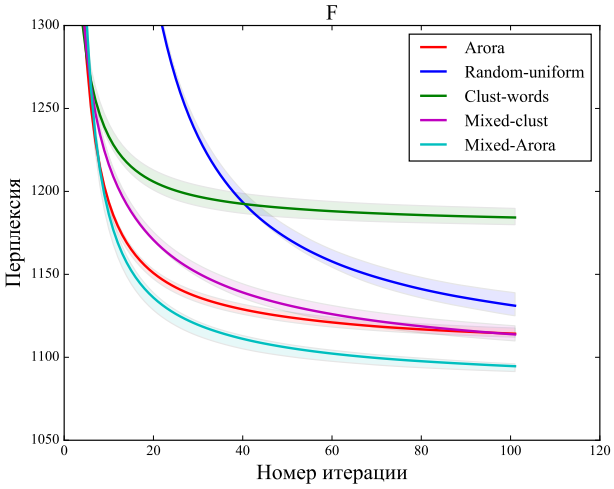

Рис. 25: Перплексия

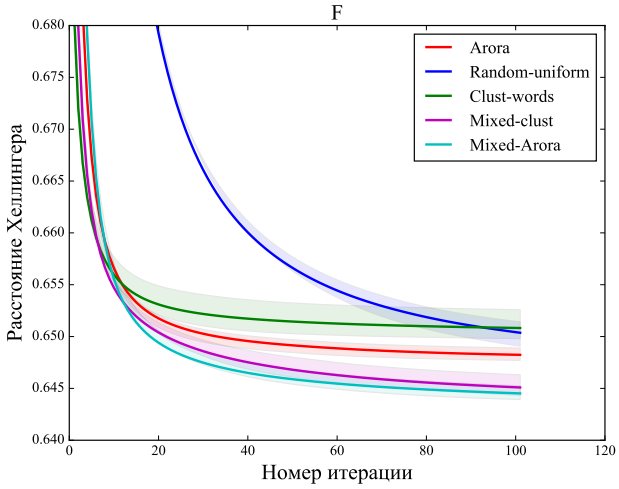

Рис. 26: Расстояние Хеллингера

Метрики качества матрицы слова-темы Φ		
	Mean PMI	Mean NHell
Arora	0.648	0.622
Random-uniform	0.632	0.662
Clust-words	0.829	0.798
Mixed-clust	0.800	0.702
Mixed-Arora	0.726	0.647

KOS

Качество восстановления коллекции:

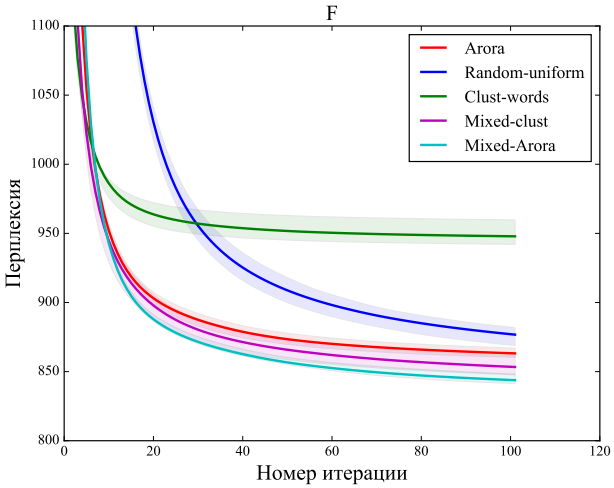

Рис. 27: Перплексия

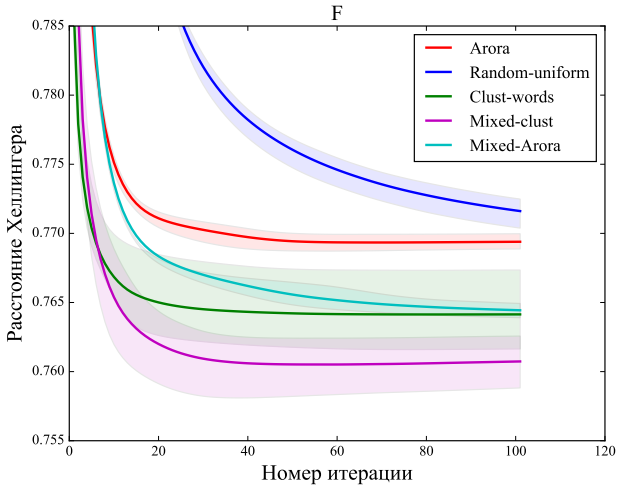

Рис. 28: Расстояние Хеллингера

Метрики качества матрицы слова-темы Φ		
	Mean PMI	Mean NHell
Arora	0.891	0.717
Random-uniform	0.709	0.757
Clust-words	0.842	0.835
Mixed-clust	0.960	0.794
Mixed-Arora	0.936	0.751

Заметно, насколько сильно такой алгоритм смеси улучшает как итоговое качество, так и скорость сходимости. По конечным значениям метрик наилучший алгоритм инициализации - рандомизированный алгоритм Ароры.

Встряхивание весов - jogging of weights В следующих экспериментах в качестве встряхивающих матриц Φ_{random}, Θ_{random} использовались разреженные матрицы со столбцами из симметричного распределения Дирихле с параметрами 0.05 и 0.1 для Φ_{random} и Θ_{random} соответственно.

Параметр смешивания α был равен 0.25^N после N-ой итерации EM-алгоритма, при этом встряхивание прекращалось после 10 итераций.

Используемые коллекции - nips и kos, число тем уменьшено до 70.

NIPS

Качество восстановления коллекции:

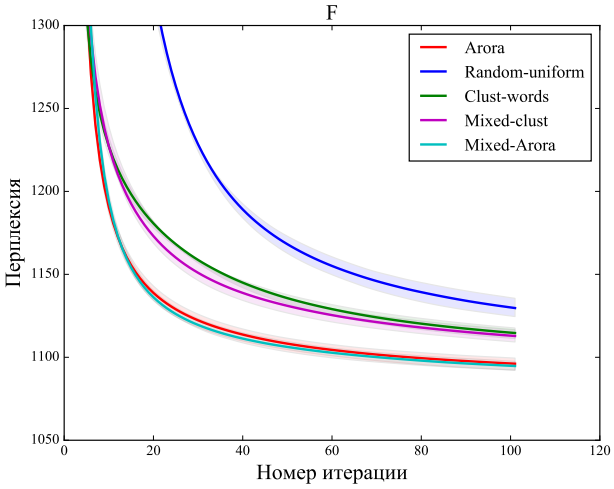

Рис. 29: Перплексия

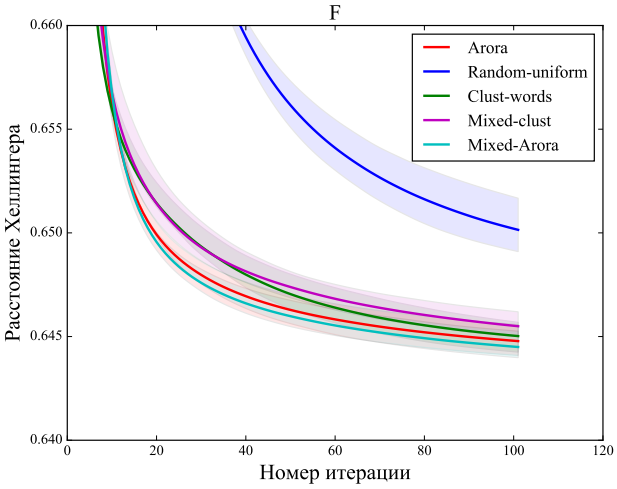

Рис. 30: Расстояние Хеллингера

Метрики качества матрицы слова-темы Φ		
	Mean PMI	Mean NHell
Arora	0.707	0.647
Random-uniform	0.633	0.665
Clust-words	0.807	0.710
Mixed-clust	0.784	0.695
Mixed-Arora	0.719	0.651

KOS

Качество восстановления коллекции:

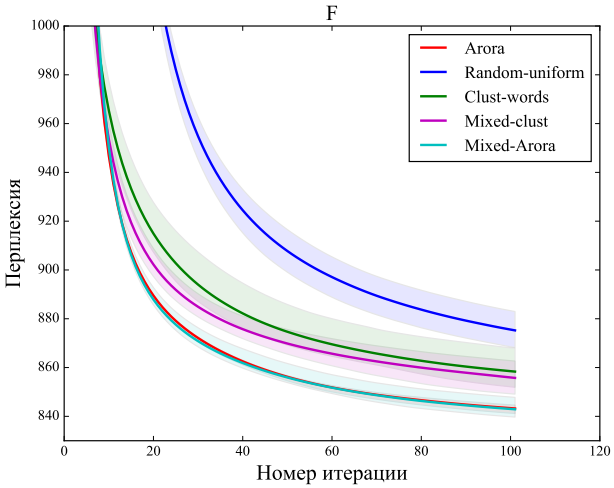

Рис. 31: Перплексия

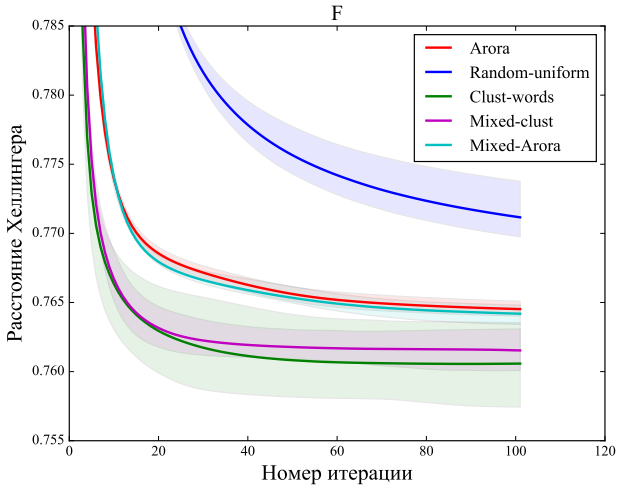

Рис. 32: Расстояние Хеллингера

Метрики качества матрицы слова-темы Φ		
	Mean PMI	Mean NHell
Arora	0.949	0.749
Random-uniform	0.677	0.760
Clust-words	0.928	0.800
Mixed-clust	0.948	0.794
Mixed-Arora	0.938	0.755

Несмотря на то, что для смеси алгоритмов инициализаций со случайной инициализацией и для встряхивания весов использовались разные случайные матрицы (почти равномерное распределение в первом случае и разреженные матрицы во втором), результаты рандомизированных и нерандомизированных алгоритмов инициализации почти не различаются.

Метод встряхивания весов позволяет достичь наилучшего качества. При этом в случае использования алгоритма Ароры хороший результат достигается уже при 30 итерациях EM-алгоритма, тогда как при случайной инициализации к этому моменту перплексия еще далека от своего минимума.

В терминах модели ARTM метод встряхивания весов можно описать как рандомизирующий регуляризатор, что позволяет легко реализовать такой

подход в существующих библиотеках.

Кластеризация документов Результаты сравнения кластеризации слов и кластеризации документов на коллекциях nips и kos приведены ниже. Количество используемых тем - 70.

NIPS:

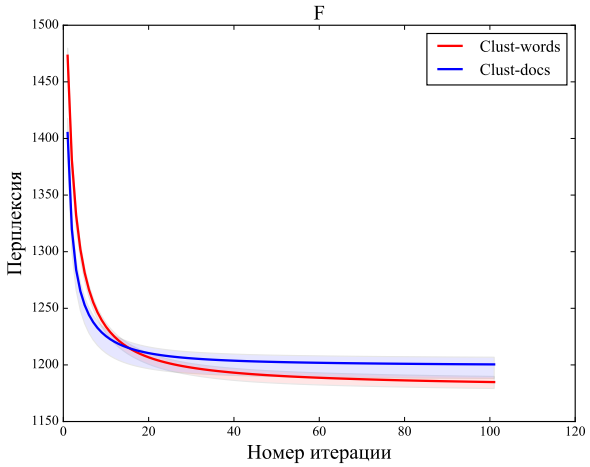

Рис. 33: Перплексия

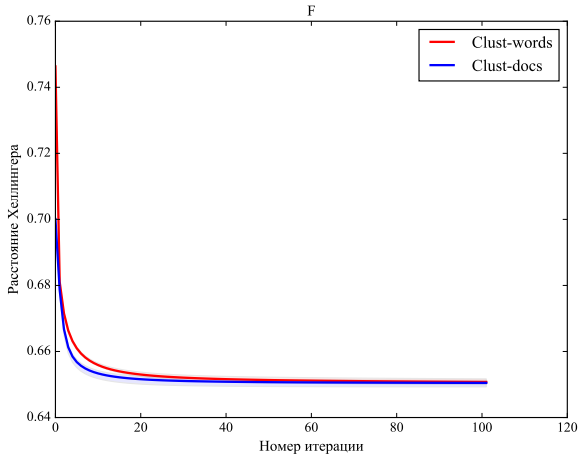

Рис. 34: Расстояние Хеллингера

KOS:

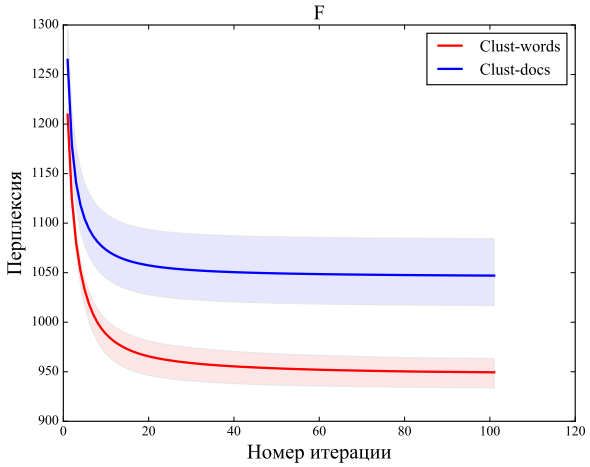

Рис. 35: Перплексия

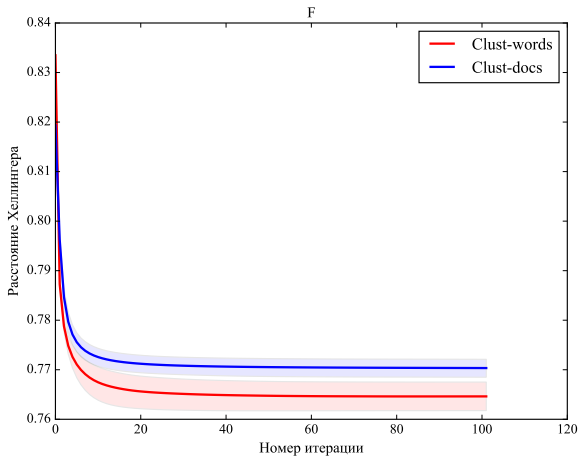

Рис. 36: Расстояние Хеллингера

По метрикам матрицы Φ кластеризация документов также уступает кластеризации слов.

Вывод из этих экспериментов - лучше использовать кластеризацию слов, что и было сделано при сравнении инициализаций.

Матрица Θ в алгоритме Ароры Алгоритм Ароры, использующий заполнение матрицы Θ значениями 1/T, на графиках обозначен как Arora-uniform. Результаты сравнения этой версии с той, что использовалась для сравнения с остальными инициализациями, приведены ниже.

Я использовал коллекции nips и kos при 70 темах:

NIPS:

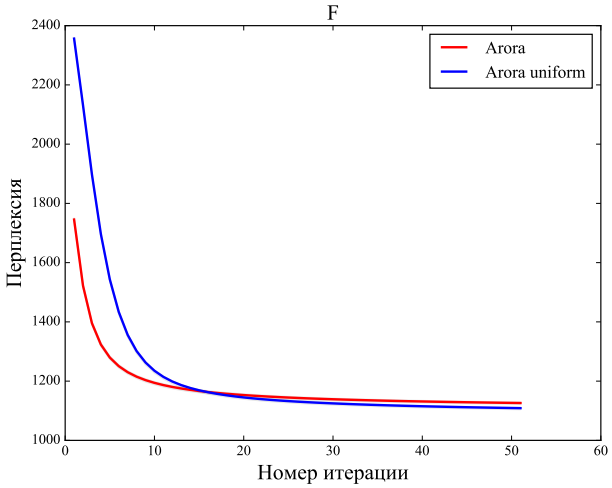

Рис. 37: Перплексия

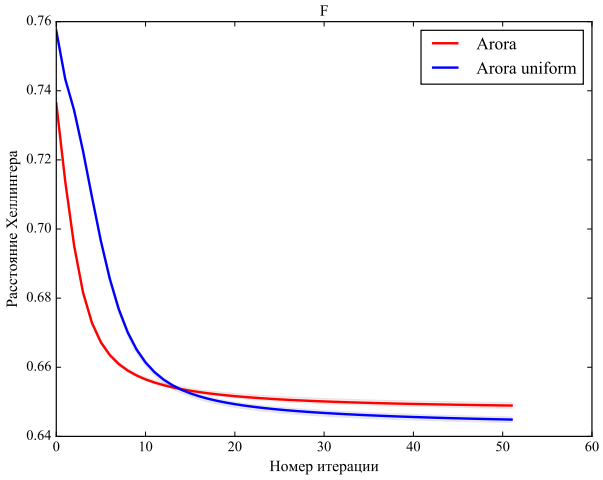

Рис. 38: Расстояние Хеллингера

KOS:

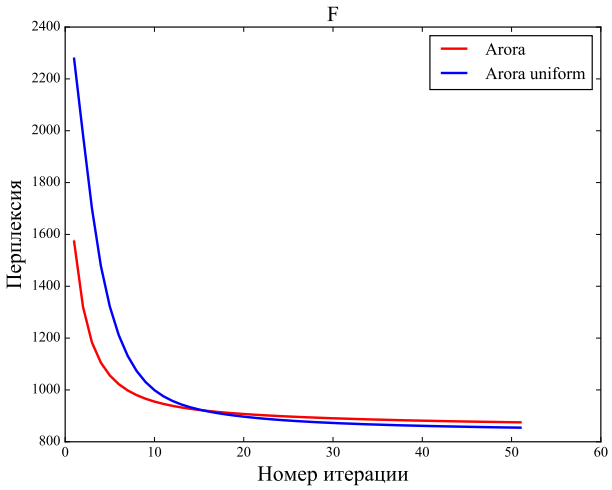

Рис. 39: Перплексия

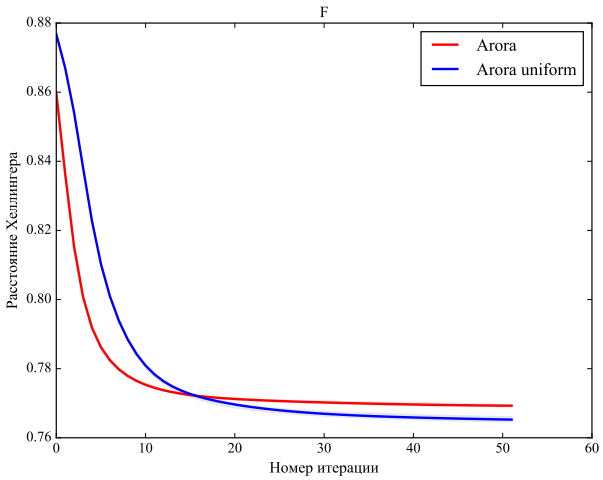

Рис. 40: Расстояние Хеллингера

Интересно посмотреть поведение этой модификации при использовании метода встряхивания весов:

NIPS:

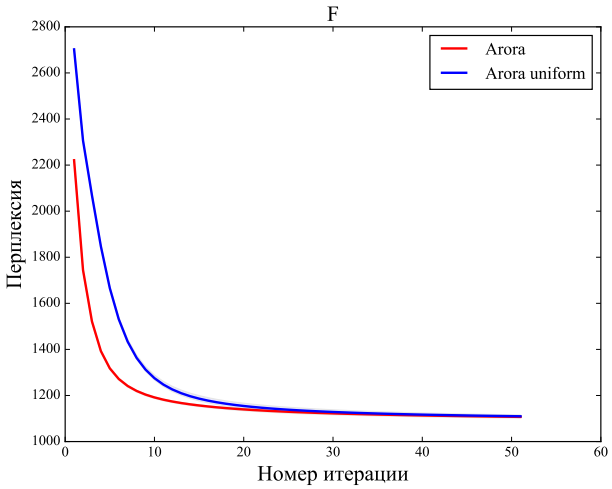

Рис. 41: Перплексия

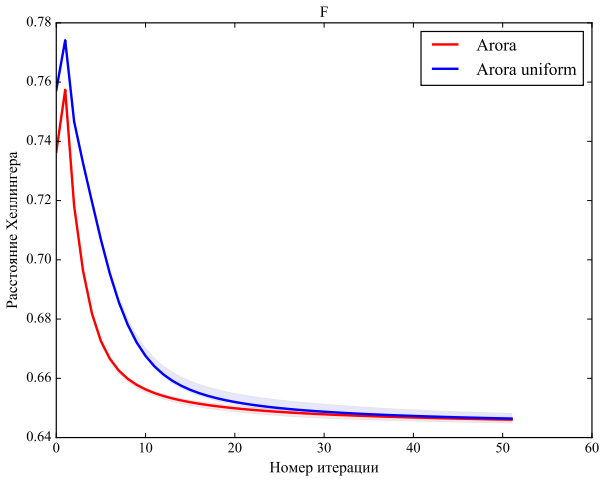

Рис. 42: Расстояние Хеллингера

KOS:

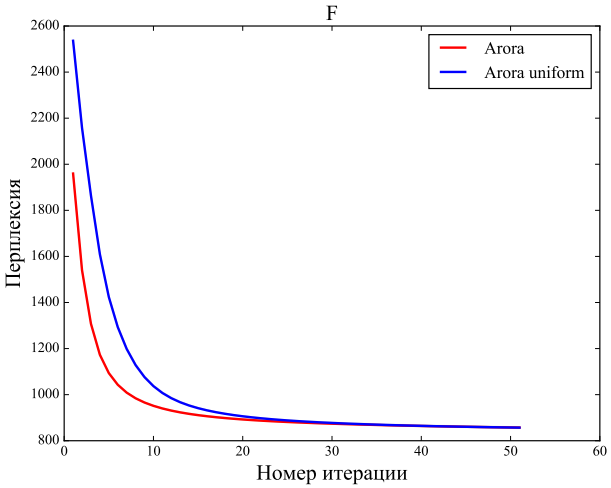

Рис. 43: Перплексия

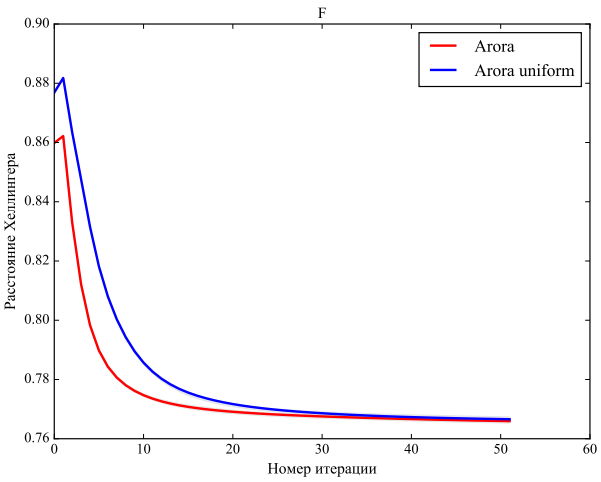

Рис. 44: Расстояние Хеллингера

По метрикам матрицы Φ такая модификация алгоритма в целом немного превосходит изначальный вариант заполнения Θ.

Как видно, гипотеза о том, что достаточно инициализировать только матрицу Φ подтвердилась. Самым большим плюсом является то, что такой метод гораздо эффективнее с точки зрения вычислительных затрат.

5 Заключение

5.1 Выводы и рекомендации

В работе была показана эффективность использования алгоритмов инициализации. Они позволяют избежать застревания в локальных оптимумах и значительно улучшить итоговое качество и скорость сходимости.

В качестве рекомендаций можно заключить следующее - при любой инициализации рекомендуется использовать метод встряхивания весов. Для достижения наилучших результатов в качестве начального приближения EM-алгоритма рекомендуется использовать линейную комбинацию алгоритма

Ароры при равномерном заполнении матрицы Θ со случайной равномерной инициализацией - параметр смешивания α стоит подбирать отдельно для каждой коллекции.

5.2 Результаты, выносимые на защиту

В работе были рассмотрены и исследованы несколько методов инициализации EM-алгоритма и их модификации.

Было произведено всестороннее исследование алгоритмов инциализации на синтетических и реальных коллекциях данных. По результатам исследований был сделан вывод, что наилучший алгоритм инициализации - алгоритм Ароры. Он позволяет за малое число итераций сойтись к наилучшему значению перплексии. Затем был предложен модифицированный вариант алгоритма Ароры с равномерным заполнением матрицы Θ, который показывает такие же результаты, но является более эффективным с точки зрения вычислений.

В работе также предложен метод улучшения алгоритмов инициализаций, основанный на смешивании исходных матриц со случайными матрицами. На основе этой идеи был предложен и исследован метод встряхивания весов для EM-алгоритма. Этот метод позволяют сильно улучшить сходимость EM-алгоритма.

По итогам этих исследований сформулированы рекомендации по практическому использованию алгоритмов инициализации для EM-алгоритма.

Список литературы

[1] *Arora S. et al.* A practical algorithm for topic modeling with provable guarantees // Proceedings of the 30th International Conference on Machine Learning (ICML-13). — 2013. — Pp. 280–288.

[2] *Blei D. M., Ng A. Y., Jordan M. I.* Latent dirichlet allocation // *the Journal of machine Learning research.* — 2003. — Vol. 3. — Pp. 993–1022.

[3] *Boutsidis.* Svd based initialization: A head start for nonnegative matrix factorization // *Pattern Recognition.* — 2008. — Vol. 41, no. 4. — Pp. 1350 – 1362.

[4] *Ding W., Rohban M. H., Ishwar P., Saligrama V.* Efficient distributed topic modeling with provable guarantees. // AISTATS. — 2014. — Pp. 167–175.

[5] *Hofmann T.* Probabilistic latent semantic analysis // Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence / Morgan Kaufmann Publishers Inc. — 1999. — Pp. 289–296.

[6] *Hofmann T.* Probabilistic latent semantic indexing // Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval / ACM. — 1999. — Pp. 50–57.

[7] *Langville A., Meyer C., Albright R., Cox J., Duling D.* Algorithms, initializations, and convergence for the nonnegative matrix factorization // *ArXiv e-prints.* — 2014.

[8] *Lovato P., Bicego M., Murino V., Perina A.* Robust initialization for learning latent dirichlet allocation // Similarity-Based Pattern Recognition - Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015, Proceedings / Ed. by A. Feragen, M. Pelillo, M. Loog. — Vol. 9370 of *Lecture Notes in Computer Science.* — Springer,

2015. — Pp. 117–132.

[9] *Sokolov E., Bogolubsky L.* Topic models regularization and initialization for regression problems // Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications. — New York, NY, USA: ACM, 2015. — Pp. 21–27.

[10] *Vorontsov K.* Additive regularization for topic models of text collections // Doklady Mathematics / Pleiades Publishing. — Vol. 89. — 2014. — Pp. 301–304.