КІБЕРНЕТИКА

А.В. АНИСИМОВ, А.А. МАРЧЕНКО, Э.М. НАСИРОВ

УДК 681.3

БЛОЧНО-ДИАГОНАЛЬНЫЙ ПОДХОД К НЕОТРИЦАТЕЛЬНОЙ ФАКТОРИЗАЦИИ РАЗРЕЖЕННЫХ ЛИНГВИСТИЧЕСКИХ МАТРИЦ И ТЕНЗОРОВ СВЕРХБОЛЬШОЙ РАЗМЕРНОСТИ С ИСПОЛЬЗОВАНИЕМ ЛАТЕНТНОГО РАСПРЕДЕЛЕНИЯ ДИРИХЛЕ

Аннотация. Описаны алгоритмы неотрицательной факторизации разреженных матриц и тензоров. Рассмотрено использование латентного распределения Дирихле для приведения матриц и тензоров к блочно-диагональной форме для параллелизации вычислений и ускорения неотрицательной факторизации лингвистических матриц и тензоров сверхбольшой размерности. Предложенная модель позволяет дополнять модели новыми данными без необходимости выполнять неотрицательную факторизацию всего сверхбольшого тензора заново.

Ключевые слова: искусственный интеллект, компьютерная лингвистика, параллельные вычисления.

ВВЕДЕНИЕ

В настоящее время неотрицательная факторизация матриц и тензоров — очень популярная технология в искусственном интеллекте вообще и в компьютерной лингвистике в частности. Используя неотрицательную факторизацию в рамках парадигмы латентно-семантического анализа, компьютерные лингвисты применяют данный подход для решения прикладных задач классификации, кластеризации текстов и терминов [1, 2], построения мер семантической близости [3], автоматического выделения из корпусов текстов таких лингвистических структур и отношений, как предпочтения сочетаемости в предложениях (Selectional Preferences) [4] и субкатегориальные фреймы глаголов (Verb Sub-Categorization Frames [5]), которые включают данные о семантических и синтаксических свойствах связей между глаголами и их аргументами — существительными в предложениях, и др.

Неотрицательная факторизация N-мерного тензора при ранге разложения k формирует N двумерных матриц, состоящих из k векторов-столбцов, соответствующих отображению каждого измерения тензора на k факторов-измерений латентного семантического пространства. Неотрицательная факторизация предоставляет уникальное средство для моделирования и выявления взаимосвязей лингвистических переменных в массиве N-мерных данных. Неотрицательную факторизацию лингвистических тензоров, полученных при частотном анализе корпусов текстов, можно использовать в качестве основы семантико-синтаксической модели естественного языка [6]. В таких тензорах каждое измерение соответствует некоторому фиксированному члену предложения — подлежащему, сказуемому, дополнению, определению, обстоятельству. Точность такой модели зависит от объема обработанных данных, а значит, и от размера обрабатываемого тензора.

Сложность выполнения неотрицательной факторизации матриц и тензоров сверхбольшой размерности заключается в необходимости сохранять большой объем данных на каждой итерации алгоритма. Так, например, в проведенных авторами экспериментах для разреженной матрицы размера $2\,437\,234 \times 4\,475\,180$, содержащей 156 236 043 ненулевых элементов, при ранге факторизации $k = 300$ необходимый объем результирующих матриц текущей и предыдущей итераций превышает 16 Гб.

В последнее время разработано множество параллельных моделей для неотрицательной факторизации матриц [7–10]. Однако ни одна из них не является приемлемым решением для данной задачи, а некоторые не соответствуют требованиям к размерам матриц [7–9].

Tensor Toolbox [7] предоставляет классы и функции для работы с плотными, разреженными и структурированными тензорами с помощью функций MATLAB. Однако встроенные функции декомпозиции тензоров хранят все данные в оперативной памяти и вычисления не распределяются. Таким образом, Tensor Toolbox нельзя использовать для неотрицательной факторизации сверхбольших тензоров.

В работе [8] предложен параллельный подход к неотрицательной факторизации матриц для кластеризации документов. Для параллелизации алгоритма факторизации используется интерфейс OpenMP и хранение промежуточных данных на диске. Такой подход занимает много времени на необходимое многократное чтение и запись данных, требует общей памяти для выполнения вычислительных процессов, следовательно, не используется для неотрицательной факторизации сверхбольших тензоров.

Применение GPU для ускорения неотрицательной факторизации матриц предложено в [9]. Такой подход действительно позволяет увеличить производительность вычисления, но, как и предыдущие, имеет ограничения в использовании оперативной памяти и памяти графического адаптера, а следовательно, требует постоянных процессов чтения/записи данных с накопителя, а также перемещения данных в памяти графического адаптера.

В модели, представленной в [10], выполняется разделение данных, распределение вычислений и параллелизм с использованием MapReduce кластеров. В этой модели неотрицательную факторизацию сверхбольших разреженных матриц можно выполнить за приемлемое время, но требуются чрезмерно большие вычислительные ресурсы, что не всегда доступно.

В настоящей работе описано построение модели параллелизации вычисления неотрицательной факторизации разреженных матриц сверхбольшой размерности с помощью их приведения к блочно-диагональному виду с использованием латентного распределения Дирихле. Такой подход особенно актуален для его применения в больших лингвистических системах, не ограниченных использованием только в узких предметных областях.

Задачи неотрицательной факторизации разреженных матриц и тензоров сверхбольшой размерности возникли в процессе разработки систем определения степени семантической близости–связности по технологии латентно-семантического анализа [11].

АЛГОРИТМ НЕОТРИЦАТЕЛЬНОЙ МАТРИЧНОЙ ФАКТОРИЗАЦИИ

Неотрицательная матричная факторизация известна давно. Исследование неотрицательной матричной факторизации проводилось группой ученых из Финляндии в середине 1990-х годов и называлось позитивной матричной факторизацией [12]. Широко известной неотрицательная матричная факторизация стала после того, как Ли и Сунг [13] исследовали свойства алгоритма и опубликовали некоторые простые и полезные алгоритмы для двух типов факторизации.

В общей постановке задача неотрицательной факторизации матриц является NP-полной [14].

Неотрицательная матричная факторизация определяется как следующая задача. Дано (V, k), где V — числовая матрица размера $m \times n$ с неотрицательными элементами, а k — целое число, такое что $1 \le k \le \min(m, n)$. Результатом является пара матриц (W, H), $W \in R^{m \times k}$, $H \in R^{k \times n}$, таких что W и H имеют неотрицательные элементы, для которых выполняется соотношение $V \approx WH$.

В качестве функции оценки сходимости можно использовать функцию измерения расстояния между двумя неотрицательными матрицами: A и B. Одной из таких мер является квадрат евклидовой метрики

$$\mu = || A - B ||^2 = \sum_{ij} (A_{ij} - B_{ij})^2.$$

Такая целевая функция ограничена снизу. Нижняя граница 0 достигается тогда и только тогда, когда $A = B$.

Следовательно, при использовании евклидовой метрики факторизация матрицы заключается в минимизации $||V - WH||^2$ при условии неотрицательности W и H.

Такая целевая функция невозрастающая, а матрицы W и H становятся постоянными только в случае достижения стационарной точки целевой функции при следующих правилах:

$$H_{ij} \leftarrow H_{ij} \frac{(W^{\mathrm{T}} V)_{ij}}{(W^{\mathrm{T}} WH)_{ij}},$$

$$W_{ij} \leftarrow W_{ij} \frac{(VH^{\mathrm{T}})_{ij}}{(WHH^{\mathrm{T}})_{ij}}.$$

Выполнение итераций алгоритма Ли и Сунга продолжается до тех пор, пока не будет достигнута стационарная точка или не будет выполнено максимальное количество итераций [13].

ПРИМЕНЕНИЕ МОДЕЛИ К ФАКТОРИЗАЦИИ ТЕНЗОРОВ

Лингвистический тензор — это N-мерная матрица, в которой сохраняются частота употребления словосочетаний из N слов в текстовых корпусах. Главная идея факторизации тензора — минимизация суммы квадратов разностей между оригинальным тензором и факторизованной моделью тензора.

Для того чтобы вычислить неотрицательные матрицы-компоненты $\{A, B, C\}$ для трехмерного тензора Y, применяется ограниченный оптимизационный подход для минимизации приемлемой функции оценки. Обычно минимизируют следующую функцию:

$$D_F(\underline{Y} || [\![A, B, C]\!]) = || \underline{Y} - [\![A, B, C]\!] ||_F^2 + \alpha_A || A ||_F^2 + \alpha_B || B ||_F^2 + \alpha_C || C ||_F^2,$$

где $\alpha_A, \alpha_B, \alpha_C$ — неотрицательные регуляционные параметры, $[\![A, B, C]\!]$ — тензорное произведение матриц A, B, C.

Существует три различных подхода к решению такой оптимизационной задачи. Первый подход заключается в использовании векторной формы функции оценки: $J(X) = \mathrm{vec}(Y - [A, B, C]) = 0$. Для решения применяется метод наименьших квадратов. Этот метод для факторизации тензоров впервые использовал Паатеро [12]. Якобиан такой функции может иметь размер $ITQJ \times (I + T + Q)$, а следовательно, такой подход требует значительных вычислительных затрат.

Во втором подходе Акар, Колда и Дунлави предложили искусственно оптимизировать функцию оценки, используя технику нелинейной связной градиентной оптимизации [1].

Третьим, и самым распространенным подходом, является использование техники Alternating Least Squares (ALS). В этом подходе подсчитывается градиент функции оценки для каждой матрицы отдельно:

$$\nabla_A D_F = -Y_{(1)}(C \cdot B) + A[(C^{\mathrm{T}}C) * (B^{\mathrm{T}}B) + \alpha_A I],$$

$$\nabla_B D_F = -Y_{(2)}(C \cdot A) + B[(C^{\mathrm{T}}C) * (A^{\mathrm{T}}A) + \alpha_B I],$$

$$\nabla_C D_F = -Y_{(3)}(B \cdot A) + A[(B^{\mathrm{T}}B) * (A^{\mathrm{T}}A) + \alpha_C I].$$

Приравнивая каждый компонент градиента к нулю и накладывая условие неотрицательности, получаем эффективные и простые правила итеративного обновления матриц:

$$A \leftarrow [Y_{(1)}(C \cdot B) + [(C^{\mathrm{T}}C) * (B^{\mathrm{T}}B) + \alpha_A I]^{-1}]_+,$$

$$B \leftarrow [Y_{(2)}(C \cdot A) + [(C^{\mathrm{T}}C) * (A^{\mathrm{T}}A) + \alpha_B I]^{-1}]_+,$$

$$C \leftarrow [Y_{(3)}(B \cdot A) + [(B^{\mathrm{T}}B) * (A^{\mathrm{T}}A) + \alpha_C I]^{-1}]_+.$$

Основными преимуществами такого подхода являются высокая скорость сходимости и возможность распределения вычислений для задач большой размерности.

БЛОЧНО-ДИАГОНАЛЬНЫЙ ПОДХОД К ФАКТОРИЗАЦИИ МАТРИЦ И ТЕНЗОРОВ

Суть предлагаемого в данной работе метода заключается в переходе от необходимости факторизовать сверхбольшой разреженный лингвистический тензор или матрицу к неотрицательной факторизации набора лингвистических тензоров и матриц значительно меньшего размера.

Лингвистические матрицы и тензоры являются высокой степени разреженными. Это позволяет привести их к блочно-диагональной форме с помощью перестановок слоев в тензорах или строк и столбцов в матрице, изменяя базис факторизации. В случае, если слово принадлежит нескольким блокам, можно создать несколько соответствующих ему копий. Графическое представление тензора блочно-диагонального вида приведено на рис. 1.

Размеры блоков лучше выбирать такими, чтобы можно было полностью загрузить в память для исключения необходимости постоянного чтения и записи на жесткий диск. В результате получится N блоков.

Приведем преимущества такого подхода.

1. Не требуются сетевые операции (передача данных между вычислительными узлами) на каждой итерации, необходимо лишь начальное распределение блоков матрицы или тензора между вычислительными узлами.

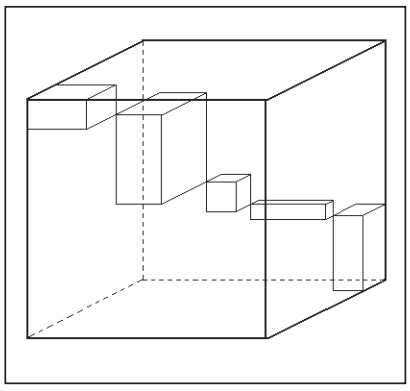

Рис. 1.

2. Как показывают эксперименты, сходимость факторизации матриц и тензоров происходит быстрее при уменьшении их размера. На рис. 2 и 3 приведено время, необходимое для факторизации соответственно квадратных матриц и кубических тензоров различных размеров в один поток без использования GPU. На рис. 4 показано необходимое количество итераций для неотрицательной факторизации квадратных матриц различных размеров.

3. Уменьшается значение k, а значит, и размер результирующих матриц и объем памяти, необходимый для хранения данных во время итераций.

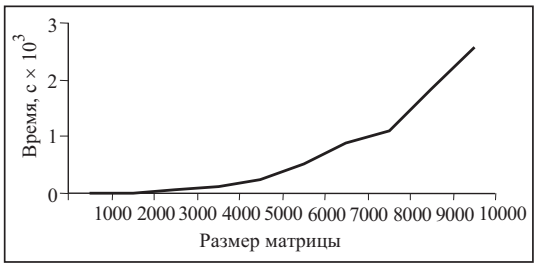

Рис. 2

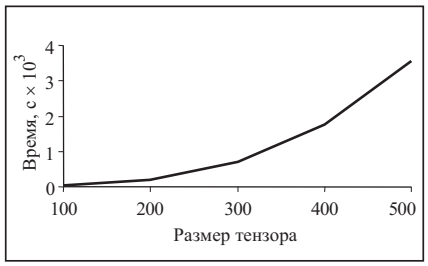

Рис. 3

4. Ускоряется вычисление функции оценки. Вычисление произведения $W \cdot H$ происходит быстрее, так как уменьшилось значение k. Количество элементов для подсчета функции оценки уменьшилось в N раз (для матрицы), также уменьшилось количество необходимых проверок, поскольку уменьшилось общее количество итераций.

5. Гарантированно эффективно поддерживается масштабируемость модели, т.е. для ее расширения и наполнения новыми данными из дополнительных текстовых корпусов не нужно выполнять неотрицательную факторизацию всего сверхбольшого тензора заново с самого начала. Вычисления проводятся лишь на отдельных блоках лингвистического тензора с последующей модификацией только соответствующих фрагментов векторов модели.

6. Восстановление значений элементов матриц и тензоров после факторизации происходит быстрее. Все ненулевые значения распределены только по диагональным блокам, поэтому все значения, которые находятся вне них, не отличны от 0. Для получения этих значений достаточно восстановления значений только в блоках вхождения слов. В случае сведения к блочно-диагональной матрице в результате факторизации отдельных блоков для каждого слова будут получены одна или несколько копий и идентификаторы блоков, в котором это слово встречается. При восстановлении, например матрицы вхождения слов в статьи, если слово и статья находятся в разных блоках, результатом будет 0, иначе — произведение соответствующих строки и столбца.

Недостатком предложенного метода является то, что создание копий слов при группировке необходимых ненулевых ячеек матрицы или тензора в блоки происходит случайным образом и может привести к созданию многих копий одного слова.

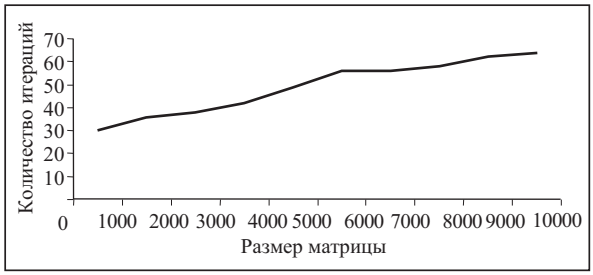

Рис. 4

ИСПОЛЬЗОВАНИЕ ЛАТЕНТНОГО РАСПРЕДЕЛЕНИЯ ДИРИХЛЕ ДЛЯ ПРИВЕДЕНИЯ МАТРИЦ И ТЕНЗОРОВ К БЛОЧНО-ДИАГОНАЛЬНОЙ ФОРМЕ

Латентное распределение Дирихле (LDA) [15] — генеративную вероятностную модель текстового корпуса, можно использовать для построения блочно-диагональной структуры внутри единого большого лингвистического тензора методом формирования тематических диагональных блоков. LDA — трехуровневая иерархическая байесовская модель, в которой каждый элемент коллекции моделируется как конечная смесь над основным набором тем. Каждая тема, в свою очередь, моделируется как бесконечная смесь над основным набором тематических вероятностей.

Основная идея заключается в том, что документы представлены как случайные смеси над латентными темами, где каждая тема характеризуется распределе-

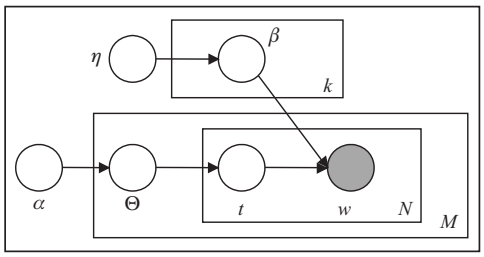

Рис. 5

нием по словам. В LDA каждый документ можно рассматривать как набор различных тематик.

Метод LDA состоит из следующих шагов для каждого документа d в корпусе D.

1. Для каждого слова из каждого документа присвоить случайно одну тему t из K возможных.

2. Для каждого слова из каждого документа подсчитать $p(t \mid d)$ — процент слов в документе d, принадлежащих теме t, и $p(w \mid t)$ — процент слова w во всех документах из темы t, а также присвоить слову w новую тему t с вероятностью $p(t \mid d) \cdot p(w \mid t)$.

3. Повторить п. 2 необходимое количество итераций.

Метод LDA основан на вероятностной модели

$$p(d \mid w) = \sum_{t \in T} p(d) \cdot p(w \mid t) \cdot p(t \mid d), \ t \in T,$$

где d — документ; t — тема; w — слово; T — множество тем; $p(d)$ — распределение на множестве документов; $p(w \mid t)$ — условное распределение слова w в теме t; $p(t \mid d)$ — условное распределение темы t в документе d.

Графическое представление метода LDA показано на рис. 5, где M — коллекция документов, N — длина документа в словах, Θ — распределение тем в документе, α — априорное распределение Дирихле на параметры Θ.

Предлагается, использовав определение вероятности принадлежности слов к тематикам методом LDA, привести тензор к блочно-диагональному виду из T блоков (количества тем). Для этого необходимо после выполнения метода LDA и получения вероятностей принадлежности слов определенным тематикам выполнить следующие шаги для трехмерного тензора.

1. Изначально для каждого слова выбрать тему t_{wi} с максимальной полученной вероятностью.

2. Значения тензора на пересечении трех слов одной тематики сразу переместить в соответствующий блок.

3. Значения на пересечении только двух слов, принадлежащих одной тематике t_i, переместить в блок этой тематики, а для третьего слова также создать копию в теме t_i.

4. Для каждого оставшегося значения тензора выбрать тему среди тематик образующих его слов с наибольшей вероятностью. Для двух других слов, образующих это значение, создать копию в выбранной теме, если она не создавалась на предыдущем шаге.

После того, как все значения тензора будут разделены между блоками, необходимо провести переиндексацию слов и смещение. В результате в левом верхнем ближнем блоке тензора образуется тензор-подпространство i-й тематики. Все его значения переносятся в отдельный тензор Y_i, при этом все соответствующие ненулевые значения в тензоре Y в диапазоне $[1 \dots n_1] \times [1 \dots m_1] \times [1 \dots k_1]$ обнуляются. После переноса тензор Y_i тематики t_i можно факторизовать отдельно.

При таком подходе копии слов создаются только в случае необходимости и в количестве, которое не превысит количество тем, а следовательно, и блоков.

Таким образом, система может не сразу обрабатывать миллионы текстов различных тематик и составлять тензор сверхбольшой размерности, а обрабатывать и анализировать весь входящий текстовый корпус по тематическим разделам. В результате образуется набор независимых лингвистических тензоров приемлемой размерности для каждой темы отдельно.

Для сравнения быстродействия предложенного подхода собран трехмерный тензор субъект–объект–предикат по статьям англоязычной Википедии из трех тематик с размерами $427 \times 363 \times 528$ слов. Использовав предложенный подход, тензор привели к блочно-диагональному виду с тремя блоками, соответствующими трем тематикам. Получены три тензора с размерами $115 \times 92 \times 171$, $126 \times 111 \times 149$ и $186 \times 160 \times 208$.

Для сравнения времени, необходимого для факторизации начального и блочно-диагонального тензоров, тестирование проводилось в один поток без использования параллелизации и GPU. Для факторизации начального тензора понадобилось 2.4489837 с; для факторизации первого блока — 0.5734838 с, второго блока — 0.6894228 с и третьего блока — 0.4832307 с. Суммарное время для факторизации тензора, приведенного к блочно-диагональной форме, составило 1.7461373 с. Таким образом, даже на небольшом тензоре и разделении на три диагональных блока ожидаемо удалось уменьшить время факторизации на 28.7 %.

ЗАКЛЮЧЕНИЕ

В работе описан алгоритм неотрицательной факторизации матриц и алгоритм неотрицательной факторизации тензоров. Предложен блочно-диагональный под-ход к неотрицательной факторизации матриц и тензоров. Приведены преимущества этого подхода в быстродействии и уменьшении необходимых сетевых операций.

Предложен новый метод блочной диагонализации и неотрицательной факторизации лингвистических тензоров, заключающийся в предварительном приведении тензора к блочно-диагональному виду с помощью метода LDA, чтобы свести неотрицательную факторизацию тензора сверхбольшого размера к выполнению вычислений неотрицательной факторизации тензорных диагональных блоков — набора тензоров гораздо меньшего размера, который требует значительно меньшего времени и вычислительных ресурсов. Результаты использования блочно-диагонального подхода на тестовом тензоре подтвердили его эффективность и показали значительное преимущество нового алгоритма во времени выполнения. При увеличении размера тензора и количества блоков это преимущество будет увеличиваться.

СПИСОК ЛИТЕРАТУРЫ

1. Xu W., Liu X., Gong Y. Document-clustering based on n-negative matrix factorization. *Proc. of SIGIR'2003.* 2003. P. 267–273.
2. Shahnaz F., Berry M.W., Paul Pauca V., Plemmons R.J. Document clustering using nonnegative matrix factorization. *Information Processing and Management.* 2006. Vol. 42. P. 649–660.
3. Anisimov A., Marchenko O., Nikolenko A., Porkhun E., Taranukha V. Ukrainian WordNet: Creation and filling. *Flexible Query Answering Systems*. *FQAS 2013*. Larsen H.L., Marnin-Bautista M.J., Vila M.A., Andreasen T., Christiasen H. (Eds.). *Lecture Notes in Computer Science*. 2013. Vol. 8132. P. 649–660.
4. Van De Cruys T. A non-negative tensor factorization model for selectional preference induction. *Journal of Natural Language Engineering*. 2010. Vol. 16, N 4. P. 417–437.
5. Van De Cruys T., Rimell L., Poibeau T., Korhonen A. Multi-way tensor factorization for unsupervised lexical acquisition. *Proc. of COLING-2012*. 2012. P. 2703–2720.
6. Марченко А.А. Метод автоматического построения онтологических баз знаний. I. Разработка семантико-синтаксической модели естественного языка. *Кибернетика и системный анализ.* 2016. Т. 52, № 1. С. 23–33.
7. Bader B.W., Kolda T.G. Matlab tensor toolbox version 2.5. URL: http://www.sandia.gov/~tgkolda/TensorToolbox/.
8. Kanjani K. Parallel non negative matrix factorization for document clustering. CPSC-659 (Parallel and Distributed Numerical Algorithms) course. Texas A & M University. Tech. Rep., 2007. URL: https://pdfs.semanticscholar.org/66ad/868f7fe55db5b64f963533a6cb8e9a245257.pdf.

9. Kysenko V., Rupp K., Marchenko O., Selberherr S., Anisimov A. GPU accelerated non-negative matrix factorization for text mining. *Natural Language Processing and Information Systems. Lecture Notes in Computer Science.* 2012. Vol. 7337. P. 158–163.
10. C. Liu, H.-c. Yang, J. Fan, L.-W. He, and Y.-M. Wang. Distributed non-negative matrix factorization for web-scale dyadic data analysis on mapreduce. *Proc. of the 19th Intern. Conf. on World Wide Web, WWW'10*. 2010. P. 681–690.
11. Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., Harshman R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*. 1990. Vol. 41, N 6. P. 391–407.
12. Paatero P., Tapper U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*. 1994. Vol. 5, N 2. P. 111–126.
13. Lee D.D., Seung H. S. Algorithms for non-negative matrix factorization. *NIPS Proc*. 2000. P. 556–562.
14. Vavasis S.A. On the complexity of non-negative matrix factorization. *SIAM J. Optim.* 2009. Vol. 20. P. 1364–1377.
15. Blei D.M., Ng A.Y., Jordan M.I. Latent Dirichlet allocation. *Journal of Machine Learning Research.* 2003. Vol. 3. P. 993–1022.

Надійшла до редакції 19.07.2018

А.В. Анісімов, О.О. Марченко, Е.М. Насіров

БЛОЧНО-ДІАГОНАЛЬНИЙ ПІДХІД ДО НЕВІД'ЄМНОЇ ФАКТОРИЗАЦІЇ РОЗРІДЖЕНИХ ЛІНГВІСТИЧНИХ МАТРИЦЬ І ТЕНЗОРІВ НАДВЕЛИКОЇ РОЗМІРНОСТІ З ВИКОРИСТАННЯМ ЛАТЕНТНОГО РОЗПОДІЛУ ДІРІХЛЕ

Анотація. Описано алгоритми невід'ємної факторизації розріджених матриць і тензорів. Розглянуто використання латентного розподілу Діріхле для приведення матриць і тензорів до блочно-діагональної форми для паралелізації обчислень і прискорення невід'ємної факторизации лінгвістичних матриць і тензорів надвеликої розмірності. За допомогою запропонованої моделі можна доповнювати моделі новими даними без необхідності знову виконувати невід'ємну факторизацію всього надвеликого тензора.

Ключові слова: штучний інтелект, комп'ютерна лінгвістика, паралельні обчислення.

A.V. Anisimov, O.O. Marchenko, E.M. Nasirov

BLOCK DIAGONAL APPROACH TO THE NON-NEGATIVE SPARSE LINGUISTIC EXTRA LARGE MATRICES AND TENSORS FACTORIZATION USING THE LATENT DIRICHLET DISTRIBUTION

Abstract. In this paper, algorithms for the non-negative factorization of sparse matrices and tensors, a popular technology in artificial intelligence in general and in computer linguistics in particular, are described. It is proposed to use the latent Dirichlet distribution to reduce matrices and tensors to block-diagonal form for parallelizing computations and accelerating the non-negative factorization of linguistic matrices and tensors of extremely large dimension. The proposed model also allows the models to be supplemented with new data without having to perform non-negative factorization of the entire super-large tensor anew from the very beginning.

Keywords: artificial intelligence, computational linguistics, parallel computations.

Анисимов Анатолий Васильевич,
чл.-кор. НАН Украины, доктор физ.-мат. наук, профессор, декан факультета Киевского национального университета имени Тараса Шевченко, e-mail: ava@unicyb.kiev.ua.

Марченко Александр Александрович,
доктор физ.-мат. наук, профессор кафедры Киевского национального университета имени Тараса Шевченко, e-mail: omarchenko@univ.kiev.ua.

Насиров Эмил Мехдиевич,
соискатель, Киевский национальный университет имени Тараса Шевченко,
e-mail: enasirov@gmail.com.