Цыплов Алексей Михайлович
Аспирант 2го курса
Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
e-mail: tsyplov80@mail.ru amtsyplov@itmo.ru

ЛОГНОРМАЛЬНАЯ РЕГУЛЯРИЗАЦИЯ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ ДЛЯ УСТРАНЕНИЯ POPULARITY BIAS

Аннотация. В рамках исследования рассматривается новый подход к регуляризации обучения нейросетевых рекомендательных систем, основанный на предположении о том, что популярность объектов в рекомендациях хорошо описывается логнормальным распределением. Предложена модель распространения популярности объектов в социальном графе, объясняющая логнормальный характер распределения популярности. Разработанный метод регуляризации позволяет оценивать будущую популярность объектов в ходе обучения и не допускать большого отличия от их популярности в исходной выборке. Для проверки нового метода были проведены численные эксперименты с моделью матричной факторизации на наборе данных MovieLens-1M, получено эмпирическое подтверждение его эффективности. Предложенный подход может быть применён для улучшения любых нейросетевых рекомендательных систем, обучение которых проводится по методологии Bayesian Pairwise Ranking.

Ключевые слова: Рекомендательные системы, матричная факторизация, регуляризация, popularity bias

Abstract. In this study, a novel approach to the regularization of neural network-based recommender systems is explored. The approach is based on the assumption that item popularity in recommendations is well characterized by a lognormal distribution. A model of item popularity propagation in a social graph is proposed, providing an explanation for the lognormal nature of the popularity distribution. The developed regularization method enables the estimation of future item popularity during training, ensuring that it does not significantly deviate from the original popularity observed in the dataset. In order to evaluate the proposed method, numerical experiments were conducted using a matrix factorization model on the MovieLens-1M dataset, yielding empirical evidence of its effectiveness. The proposed approach can be applied to enhance any neural network-based recommender system trained via the Bayesian Pairwise Ranking methodology.

Key words: Recommender system, matrix factorization, regularization, popularity bias

Введение

В современном мире в задачах поиска информации огромную роль играют рекомендательные системы. Их используют в самых различных сферах, от онлайн-торговых площадок до видео-сервисов и социальных сетей. Рекомендательные технологии используются для оценки интересов пользователей и предоставления персональных рекомендаций, что повышает удовлетворённость пользователей и повышает выручку компаний.

При разработке рекомендательных систем возникает множество различных задач, одной из которых является необходимость устранения popularity bias. Popularity bias – это склонность рекомендательной системы предлагать пользователям популярные объекты независимо от того, насколько данные объекты релевантны их интересам. Такой эффект влечёт за собой множество негативных последствий. Во-первых, такое распределение не является справедливым по отношению к нишевым объектам с низкой популярностью. Во-вторых, использование только популярных объектов сильно снижает разнообразие рекомендаций и укрепляет имеющиеся тренды.

В-третьих, возрастание популярности может не коррелировать с точностью и релевантностью, что приводит к общему снижению качества рекомендаций и ухудшению пользовательского опыта.

Существует множество подходов к устранению popularity bias в рекомендательных системах, и особое место среди них занимает добавление компоненты регуляризации при обучении модели интересов пользователей. К сожалению, получение прямой оценки популярности в рекомендациях является трудоёмкой задачей, поэтому обычно соответствующие работы сводятся к построению некоторой метрики, которая косвенно оценивает разнообразие рекомендаций, и последующем добавлении данной метрики в функцию потерь.

В рамках данной работы приводится пример построения регуляризации, основанной на сходстве популярности объектов в обучающей выборке и в персональных рекомендациях. В частности, здесь демонстрируется, что распределение популярности подчиняется логнормальному закону, определяется вид компоненты регуляризации и предлагается новый метод сэмплирования при обучении, который позволяет оценивать популярность объектов внутри одного батча.

1. Обзор литературы

Среди работ, посвящённых использованию регуляризации для устранения popularity bias, следует отметить Countering Popularity Bias by Regularizing Score Differences [2] и Incorporating Diversity in a Learning to Rank Recommender System [5].

В исследовании [2] авторы разрабатывают подход к регуляризации, направленный на улучшение разделения положительных и отрицательных примеров. Представлено два метода регуляризации: ZeroSum и Pos2Neg2. В рамках ZeroSum минимизируется модуль суммы предсказаний для положительного и отрицательного объекта, что приводит к симметричным по модулю, но противоположным по знаку оценкам рекомендательной системы. В методе Pos2Neg2 минимизируется абсолютная величина разности предсказаний между парами положительных и между парами отрицательных объектов, что позволяет сблизить предсказания модели внутри этих групп. Оба варианта регуляризации позволяют получить статистически значимый прирост метрик качества рекомендаций.

В работе [5] авторы рассматривают только рекомендательные системы, основанные на методах разложения матрицы взаимодействий пользователей с объектами, а также предлагают новую метрику intra-list diversity для количественной оценки разнообразия рекомендаций. В статье рассматриваются рекомендации фильмов, а в основе предлагаемых компонент регуляризации лежит матрица расстояний между объектами, основанная на их жанрах. Авторы отмечают, что добавление компоненты регуляризации приводит к снижению общего качества рекомендации, вызывая trade-off между точностью и разнообразием.

2. Метод

2.1. Задача построения персональных рекомендаций

Пусть U – множество пользователей сервиса, использующего рекомендательные технологии, а I – множество объектов, которые могут быть использованы в рекомендациях. Функционирование сервиса разделено на периоды, например, на дни или недели. Множество объектов, с которыми пользователь $u \in U$ взаимодействовал в периоде t обозначим $S_t(u) \subset I$, а множество всех пользовательских взаимодействий – $\mathcal{S}_t = \{S_t(u)\}_{u \in U}$. Множество объектов, рекомендованных пользователю u в период t обозначим $R_t(u)$, а множество всех рекомендаций, соответственно, $\mathcal{R}_t$. Задача построения рекомендаций состоит в том, чтобы по тройке $(U, I, \mathcal{S}_{t-1})$ построить рекомендации $\mathcal{R}_t$ так, чтобы максимизировать некоторую целевую функцию $L = L(\mathcal{S}_t, \mathcal{R}_t)$ при условии, что $S_t(u) \subseteq R_t(u) \quad \forall u \in U$. Обычно в качестве такой функции используется hit rate (HR):

$$L(\mathcal{S}_t, \mathcal{R}_t) = \frac{1}{nk} \sum_{u \in U} |S_t(u) \cap R_t(u)|,$$

где $n = |U|$, $k = |R_t(u)| \quad \forall u \in U$ и выполняется условие $S_{t-1}(u) \cap R_t(u) = \emptyset$.

Популярность объектов в рекомендательной системе – это количество взаимодействий с объектом, нормированное на количество пользователей.

$$p_t(i) = \frac{1}{n} \sum_{u \in U} 1_{S_t(u)}(i),$$

где $1_{S_t(u)}$ – индикаторная функция множества $S_t(u)$.

Популярность обычно подчиняется степенному закону, то есть, количество популярных объектов очень небольшое, в то время как остальные объекты обладают низкой популярностью. На рисунке 1 приведен пример распределения популярности фильмов в наборе данных MovieLens-1M [14].

Существует несколько распределений, которые подчиняются степенному закону, например, логнормальное [3] или распределение Парето [9]. В следующем пункте будет представлена модель распространения популярности в социальном графе, которая приводит к логнормальному распределению.

2.2. Распространение популярности в социальном графе

Пусть $G = G(U, A, w)$ – взвешенный ориентированный социальный граф, вершинами которого являются пользователи U рекомендательной системы, а дуги A и веса w определяют силу влияния пользователей друг на друга. Пусть $\deg^+(u)$ и $\deg^-(u)$ полустепени исхода и захода вершины u соответственно.

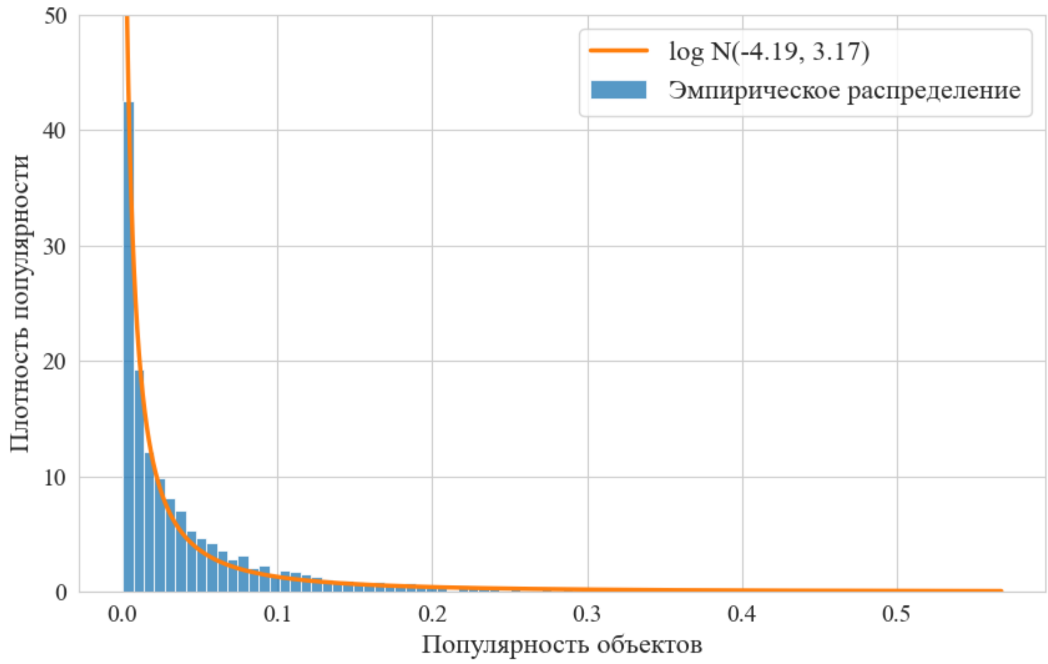

Рисунок 1 – Эмпирическая плотность распределения популярности объектов в наборе данных MovieLens-1M и его аппроксимация плотностью логнормального распределения, полученная методом максимального правдоподобия

Будем считать, что взаимодействие пользователей с объектом в периоде t зависит от взаимодействий предыдущего периода:

$$E[p_t(i)] = \frac{1}{n}\sum_{u\in U} 1_{S_{t-1}(u)}(i) \sum_{u\to v\in A} w_{u\to v},$$

то есть, веса графа отражают вероятность взаимодействия пользователя v с i при условии, что u ранее провзаимодействовал с i. Для дальнейшего упрощения модели дополнительно предположим, что веса $w_{u\to v}$, полустепени исхода $\deg^+(u)$ и $1_{S_{t-1}(u)}(i)$ независимы в совокупности. В таком случае

$$E[p_t(i)] = p_{t-1}(i)E[\deg^+(u)]E[w_{u\to v}] = \ p_0(i)(E[\deg^+(u)])^t(E[w_{u\to v}])^t$$

Следовательно, популярность объектов в подобном графе носит мультипликативный характер, и по центральной предельной теореме может быть смоделирована логнормальным распределением: $p_t(i) \sim \log\mathcal{N}(\mu,\sigma^2)$. На рисунке 1 представлена логнормальная аппроксимация распределения популярности в наборе данных MovieLens-1M.

2.3. Логнормальная регуляризация

Опираясь на предположение о том, что популярность хорошо аппроксимируется логнормальным распределением, можно построить функцию потерь, которая сравнивала бы истинную популярность из обучающей выборки и модельную популярность объектов [3, 10]. Для этого можно воспользоваться тем фактом, что при фиксированном параметре σ^2 логнормальное распределение относится к экспоненциальному семейству [15]. Его функция дисперсии в таком случае может напрямую быть выражена через математическое ожидание.

$$E[y] = e^{\mu + \frac{\sigma^2}{2}}$$

$$V[y] = (e^{\sigma\wedge 2} - 1)e^{2\mu+\sigma^2} = \left(e^{\sigma^2} - 1\right)(E[y])^2$$

Следовательно, $Var(t) = \left(e^{\sigma^2} - 1\right)t^2$. Для построения функции потерь воспользуемся формулой единичного отклонения:

$$d(y, \hat{y}) = 2\int_{\hat{y}}^{y} \frac{y-t}{Var(t)} dt = \frac{2}{e^{\sigma^2} - 1}\left(\frac{y}{\hat{y}} - \log\frac{y}{\hat{y}} - 1\right).$$

На рисунке 2 изображена зависимость единичного отклонения от отношения $\frac{y}{\hat{y}}$.

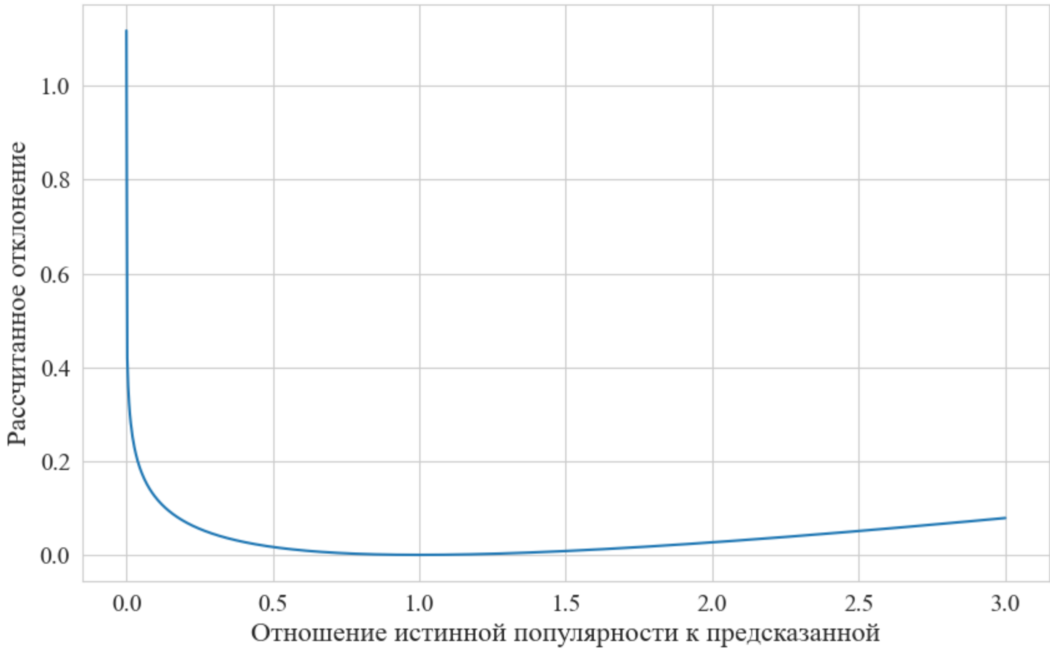

Рисунок 2 – Зависимость логнормального единичного отклонения от отношения истинной популярности к предсказанной

Построенная функция потерь неотрицательна и равняется нулю тогда и только тогда, когда $\hat{y} = y$. Примечательно, что функция не определена, когда одно из её значений равно нулю.

2.4. Оценка популярности объекта внутри батча

Одним из наиболее распространённых способов построения рекомендательных систем на данный момент является обучение нейросетевых моделей на pairwise loss [11]. Обозначим $m(u,i)$ оценку релевантности объекта i пользователю u некоторой нейросетевой моделью. Тогда функцию потерь для обучения можно записать в виде:

$$L = \frac{1}{|B|} \sum_{(u,i^+,i^-)\in B} \log \sigma\big(m(u,i^+) - m(u,i^-)\big),$$

где i^+ – объект, с которым пользователь взаимодействовал, i^- – объект, с которым пользователь не взаимодействовал, обычно он сэмплируется из белого шума, B – один батч для обучения.

При таком подходе к сэмплированию у каждого объекта разное количество пользователей, в тройки к которым он попадает, поэтому оценивать по ним популярность нельзя. Кроме того, в случае небольших $|B|$ у большинства объектов будет всего одна тройка. Поэтому, чтобы использовать логнормальное единичное отклонение, тройки следует генерировать следующим образом.

Алгоритм расчёта функции потерь

Входные данные:

n_B – количество пользователей в батче

m_B – количество позитивных объектов у пользователя в батче

k_B – количество рекомендаций

$\mathcal{S}_{t-1}$ – история взаимодействий пользователей в виде пар $\{(u,i) | i \in \mathcal{S}_{t-1}(u)\}$

$l(u,i,j) = \log \sigma\big(m(u,i) - m(u,j)\big)$ – функция потерь на одной тройке

$s(u,i,j) = 1_{\mathcal{S}_{t-1}(u)}(i)\left(1 - 1_{\mathcal{S}_{t-1}(u)}(j)\right)$ – индикаторная функция, определяющая, что i – позитивный объект для u, а j – негативный

$\lambda \leftarrow$ сила регуляризации

Алгоритм расчёта функции потерь:

$\mathcal{S} \leftarrow \mathcal{S}_{t-1}$ – скопировать список позитивных взаимодействий

while $\mathcal{S} \neq \emptyset$ do:

  $U_B \leftarrow$ выбрать n_B пользователей так, чтобы у них были пары в $\mathcal{S}$

  $I_B \leftarrow$ у каждого пользователя выбрать m_B объектов из $\mathcal{S}$

  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(u,i) | u \in U_B \wedge i \in I_B\}$ – удалить из $\mathcal{S}$ выбранные пары

  оценить основную функцию потерь

$$L(B) = \left(\sum_{u \in U_B} \sum_{i \in I_B} \sum_{j \in I_B} s(u,i,j)\right)^{-1} \sum_{u \in U_B} \sum_{i \in I_B} \sum_{j \in I_B} s(u,i,j) l(u,i,j)$$

  оценить истинную и модельную популярность объектов по выборке

$$p_{t-1}^B(i) \leftarrow \sum_{u \in U_B} 1_{S_{t-1}(u)}(i)$$

$$\hat{p}_{t-1}^B(i) \leftarrow \sum_{u \in U_B} 1\left[\sum_{j \in I_B} 1[m(u,i) \leq m(u,j)] < k\right]$$

оценить компоненту регуляризации

$$\Omega(B) = \frac{2}{|I_B|\left(e^{\sigma^2} - 1\right)} \sum_{i \in I_B} \left(\frac{p_{t-1}^B(i)}{\hat{p}_{t-1}^B(i)} - \log\frac{p_{t-1}^B(i)}{\hat{p}_{t-1}^B(i)} - 1\right)$$

вычислить функцию потерь по батчу

$$obj(B) = L(B) + \lambda\Omega(B)$$

При таком подходе к генерации количество троек у каждого пользователя может быть разным, но при этом популярность каждого объекта в батче оценивается по одинаковому количеству пользователей и одинаковому количеству «конкурентов» – объектов, которые могут быть выбраны в качестве рекомендаций вместо данного.

Предложенная функция обладает ещё одним интересным преимуществом по сравнению с классическими подходами к регуляризации. Из-за ограничения на k регуляризации подвергаются не все объекты, а только те, кто получил наибольшие значения $m(u,i)$. Это позволяет естественным образом не подвергать регуляризации объекты, значение $m(u,i)$ для которых мало или нестабильно.

3. Числовые расчёты

Для численной проверки новой функции потерь был выбран набор данных MovieLens-1M. Набор данных содержит около одного миллиона анонимных оценок примерно 3,900 фильмов, сделанных 6,040 пользователями MovieLens, которые присоединились к MovieLens в 2000 году [14].

По предложенному алгоритму была обучена модель матричной факторизации [7, 11]. В модели матричной факторизации для каждого объекта i и для каждого пользователя u строятся эмбеддинги $e_i, e_u \in \mathbb{R}^d$ соответственно, и

$$m(u, i) = e_u^\top e_i,$$

где d – размерность эмбеддингов, была выбрана равной 64.

Модель обучалась на 20 эпохах, параметры алгоритма оценки функции потерь представлены в таблице 1.

Таблица 1 – Параметры сэмплирования в численных экспериментах

Параметр	**Значение**
n_B	50
m_B	5
k_B	10

Всего было проведено несколько экспериментов с разной силой регуляризации λ. Для оценки popularity bias была выбрана метрика popularity lift (PL):

$$PL(\mathcal{S}_t, \mathcal{R}_t) = \frac{1}{nk} \sum_{u \in U} \sum_{i \in R_t(u)} \frac{p_t(i) - \overline{p_t(u)}}{\overline{p_t(u)}},$$

где $\overline{p_t(u)}$ – средняя популярность объектов, с которыми взаимодействовал u:

$$\overline{p_t(u)} = \sum_{i \in S_t(u)} p_t(i).$$

Результаты расчётов приведены в таблице 2.

Таблица 2 – Результаты численных экспериментов

Сэмплирование	$\boldsymbol{\lambda}$	**HR**	**PL**
Классическое	0.0	0.0081	0.1233

Авторское	0.0	0.0079	0.1074
	0.1	0.0084	0.0235
	1.0	0.0083	0.0302
	10.0	0.0086	0.0226
	100.0	0.0084	0.0211

Сравнительный анализ показывает, что введённая регуляризация действительно оказывает серьёзное влияние на PL (снижение почти в 4 раза), что свидетельствует об уменьшении popularity bias модели, а также немного увеличивает HR (примерно на 9%).

Заключение

В данной статье были рассмотрены основные статистические закономерности, характеризующие распространение популярности объектов в рекомендательных системах, а также представлен формальный подход к моделированию популярности с учетом структуры социального графа. На примере реальных данных из набора MovieLens-1M было показано, что эмпирическое распределение популярности объектов хорошо аппроксимируется логнормальным законом, что объясняется мультипликативным характером распространения взаимодействий между пользователями.

Предложенная модель распространения популярности в социальном графе позволяет описывать динамику популярности на основании структуры пользовательских связей и параметров их взаимодействий, а также строить различные функции потерь для обучения рекомендательных систем на основе гипотезы о форме распределения популярности. Использование аналитических выражений для оценки функции потерь с учетом свойств логнормального распределения позволяет более точно настраивать моделей рекомендаций и учитывать особенности распространения интереса пользователей к объектам.

К перспективным направлениям развития работы стоит отнести дальнейшее исследование влияния топологии социального графа на параметры логнормального распределения, а также рассмотрение других форм распределения популярности объектов.

Список использованных источников

1. A Computationally Efficient Method for Learning Exponential Family Distributions // JMLR Workshop and Conference Proceedings. – 2015. – Vol. 38.

2. Countering Popularity Bias by Regularizing Score Differences // Proceedings of the 13th ACM Conference on Recommender Systems. – 2019.

3. Crow E.L., Shimizu K. Lognormal Distributions, Theory and Applications. – Statistics: Textbooks and Monographs, Vol. 88. – New York: Marcel Dekker, Inc., 1988. – 387 p.

4. Duchi J., Hazan E., Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2121–2159. – URL: http://dl.acm.org/citation.cfm?id=1953048.2021068

5. Incorporating Diversity in a Learning to Rank Recommender System // Proceedings of the 28th ACM International Conference on Information and Knowledge Management. – 2019.

6. Isinkaye F.O., Folajimi Y.O., Ojokoh B.A. Recommendation systems: Principles, methods and evaluation // Egyptian Informatics Journal. – 2015.

7. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems // Computer. – 2009. – Vol. 42, No. 8. – P. 30–37. – doi:10.1109/MC.2009.263.

8. Lalmas M. Algorithmic Effects on the Diversity of Consumption on Spotify // Proceedings of The Web Conference 2020 (WWW '20). – April 20–24, 2020, Taipei, Taiwan. – ACM, New York, NY, USA. – 2020. – 11 p. – https://doi.org/10.1145/3366423.3380281

9. Malik H.J. Estimation of the Parameters of the Pareto Distribution // Metrika. – 1970. – Vol. 15. – P. 126–132. – doi:10.1007/BF02613565.

10. McCullagh P., Nelder J. Generalized Linear Models. 2nd ed. – Boca Raton: Chapman and Hall/CRC, 1989. – 532 p.

11. Ricci F., Rokach L., Shapira B., Kantor P.B., eds. Recommender Systems Handbook. – Springer, 2011.

12. Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendation // Proceedings of the 16th International Conference on Data Mining. – 2016.

13. Silveira T., Zhang M., Lin X., Liu Y., Ma S. How good your recommender system is? A survey on evaluations in recommendation // ACM Computing Surveys. – 2019.

14. The MovieLens Datasets: History and Context // ACM Transactions on Interactive Intelligent Systems. – 2015.

15. Jørgensen B. The theory of exponential dispersion models and analysis of deviance. – Monografias de matemática, No. 51. – 1992.