Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математический методов прогнозирования

Молчанов Дмитрий Александрович

Масштабируемые методы автоматического определения значимости

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
к.ф.-м.н.,
Д.П. Ветров

Москва, 2016

Содержание

1 Введение

Данная работа посвящена разработке новых эффективных методов обучения для вероятностных моделей, обладающих свойством автоматического определения релевантности (Automatic Relevance Determination, ARD), которые смогли бы работать на выборках большого размера. ARD-регуляризованные модели — один из способов построения моделей, позволяющих проводить автоматический отбор признаков во время обучения, основанный на Байесовском подходе к выбору модели. Этот подход также позволяет осуществить прореживание модели и сократить число используемых параметров, что приводит к упрощению модели и улучшению ее обобщающей способности. Это особенно актуально в популярных последнее время моделей глубинного обучения, где число параметров может быть очень велико (миллионы и более), а сами модели подвержены переобучению.

В данной работе рассматриваются три различные вероятностные модели (линейная регрессия, логистическая регрессия и машина релевантных тегов) и их модификации, позволяющие добиться ARD-эффекта. Предложено несколько новых способов обучения этих моделей, работающих быстрее и/или точнее существующих аналогов.

Также предложен новый подход к получению ARD-эффекта, основанный на недавней технике вариационного дропаута (Variational Dropout). Этот подход основан на идее, ранее не применявшейся для решения задач такого типа, поэтому это является интересным и важным результатом.

Работа структурирована следующим образом:

1. В четвертом разделе описываются рассмотренные модели, а также известные способы ARD-регуляризации для этих моделей;

2. В пятом разделе приводится обзор методов и подходов, на которых основываются предложенные методы;

3. В шестом разделе описываются применение уже существующих методов к обучению рассматриваемых моделей;

4. В седьмом разделе описываются методы, предложенные в данной работе;

5. В восьмом разделе приводятся описание и результаты практического исследования предложенных методов и их экспериментального сравнения с существующими аналогами.

2 Обзор литературы

Эффективные и точные методы обучения простых (линейная и логистическая регрессия) ARD-регуляризованных моделей на небольших выборках давно известны. Первые результаты в этом направлении были получены Типпингом ([18]), Джааккола и Джорданом ([6]). Дальнейшие исследования таких моделей в основном заключались в использовании приближенного вариационного вывода ([4]) и стохастического вариационного вывода ([19]). Также в последнее время были предложены возможные обобщения этого подхода на более сложные модели. Так, этот подход, например, использовался для выбора размерности пространства скрытых переменных в глубинных генеративных моделях ([8]). В работе [11] также рассматривается похожий подход для регуляризации и разреживания нейронных сетей.

3 Используемые обозначения

В работе активно используется матричная нотация. Заглавные символы, набранные жирным прямым шрифтом, обозначают матрицы: $\mathbf{A} \in \mathbb{R}^{D \times D}$. Элементы матрицы обозначаются соответствующими строчными символами с необходимыми нижними индексами: $(\tilde{\mathbf{A}})_{ij} = \tilde{a}_{ij}$. Строчные символы, набранные жирным курсивным шрифтом, обозначают вектора или вектор-столбцы: $\boldsymbol{x} \in \mathbb{R}^D$. Элементы вектора обозначаются аналогично элементам матрицы: $(\boldsymbol{x})_d = x_d$.

Символом $\circ$ обозначается покомпонентное произведение векторов или матриц, например, $(\boldsymbol{a} \circ \boldsymbol{b})_i = a_i b_i$. Для краткости записи применение вещественнозначной функции вещественной переменной к вектору или матрице означает покомпонентное применение этой функции к элементам вектора или матрицы. Так, например, $\boldsymbol{\theta}^2$ обозначает вектор $(\theta_1^2, \ldots, \theta_D^2)^T$, а $\log \boldsymbol{\alpha}$ обозначает вектор $(\log \alpha_1, \ldots, \log \alpha_D)^T$.

Во всех задачах, рассмотренных в работе, используются выборки данных, представляющие собой списки объектов. Во всех разделах данной работы число объектов в обучающей выборки обозначается как N, число признаков обозначается как D. Объекты задаются D-мерными векторами: $\boldsymbol{x}_n \in \mathbb{R}^D$. Объекты обучающей выборки обозначается как $\mathbf{X}$ и задается матрицей «объекты-признаки» размера $N \times D$. Вектор из целевых переменных обучающей выборки обозначается как $\boldsymbol{t}$. В данной работе активно используются методы стохастической оптимизации. В этих методах активно используются случайные подвыборки обучающей выборки. Размер таких подвыборок обозначается как M, а сами случайные

подвыборки обозначаются как $(\tilde{\mathbf{X}}^M, \tilde{\boldsymbol{t}}^M)$. Считается, что на каждом шаге итерационного метода оптимизации эта подвыборка генерируется заново.

4 Используемые модели

В данной работе ARD-эффект исследуется на модификациях следующих моделей:

1. Линейная регрессия
2. Логистическая регрессия
3. Машина релевантных тегов

4.1 Линейная регрессия

Модель линейной регрессии предназначена для описания задачи регрессии. Целевая переменная t принимает вещественные значения: $t \in \mathbb{R}$. Объект $\boldsymbol{x}$ задается d-мерным вектором его признаков с вещественными компонентами: $\boldsymbol{x} \in \mathbb{R}^d$. Предполагается, что целевая переменная t связана с объектом $\boldsymbol{x}$ и параметрами модели w и β следующим образом:

$$t = \boldsymbol{x}^T\boldsymbol{w} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(\varepsilon \,|\, 0, \beta^{-1}),$$

где ε — нормальный шум, независимый от объекта и независимый и одинаково распределенный для всех объектов.

Тогда функция правдоподобия в соответствующей вероятностной модели будет выглядеть следующим образом:

$$p(t \,|\, \boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t \,|\, \boldsymbol{w}^T\boldsymbol{x}, \beta^{-1}),$$

4.2 Логистическая регрессия

Модель логистической регрессии служит для описания задачи классификации на два класса. Целевая переменная t принимает значения 1 или -1 — метки классов. В этой модели считается, что функция правдоподобия имеет следующий вид:

$$p(t \,|\, \boldsymbol{x}, \boldsymbol{w}) = \sigma(t\boldsymbol{x}^T\boldsymbol{w}),$$

где $\sigma(a) = \frac{1}{1+\exp(-a)}$ — так называемая логистическая функция, или сигмоида.

4.3 Машина релевантных тегов

Машина релевантных тегов — модель для решения задачи бинарной классификации объектов с бинарными признаками. Эта модель была впервые преложена в моей курсовой работе "Машина релевантных тегов". Признаки объекта имеют смысл тегов. Каждый объект задается множеством своих тегов и считается, что только они влияют на метку класса этого объекта. Целевая переменная $t \in \{0, 1\}$, вектор признаков объекта $\boldsymbol{x} \in \{0, 1\}^d$. $x_d = 1 \Leftrightarrow$ объект $\boldsymbol{x}$ отмечен тегом d. Параметры модели — D-мерный вектор w, $w_d \in (0, 1)$. Предполагается, что теги влияют на значение метки класса независимо, причем:

$$\mathsf{P}(t = 1 \,|\, x_d = 1) = w_d;$$

Тогда при дополнительном предположении о сбалансированности классов функция правдоподобия в модели машины релевантных тегов будет выглядеть следующим образом:

$$\mathsf{P}(t \,|\, \boldsymbol{x}, \boldsymbol{w}) = \frac{\prod\limits_{d=1}^{D} w_d^{tx_d}(1 - w_d)^{(1-t)x_d}}{\prod\limits_{d=1}^{D} w_d^{x_d} + \prod\limits_{d=1}^{D}(1 - w_d)^{x_d}}$$

4.4 ARD для описанных моделей

Для описанных моделей известны модификации, позволяющие добиться ARD-эффекта. Так, для моделей линейной и логистической регрессии наиболее известная такая модификация — машина релевантных векторов. В эти модели на вектор весов $\boldsymbol{w}$ вводится нормальное априорное распределение с диагональной матрицей ковариации:

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \,|\, \boldsymbol{0}, \operatorname{diag}(\boldsymbol{\alpha})^{-1})$$

В модели машины релевантных векторов априорное распределение задается следующим образом:

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \operatorname{Beta}(w_d \,|\, \alpha_d + 1, \alpha_d + 1)$$

Таким образом, в модели вводится вектор гиперпараметров $\boldsymbol{\alpha}$. Во всех случаях $\alpha_d \geq 0$. $\alpha_d = 0$ означает отсутствие регуляризации соответствующей компоненты вектора весов. При $\alpha_d = +\infty$ же соответствующие априорные распределения вырождаются в дельта-функции. В

случае линейной и логистической регрессии это означает, что соответствующий вес w_d становится обязан равняться нулю. В случае модели машины релевантных векторов — $w_d = 0.5$. В любом случае это означает, что соответствующий признак перестает влиять на значение функции правдоподобия, а значит исключается из модели.

В случае логистической и линейной регрессии такое априорное распределение эквивалентно введению $L2$-регуляризации на вектор весов $\boldsymbol{w}$, где каждой компоненте вектора весов соответствует свой параметр регуляризации α_d. Тогда максимизация регуляризованного функционала соответствует взятию MAP-оценки для вектора весов $\boldsymbol{w}$ в рамках модели релевантных векторов:

$$\boldsymbol{w}^{MAP} = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w} \mid \mathbf{X}, \boldsymbol{t}, \boldsymbol{\alpha}) = \arg\max_{\boldsymbol{w}} \left[\log p(\boldsymbol{t} \mid \mathbf{X}, \boldsymbol{w}) + \frac{1}{2} \sum_{d=1}^{D} \alpha_d w_d^2 \right]$$

Настройка гиперпараметров этих моделей ведется путем максимизации маргинальной функции правдоподобия, или так называемой *обоснованности* модели:

$$E(\boldsymbol{\alpha}, \beta) = \int \left(\prod_{n=1}^{N} p(t_n \mid \boldsymbol{x}_n, \boldsymbol{w}, \beta) \right) p(\boldsymbol{w} \mid \boldsymbol{\alpha}) d\boldsymbol{w} \to \max_{\boldsymbol{\alpha}, \beta}$$

Собственно, ARD-эффект заключается в том, что при максимизации обоснованности таких моделей гиперпараметры, соответствующие нерелевантным признакам, уходят в плюс бесконечность, что соответствует их исключению из модели.

Для машины релевантных векторов регрессии и классификации известны достаточно точные методы обучения, основанные на методе простой итерации ([18]) и последовательном Байесовском выводе ([6, 3]). В случае небольших объемов данных эти методы достаточно эффективны, однако для работы на больших выборках они непригодны.

5 Используемые методы

5.1 Стохастическая оптимизация

Большинство работ, посвященных обработке больших данных, так или иначе используют стохастическую оптимизацию. Этот подход применяется для оптимизации функционалов вида

$$\frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x}_n, \boldsymbol{w}) + g(\boldsymbol{w}) \to \max_{\boldsymbol{w}}$$

Обычный метод градиентного подъема для решения этой задачи заключается в итерационном пересчете параметров $\boldsymbol{w}$ по следующим формулам:

$$\boldsymbol{w}^{t+1} := \boldsymbol{w}^t + \rho_t \left(\frac{1}{N}\sum_{n=1}^{N} \nabla_{\boldsymbol{w}} f(\boldsymbol{x}_n, \boldsymbol{w}^t) + \nabla_{\boldsymbol{w}} g(\boldsymbol{w}^t) \right)$$

Аналогичная формула пересчета для стохастического метода градиентного спуска будет выглядеть так:

$$\boldsymbol{w}^{t+1} := \boldsymbol{w}^t + \rho_t \left(\frac{1}{M}\sum_{m=1}^{M} \nabla_{\boldsymbol{w}} f(\tilde{\boldsymbol{x}}_m, \boldsymbol{w}^t) + \nabla_{\boldsymbol{w}} g(\boldsymbol{w}^t) \right),$$

где $\{\tilde{\boldsymbol{x}}_m\}_{m=1}^{M}$ — случайная подвыборка (минибатч) исходной выборки, своя на каждом шаге. Обычно размер минибатча выбирают много меньшим, чем размер исходной выборки. Это ухудшает скорость сходимости метода, однако значительно упрощает итерации (в N/M раз). В данной работе используется как этот метод, так и его современные модификации (такие методы, как rmsprop [17] и Adam [9]).

5.2 Оптимизация с помощью глобальных нижних оценок

Другим популярным приемом является оптимизация с помощью глобальных нижних оценок. Основная идея этого подхода заключается в том, что исходную оптимизационную задачу $\mathcal{L}(\boldsymbol{w}) \to \max\limits_{\boldsymbol{w}}$ можно заменить на другую, обычно значительно более простую задачу:

$$\tilde{\mathcal{L}}(\boldsymbol{w}, \boldsymbol{\xi}) \leq \mathcal{L}(\boldsymbol{w}) \quad \forall \boldsymbol{w}$$

$$\tilde{\mathcal{L}}(\boldsymbol{w}, \boldsymbol{\xi}) \to \max_{\boldsymbol{w}, \boldsymbol{\xi}}$$

$$\boldsymbol{w}^{t+1} = \arg\max_{\boldsymbol{w}} \tilde{\mathcal{L}}(\boldsymbol{w}, \boldsymbol{\xi}^t); \quad \boldsymbol{\xi}^{t+1} = \arg\max_{\boldsymbol{\xi}} \tilde{\mathcal{L}}(\boldsymbol{w}^{t+1}, \boldsymbol{\xi}).$$

Такая функция $\tilde{\mathcal{L}}(\boldsymbol{w}, \boldsymbol{\xi})$ называется вариационной нижней оценкой на функцию $\mathcal{L}(\boldsymbol{w})$, а параметры $\boldsymbol{\xi}$ называются вариационными параметрами этой оценки. Основным преимуществом данного подхода является

то, что он не требует вычисления исходного функционала или его градиентов. Многие популярные методы оптимизации в машинном обучении являются частными случаями этого подхода. Так, EM-алгоритм и его модификации, алгоритм нахождения неотрицательного матричного разложения и выпукло-вогнутая процедура являются известными примерами таких методов [15].

5.3 Expectation Propagation и Power EP

Метод распространения ожидания (Expectation Propagation, EP, [12]) — метод для аппроксимации произвольного непрерывного распределения, заданного в виде произведения отдельных факторов, произведением распределений из экспоненциального класса распределений.

Основная идея этого метода заключается в так называемой *контекстной аппроксимации*. Пусть исходное распределение задано как $p(x) = \prod_{i=1}^{N} t_i(x)$. Будем искать его приближение в виде $q(x) = \prod_{i=1}^{N} \tilde{t}_i(x)$. Определим *контекст* для фактора $t_i(x)$ следующим образом:

$$q^{\setminus i}(x) = \frac{q(x)}{\tilde{t}_i(x)}$$

EP — итерационный метод, где на каждой итерации перебираются факторы, считаются и фиксируются их контексты, а затем эти факторы приближаются с учетом их контекста:

$$q^{\setminus i}(x)t_i(x) \approx q^{\setminus i}(x)\tilde{t}_i^{new}(x)$$

Если говорить формально, то новое значение приближенного фактора считается путем минимизации обратной KL-дивергенции:

$$\tilde{t}_i^{new} = \frac{\arg\min_{g \in \mathcal{Q}} KL(q^{\setminus i} \cdot t_i \| g)}{q^{\setminus i}} = \frac{\mathrm{proj}_{\mathcal{Q}}\left[q^{\setminus i} \cdot t_i\right]}{q^{\setminus i}},$$

где $\mathcal{Q}$ — какое-то фиксированное семейство распределений из экспоненциального класса, а $\mathrm{proj}_{\mathcal{Q}}[\cdot]$ — так называемый оператор KL-проекции. Интересным свойством этого класса является то, что в случае, когда $\mathcal{Q}$ — семейство из экпоненциального класса распределений, применение оператора KL-проекции эквивалентно приравниванию матожиданий достаточных статистик семейства $\mathcal{Q}$. Таким образом, если $\{\phi_j(x)\}_{j=1}^{m}$ — набор достаточных статистик семейства $\mathcal{Q}$, то:

$$g = \mathrm{proj}_{\mathcal{Q}}[f] \Leftrightarrow \int \phi_j(x)g(x)dx = \int \phi_j(x)f(x)dx \quad \forall j = 1..m$$

В случае когда эти матожидания берутся аналитически, метод EP на практике оказывается достаточно быстрым и точным. Проблемы начинаются, когда интеграл в правой части формулы посчитать аналитически нельзя.

Power EP — одна из модификаций алгоритма распространения ожидания, расширяющая границы применимости этого метода [14]. Основная его идея заключается в том, что даже если интегралы вида $\int \phi_j(x) q^{\setminus i}(x) t_i(x) dx$ не могут быть вычислены аналитически, для широкого круга задач интегралы вида $\int \phi_j(x) q^{\setminus i}(x) (t_i(x))^{\eta_i} dx$ могут быть вычислимы при некоторых η_i. В таком случае предлагается делать пересчет приближенных факторов следующим образом:

$$\tilde{t}_i^{new} = \left(\frac{\text{proj}_{\mathcal{Q}} \left[q^{\setminus i} \cdot t_i^{\eta_i} \right]}{q^{\setminus i}} \right)^{\frac{1}{\eta_i}}$$

Этот метод обладает такими же теоретическими гарантиями, что и метод EP (то есть очень слабыми), однако неплохо работает на практике.

5.4 Дважды стохастический вариационный вывод

Дважды стохастический вариационный вывод (DSVI, [19]) — достаточно новая общая процедура настройки вероятностных моделей, предназначенная для работы в условиях данных большой размерности и сложных, несопряженных моделей. Этот подход основан таких приемах как стохастический вариационный вывод ([1]) и репараметризация (reparametrization trick, [10]).

Пусть вероятностная модель задана своим совместным вероятностным распределением $g(\boldsymbol{\theta}) = p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$, где $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ — функция правдоподобия, а $p(\boldsymbol{\theta})$ — априорное распределение. Задача заключается в поиске апостериорного распределения на вектор параметров $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{g(\boldsymbol{\theta})}{\int g(\boldsymbol{\theta}) d\boldsymbol{\theta}}$$

Обычно эта величина не может быть вычислена аналитически, поэтому она ищется приближенно:

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \approx q(\boldsymbol{\theta} \mid \boldsymbol{\xi}) \Leftrightarrow KL(q(\boldsymbol{\theta} \mid \boldsymbol{\xi}) \,||\, p(\boldsymbol{\theta} \mid \boldsymbol{y})) \to \min_{\boldsymbol{\xi}}$$

Известно [7], что эта оптимизационная задача эквивалентна максимизации следующей нижней оценки на логарифм маргинальной функции правдоподобия:

$$\mathcal{F}(\boldsymbol{\xi}) = \int q(\boldsymbol{\theta} \mid \boldsymbol{\xi}) \log \frac{g(\boldsymbol{\theta})}{q(\boldsymbol{\theta} \mid \boldsymbol{\xi})} d\boldsymbol{\theta} \to \max_{\boldsymbol{\xi}}$$

Наконец, в методе DSVI вариационное приближение строится следующим образом. Предполагается, что вектор весов можно представить в виде $\boldsymbol{\theta} = C\boldsymbol{z} + \boldsymbol{\mu}$, где C — нижнетреугольная квадратная матрица с положительными элементами на диагонали, $\boldsymbol{\mu}$ — вещественнозначный вектор подходящей размерности, а $\boldsymbol{z}$ — случайный шум, подчиняющийся некоторому стандартному распределению $\phi(\boldsymbol{z})$. В качестве такого распределения может быть выбрано, например, стандартное нормальное распределение, распределение Стьюдента или любое другое распределение с нулевым матожиданием и независимыми компонентами с единичной дисперсией. Это означает, что вариационное приближение апостериорного распределения ищется в следующем виде:

$$q(\boldsymbol{\theta} \mid \boldsymbol{\mu}, C) = \frac{1}{|C|} \phi(C^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}))$$

Вариационными параметрами в этом случае будут параметры сдвига $\boldsymbol{\mu}$ и параметры масштаба C. В этом случае нижнюю оценку можно переписать в следующем виде:

$$\mathcal{F}(\boldsymbol{\mu}, C) = \mathbb{E}_{\phi(\boldsymbol{z})} \left[\log g(C\boldsymbol{z} + \boldsymbol{\mu}) \right] + \log |C| + \mathcal{H}_{\phi},$$

где $\mathcal{H}_{\phi}$ — энтропия распределения $\phi(\boldsymbol{z})$, константа, не зависящая от $\boldsymbol{\mu}$ и C. Основное преимущество данного метода заключается в том, что градиенты оптимизируемого функционала по вариационным параметрам приобретают очень простой вид и выражаются через градиент логарифма совместной плотности:

$$\nabla_{\boldsymbol{\mu}} \mathcal{F}(\boldsymbol{\mu}, C) = \mathbb{E}_{\boldsymbol{z} \sim \phi(\boldsymbol{z})} \, \nabla_{\boldsymbol{\theta}} \log g(\boldsymbol{\theta})|_{\boldsymbol{\theta} = C\boldsymbol{z} + \boldsymbol{\mu}}$$

$$\nabla_C \mathcal{F}(\boldsymbol{\mu}, C) = \mathbb{E}_{\boldsymbol{z} \sim \phi(\boldsymbol{z})} \left[\nabla_{\boldsymbol{\theta}} \log g(\boldsymbol{\theta})|_{\boldsymbol{\theta} = C\boldsymbol{z} + \boldsymbol{\mu}} \times \boldsymbol{z}^T \right] + \operatorname{diag}(c_{11}^{-1}, \ldots, c_{DD}^{-1})$$

Наконец, вместо взятия матожиданий в этих выражениях предлагается взять их несмещенную оценку, генерируя на каждой итерации по одному семплу $\boldsymbol{z} \sim \phi(\boldsymbol{z})$ (первая стохастичность). Вторая же стохастичность повляется, когда на каждой итерации оптимизируется нижняя оценка не для всей выборки, а для минибатча.

Для контроля сходимости этого метода можно использовать усредненное по окну из нескольких итераций *мгновенное значение нижней оценки*:

$$\mathcal{F}_t(\boldsymbol{\mu}, C) = \log g(C^t \boldsymbol{z}^t + \boldsymbol{\mu}^t) + \log |C^t| + \mathcal{H}_\phi$$

Основное преимущество данного подхода заключается в том, что при обучении модели необходимо только аналитическа задать логарифм совместной плотности. Производные же могут быть эффективно вычислены с помощью процедуры автоматического дифференцирования [5], реализованной во многих популярных прикладных программных пакетах (например, Theano [2]).

На основе этого метода авторами метода был предложен алгоритм DSVI-ARD для обучения ARD-регуляризованный логистической регрессии в условиях больших данных. Его особенность заключается в том, что можно максимизировать вариационную нижнюю оценку по гиперпараметрам $\boldsymbol{\alpha}$ аналитически и получить новый оптимизируемый функционал:

$$\mathcal{F}(\boldsymbol{\mu}, \boldsymbol{c}) = \mathbb{E}_{\phi(\boldsymbol{z})} \left[\log \tilde{g}(\boldsymbol{c} \circ \boldsymbol{z} + \boldsymbol{\mu})\right] + \frac{1}{2} \sum_{d=1}^{D} \log \frac{c_d^2}{c_d^2 + \mu_d^2},$$

$$\alpha_d = \left(c_d^2 + \mu_d^2\right)^{-1}$$

где $\tilde{g}(\boldsymbol{\theta})$ — функция правдоподобия. Так как метод рассчитан на работу с большим числом признаков, матрица C считается диагональной: $C = \mathrm{diag}(\boldsymbol{c})$

5.5 G-KL

G-KL (Gaussian Kullback-Leibler Approximate Inference, [4]) — частный случай описанной выше процедуры дважды стохастического вариационного вывода. В широком классе задач апостериорное распределение на вектор весов можно представить в следующем виде:

$$p(\boldsymbol{w} \mid \mathbf{X}, \boldsymbol{t}) = \frac{1}{Z} \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{n=1}^{N} \phi_n(\boldsymbol{w})$$

В случае, когда каждый множитель ϕ_n зависит только от линейной комбинации компонент вектора $\boldsymbol{w}$ с некоторыми фиксированными весами $\boldsymbol{h}_n$, $\phi_n(\boldsymbol{w}) = \phi_n(\boldsymbol{w}^T \boldsymbol{h}_n)$, получение вариационного приближения апостериорного распределения можно получить достаточно эффективно. Заметим при этом, что априорное распределение не обязательно должно быть нормальным. В таком случае оно должно представляться в виде произведения множителей, зависящих только от линейных комбинаций

компонент вектора весов. Например, в таком виде представляются все априорные распределения, факторизуемые по компонентам вектора весов.

Этот метод также основан на максимизации вариационной нижней оценки на обоснованность. В данной модели эта оценка быдет выглядеть следующим образом:

$$\mathcal{F}(\boldsymbol{m}, \mathbf{S}) = \frac{1}{2}\log\det(2\pi e\mathbf{S}) + \sum_{n=1}^{N}\mathbb{E}_{\mathcal{N}(z\,|\,0,1)}\left[\log\phi_n(m_n + zs_n)\right] -$$

$$-\frac{1}{2}\left[\log\det(2\pi\boldsymbol{\Sigma}) + (\boldsymbol{m}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{m}-\boldsymbol{\mu}) + \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S})\right],$$

где $m_n = \boldsymbol{m}^T\boldsymbol{h}_n$, а $s_n^2 = \boldsymbol{h}_n^T\mathbf{S}\boldsymbol{h}_n$.

Для вычисления этой нижней оценки и ее градиентов по вариационным параметрам требуется считать одномерные интегралы вида $\mathbb{E}_{\mathcal{N}(z\,|\,0,1)}\left[\log\phi_n(m_n + zs_n)\right]$. В некоторых моделях (например, в модели линейной регрессии) это может быть сделано аналитически. В других моделях эти интегралы можно посчитать численно, например, с помощью квадратур Гаусса-Эрмита.

Известно ([4]), что если логарифм каждого множителя в апостериорном распределении вогнутый, то и итоговая нижняя оценка будет вогнутой по совокупности переменных $(\boldsymbol{m}, \mathbf{C})$, где $\mathbf{C}$ — результат разложения Холецкого для матрицы $\mathbf{S}$. Также как и в методе DSVI, в этом методе также можно вводить структурные ограничения на матрицу $\mathbf{C}$ для ускорения работы метода и уменьшения числа параметров.

5.6 Вариационный дропаут

Обычный дропаут — один из методов регуляризации в машинном обучении [16]. Сейчас этот метод особенно популярен при обучении глубинных нейронных сетей и применяется обычно при стохастической градиентной оптимизации параметров модели. Основная идея дропаута заключается в следующем. На каждом шаге метода оптимизации каждый признак каждого объекта зануляется с вероятностью p. Параметр дропаута p обычно является фиксированным. Гауссовский дропаут — аналогичная процедура, при которой каждый признак каждого объекта домножается на величину $\xi_{nd} \sim \mathcal{N}(1, \alpha)$ (то есть к данным добавляется мультипликативный гауссовский шум). Известно, что при $\alpha = \frac{p}{1-p}$ гауссовский дропаут в некотором смысле эквивалентем обычному дропауту с параметром p. Также показано, что данный подход работает не хуже, чем обычный дропаут. Наконец, в вариационном дропауте предлагается

настраивать параметр α адаптивно и, более того, предлагается использовать для каждого признака свое значение параметра α.

Если говорить формально, вариационный дропаут максимизирует следующий функционал:

$$\mathcal{L}(\phi) = -KL(q_\phi(\boldsymbol{w})\|p(\boldsymbol{w})) + L_{\mathcal{D}}(\phi) \to \max_{\phi}$$

$$L_{\mathcal{D}}(\phi) = \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{w})}[\log p(y_n \mid \boldsymbol{x}_n, \boldsymbol{w})]$$

Априорное распределение $p(\boldsymbol{w})$ в данном методе — равномерное в логарифмическом масштабе. Вариационное приближение апостериорного распределения ищется в факторизованном виде с $q_\alpha(w_j) = \mathcal{N}(w_j \mid \theta_j, \alpha_j\theta_j^2)$. Само обучение проводится с помощью дважды стохастической процедуры, аналогичной используемой в методе DSVI.

6 Применение существующих методов к обучению рассматриваемых моделей

6.1 G-KL RVR

Для обучения ARD-регуляризованной линейной регрессии был применен описанный выше метод G-KL. В случае этой модели нижняя оценка и ее градиенты могут быть подсчитаны аналитически. Ковариационная матрица вариационного приближения апостериорного распределения считается диагональной. Также можно последовать примеру метода DSVI-ARD и провести оптимизацию нижней оценки по гиперпараметрам $\boldsymbol{\alpha}$ аналитически. Нижняя оценка тогда будет выглядеть так:

$$\mathcal{F}(\boldsymbol{\mu}, \boldsymbol{c}) = \sum_{n=1}^{N} \mathbb{E}_{\mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I})} \sigma(t_n \boldsymbol{x}_n^T(\boldsymbol{c} \circ \boldsymbol{z} + \boldsymbol{\mu})) + \frac{1}{2}\sum_{d=1}^{D} \log \frac{c_d^2}{c_d^2 + \mu_d^2},$$

$$\alpha_d = \left(c_d^2 + \mu_d^2\right)^{-1},$$

где $\tilde{g}(\boldsymbol{\theta})$ — функция правдоподобия.

Так как логарифм функции правдоподобия этой модели представляет собой квадратичную форму относительно $\boldsymbol{z}$, ее матожидание может быть вычислено аналитически. Итоговое значение нижней оценки тогда будет следующим:

$$\mathcal{F}(\boldsymbol{\mu}, \boldsymbol{c}) = \frac{N}{2}\log\beta - \frac{\beta}{2}\|\mathbf{X}\boldsymbol{\mu} - \boldsymbol{t}\|^2 - \frac{\beta}{2}\mathrm{Tr}[\mathbf{X}^T\mathbf{X}\mathrm{diag}(\boldsymbol{c}^2)] + \frac{1}{2}\sum_{d=1}^{D}\log\frac{c_d^2}{c_d^2+\mu_d^2},$$

При использовании стохастической оптимизации выражение для оценки остается похожим:

$$\mathcal{F}^M(\boldsymbol{\mu}, \boldsymbol{c}) = \frac{N}{2}\log\beta - \frac{\beta}{2}\cdot\frac{N}{M}\|\tilde{\mathbf{X}}^M\boldsymbol{\mu} - \tilde{\boldsymbol{t}}^M\|^2 -$$

$$-\frac{\beta}{2}\cdot\frac{N}{M}\mathrm{Tr}[\tilde{\mathbf{X}}^{M\,T}\tilde{\mathbf{X}}^M\mathrm{diag}(\boldsymbol{c}^2)] + \frac{1}{2}\sum_{d=1}^{D}\log\frac{c_d^2}{c_d^2+\mu_d^2},$$

Градиенты этой оценки были найдены с помощью автоматического дифференцирования. Оптимизация проводилась по $\boldsymbol{\mu}$ и $\log\boldsymbol{c}$, что показало более высокую стабильность работы, чем оптимизация по $\boldsymbol{c}$.

6.2 G-KL RVC

Аналогичным образом этот метод был применен для обучения ARD-регуляризованной логистической регрессии. Нижняя оценка получается точно такой же, как и в методе DSVI-ARD:

$$\mathcal{F}(\boldsymbol{\mu}, \boldsymbol{c}) = \sum_{n=1}^{N}\mathbb{E}_{\mathcal{N}(\boldsymbol{z}\,|\,\mathbf{0},\mathbf{I})}\log\sigma(t_n\boldsymbol{x}_n^T(\boldsymbol{c}\circ\boldsymbol{z}+\boldsymbol{\mu})) + \frac{1}{2}\sum_{d=1}^{D}\log\frac{c_d^2}{c_d^2+\mu_d^2},$$

$$\alpha_d = \left(c_d^2+\mu_d^2\right)^{-1},$$

где $\tilde{g}(\boldsymbol{\theta})$ — функция правдоподобия.

В методе DSVI-ARD для оценки матожидания используется один семпл $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}\,|\,\mathbf{0},\mathbf{I})$. В этом же методе это матожидание считается с помощью квадратур Гаусса-Эрмита. Аналогичным образом считаются и градиенты нижней оценки.

6.3 RTM-DSVI

Другой метод для обучения машины релевантных тегов, рассмотренный в данной работе, основан на дважды стохастическом вариационном выводе. Предлагается сделать репараметризацию $w_d = \sigma(\xi_d)$ и искать

вариационное приближение апостериорного распределение $\boldsymbol{\xi}$ в виде нормального распределения с диагональной матрицей ковариации. Рассматривается два способа дальнейшей настройки модели — RTM-DSVI-1 и RTM-DSVI-2. В методе RTM-DSVI-1 в том же семействе ищется вариационное приближение апостериорного распределения логарифмов гиперпараметров $\log\alpha$. В методе RTM-DSVI-2 вместо исходного априорного распределения на вектор весов $\boldsymbol{w}$ вводится обычный ARD-прайор на репараметризованный вектор $\boldsymbol{\xi}$. Это позволит добиться того же эффекта, так как $\xi_d = 0 \Rightarrow w_d = \frac{1}{2}$, то есть обнуление компонент вектора $\boldsymbol{\xi}$ также означает разреживание модели. В методе RTM-DSVI-2 оптимизация гиперпараметров производится аналитически аналогично методам DSVI-ARD, G-KL RVR и G-KL RVC. Оптимизация полученных нижних оценок производится так же, как и в обычном DSVI.

7 Предложенные методы

7.1 Стохастический RVR

Этот метод заключается в максимизации следующей стохастической оценки на обоснованность модели:

$$E_M(\boldsymbol{\alpha}, \beta) = \int p(\tilde{\boldsymbol{t}}^M \,|\, \tilde{\mathbf{X}}^M, \boldsymbol{w}, \beta)^{\frac{N}{M}} p(\boldsymbol{w} \,|\, \boldsymbol{\alpha}) d\boldsymbol{w},$$

где $(\tilde{\boldsymbol{t}}^M, \tilde{\mathbf{X}}^M)$ — случайная подвыборка исходной выборки размера M. Эта стохастическая оценка может быть вычислена аналитически:

$$\boldsymbol{\Sigma}^{-1} = \beta \frac{N}{M} \tilde{\mathbf{X}}^{M\,T} \tilde{\mathbf{X}}^M + \mathbf{A}$$
$$\boldsymbol{\mu} = \beta \frac{N}{M} \boldsymbol{\Sigma}\, \tilde{\mathbf{X}}^{M\,T}\, \tilde{\boldsymbol{t}}^M$$

$$\log E_M(\boldsymbol{\alpha}, \beta) = \frac{N}{2} \log \frac{\beta}{2\pi} - \frac{N}{2M} \beta\, \tilde{\boldsymbol{t}}^{M\,T}\, \tilde{\boldsymbol{t}}^M + \frac{1}{2} \sum_{d=1}^{D} \log \alpha_d - \frac{1}{2} \log\det(\boldsymbol{\Sigma}^{-1}) + \frac{1}{2} \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}$$

Градиенты по $\log\boldsymbol{\alpha}$ и по $\log\beta$:

$$\nabla_{\log\boldsymbol{\alpha}} \log E_M(\boldsymbol{\alpha}, \beta) = \frac{1}{2}\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha} \circ \mathrm{diag}(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T)$$

$$\nabla_{\log\beta} \log E_M(\boldsymbol{\alpha}, \beta) = \frac{N}{2} - \frac{\beta}{2} \cdot \frac{N}{M}\, \tilde{\boldsymbol{t}}^{M\,T}\, \tilde{\boldsymbol{t}}^M + \frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}\mathbf{A}) - \frac{D}{2} + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{1}{2}\sum_{d=1}^{D} \alpha_d \mu_d^2$$

На каждом шаге алгоритма вычисляются градиенты этой оценки по $\log \boldsymbol{\alpha}$ и $\log \beta$, которые затем могут использоваться в любом современном методе оптимизации типа rmsprop. Сложность одной итерации метода составляет $O(MD^2 + D^3)$. Она не зависит от количества объектов в выборке, однако быстро растет с ростом D, что делает этот метод пригодным только для выборок с небольшим числом признаков.

7.2 Стохастический JJ

Одним из методов обучения модели RVM-классификации является метод, основанный на так называемой локальной вариационной оценке Джааккола-Джордана [6]. Исходная задача выглядит следующим образом:

$$E(\boldsymbol{\alpha}) = \int \left[\prod_{n=1}^{N} \sigma(t_n \boldsymbol{x}_n^T \boldsymbol{w})\right] \mathcal{N}(\boldsymbol{w} \,|\, \mathbf{0}, \operatorname{diag}(\boldsymbol{\alpha})^{-1}) d\boldsymbol{w} \to \max_{\boldsymbol{\alpha}}$$

Этот метод основан на изложенной выше идее использования вариационных оценок. На логарифм обоснованности для этой модели можно получить нижнюю оценку такого вида:

$$\log E(\boldsymbol{\alpha}) \geq \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\xi}) = \int \left[\prod_{n=1}^{N} f_n(\boldsymbol{w}, \xi_n)\right] \mathcal{N}(\boldsymbol{w} \,|\, \mathbf{0}, \operatorname{diag}(\boldsymbol{\alpha})^{-1}) d\boldsymbol{w},$$

$$f_n(\boldsymbol{w}, \xi_n) = \sigma(a_n) \exp\left\{\frac{a_n - \xi_n}{2} - \lambda(\xi_n)(a_n^2 - \xi_n^2)\right\},$$

$$a_n = t_n \boldsymbol{x}_n^T \boldsymbol{w}, \quad \lambda(\xi) = -\frac{1}{4\xi} \tanh\left(\frac{\xi}{2}\right)$$

Для максимизации этой оценки известны следующие формулы пересчета:

$$\boldsymbol{\Sigma}^t = \left(\operatorname{diag}(\boldsymbol{\alpha}^{(t-1)}) + 2\sum_{n=1}^{N} \lambda(\xi_n^{(t-1)}) \boldsymbol{x}_n \boldsymbol{x}_n^T\right)^{-1},$$

$$\boldsymbol{\mu}^t = \frac{1}{2} \boldsymbol{\Sigma}^t \sum_{n=1}^{N} t_n \boldsymbol{x}_n,$$

$$(\xi_n^t)^2 = \boldsymbol{x}_n^T \left(\boldsymbol{\Sigma}^t + \boldsymbol{\mu}^t (\boldsymbol{\mu}^t)^T\right) \boldsymbol{x}_n$$

$$\alpha_d^t = \frac{1 - \alpha_d^{t-1}\Sigma_{dd}^t}{(\mu_d^t)^2}$$

Сложность одной итерации этого метода составляет $O(ND^2 + D^3)$. В случае небольших N и D этот метод работает достаточно быстро, однако для работы с большими выборками этот метод непригоден.

В моей работе предлагается модификация этого метода, аналогичная модификации, использованной в предыдущем разделе. Во-первых, предлагается проводить градиентную оптимизацию по $\boldsymbol{\alpha}$ вместо использования фиксированных формул для пересчета. Во-вторых, вместо подсчета обоснованности всей выборки на каждом шаге метода оптимизации предлагается считать обоснованность случайной подвыборки небольшого размера (батча). Шаг процедуры обучения будет выглядеть следующим образом:

$$\tilde{\xi}_m^2 = \tilde{\boldsymbol{x}}_m^T \left(\boldsymbol{\Sigma}^{(t-1)} + \boldsymbol{\mu}^{(t-1)}(\boldsymbol{\mu}^{(t-1)})^T\right) \tilde{\boldsymbol{x}}_m$$

$$\boldsymbol{\Sigma}^t = \left(\operatorname{diag}(\boldsymbol{\alpha}^{(t-1)}) + 2\frac{N}{M}\sum_{m=1}^{M} \lambda(\tilde{\xi}_m)\tilde{\boldsymbol{x}}_m\tilde{\boldsymbol{x}}_m^T\right)^{-1},$$

$$\boldsymbol{\mu}^t = \frac{N}{2M}\boldsymbol{\Sigma}^t \sum_{m=1}^{M} \tilde{t}_m\tilde{\boldsymbol{x}}_m,$$

$$\nabla_{\log\boldsymbol{\alpha}} E(\boldsymbol{\alpha}, \boldsymbol{\xi}^t) = -\frac{1}{2}\boldsymbol{\alpha} \circ \operatorname{diag}\boldsymbol{\Sigma}^t - \frac{1}{2}\boldsymbol{\alpha} \circ \boldsymbol{\mu} \circ \boldsymbol{\mu} + \frac{1}{2}\mathbf{1}$$

Сложность итерации метода составляет $O(MD^2 + D^3)$. Видно, что сложность не зависит от числа объектов в выборке, что позволяет использовать его на датасетах с большим числом объектов. По количеству признаков сложность не изменилась и осталась достаточно высокой, поэтому, в отличие от аналогов, основанных на дважды стохастическом вариационном выводе, этот метод неприменим к выборкам с большим числом признаков. Тем не менее, в этих аналогах обычно используется приближение апостериорного распределения нормальным с диагональной матрицей ковариации, а в этом методе восстанавливается полная матрица ковариации, поэтому предложенный метод должен быть точнее.

7.3 VD-RVR

Этот метод основан на недавно представленной технике вариационного дропаута.

В оригинальной работе рассматривается только случай $0 < \alpha_d \leq 1$. В моей же работе, напротив, особый интерес представляет случай больших α_d, поскольку в этом случае полученная процедура обучения будет эквивалентна обычному дропауту с параметром $p_d = 1$. Это означает, что признак с номером d будет исключаться из рассмотрения на каждой итерации и не будет влиять на итоговое решающее правило. Значит, его можно исключить из модели.

Итак, в вариационном дропауте вариационное приближение апостериорного распределение ищется в семействе $q(\boldsymbol{w}) = \prod \mathcal{N}(w_d \,|\, \theta_d, \alpha_d \theta_d^2)$. KL-дивергенция не может быть вычислена аналитически, поэтому она была подсчитана численно для всех релевантных значения $\boldsymbol{\alpha}$, а затем приближена функцией следующего вида:

$$-KL(q_{\boldsymbol{\alpha}}(\boldsymbol{w})\|p(\boldsymbol{w})) \approx \sum_{d=1}^{D} \left[-\frac{1}{2}\log(1+\alpha_d^{-1}) + a_1 + a_2\sigma(a_3(\log\alpha_d + a_4)) \right],$$

где $a_1 = 0.03$, $a_2 = 0.64$, $a_3 = 1.5$ и $a_4 = 1.3$.

В случае линейной регрессии имеем:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \frac{N}{2}\log\frac{\beta}{2\pi} - \frac{\beta}{2}\|\mathbf{X}\boldsymbol{\theta} - \boldsymbol{t}\|^2 - \frac{\beta}{2}\mathrm{Tr}\left[\mathbf{X}^T\mathbf{X}\mathrm{diag}(\boldsymbol{\alpha}\circ\boldsymbol{\theta}^2)\right] -$$

$$-KL(q_{\boldsymbol{\alpha}}(\boldsymbol{w})\|p(\boldsymbol{w})) \to \max_{\boldsymbol{\theta},\boldsymbol{\alpha}}$$

Продифференцировав по $\boldsymbol{\theta}$ и $\boldsymbol{\alpha}$, получаем:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = -\beta\mathbf{X}^T(\mathbf{X}\boldsymbol{\theta} - \boldsymbol{t}) - \beta\boldsymbol{\varkappa}\circ\boldsymbol{\theta}$$

$$\nabla_{\log\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = -\frac{\beta}{2}\boldsymbol{\varkappa}\circ\boldsymbol{\theta}^2 - \nabla_{\log\boldsymbol{\alpha}}KL(q_{\boldsymbol{\alpha}}(\boldsymbol{w})\|p(\boldsymbol{w}))$$

$$\boldsymbol{\varkappa} = \boldsymbol{\alpha}\circ\sum_{n=1}^{N}\boldsymbol{x}_n^2$$

Эти формулы можно использовать для градиентной оптимизации. Сложность итерации в таком случае получается равной $O(ND)$. По аналогии с предыдущими методами сюда можно добавить стохастичность, что приведет к тем же модификациям формул, что и в предыдущих методах. Сложность итерации тогда поличится равной $O(MD)$, что приемлемо даже для выборок большого размера с большим числом параметров.

Приравнивая градиент по $\boldsymbol{\theta}$ к нулю, можно получить следующее выражение для пересчета $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^* = \left(\mathbf{X}^T\mathbf{X} + \operatorname{diag}(\boldsymbol{\varkappa})\right)^{-1}\mathbf{X}^T\boldsymbol{t}$$

$$\alpha_d \to +\infty \Rightarrow \varkappa_d \to +\infty \Rightarrow \theta_d^* \to 0$$

Видно, что здесь, как и в моделях машины релевантных векторов и машины релевантных тегов, при стремлении α_d к плюс бесконечности, θ_d стремится к нулю. Также видно, что полученное выражение очень похоже на аналогичное выражение для гребневой регрессии. Это согласуется с результатом, полученным в работе [16], где была показана аналогия между обычным бинарным дропаутом и гребневой регрессией.

Таким образом, от этой модели также можно ожидать ARD-эффекта. При этом в случае этой модели его природа получается совершенно иной. Если, например, в машине релевантных векторов регрессии разреженность получается из-за того, что априорное распределение вырождается в дельта-функцию с центром в нуле, то здесь это происходит при добавлении к данным бесконечно сильного мультипликативного шума.

7.4 VD-RVC

Для модели логистической регрессии можно применить тот же подход. Так, оптимизируемый функционал будет выглядеть следующим образом:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{n=1}^{N} \int \log \sigma(t_n \boldsymbol{x}_n^T \boldsymbol{w}) \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\theta}, \operatorname{diag}(\boldsymbol{\alpha} \circ \boldsymbol{\theta}^2)) d\boldsymbol{w} - KL(q_{\boldsymbol{\alpha}}(\boldsymbol{w}) \| p(\boldsymbol{w}))$$

Матожидание не берется аналитически, однако вычисление градиентов нижней оценки можно свести к подсчету интегралов следующего вида:

$$I_1(\boldsymbol{a}, b) = \int \sigma(\boldsymbol{a}^T\boldsymbol{\varepsilon} + b)\mathcal{N}(\boldsymbol{\varepsilon} \mid \mathbf{0}, \mathbf{I}) d\boldsymbol{\varepsilon}$$

$$I_2(\boldsymbol{a}, b, i) = \int \varepsilon_i \sigma(\boldsymbol{a}^T\boldsymbol{\varepsilon} + b)\mathcal{N}(\boldsymbol{\varepsilon} \mid \mathbf{0}, \mathbf{I}) d\boldsymbol{\varepsilon}$$

После соответствующих замен переменных и отдельного интегрирования по ε_i во втором случае, эти интегралы сводятся к одномерным. В данной работе для подсчета как самой нижней оценки, так и для подсчета ее градиентов используются квадратуры Гаусса-Эрмита. Итоговая сложность одной итерации этого метода получается такой же, как и в случае вариационного дропаута для линейной регрессии: $O(MD)$.

7.5 RTM-PEP

Ранее было предложено два способа для обучения машины релевантных векторов. Один из них был основан на использовании вариационных нижних оценок, а другой — на методе Expectation Propagation. Интегралы, необходимые для приравнивания матожиданий достаточных статистик, не брались аналитически, поэтому приходилось приравнивать матожидания других статистик и считать их численно. В связи с этим метод работал довольно медленно и нестабильно.

В этой же работе предлагается применить к этой модели метод Power EP. С его помощью ищется приближение функции правдоподобия модели в виде произведения ненормированных бета-распределений:

$$q(\boldsymbol{w}) \propto \prod_{n=1}^{N} \prod_{d=1}^{D} \mathrm{Beta}(w_d \,|\, a_{nd}, b_{nd})$$

Определим фактор $t_n(\boldsymbol{w})$ как функцию правдоподобия для объекта x_n. При использовании параметра Power EP $\eta_i = -1$, сумма из знаменателя функции распределения переходит в числитель и приравнивание моментов значительно упрощается:

$$t_n(\boldsymbol{w})^{-1} \cdot q^{\setminus i}(\boldsymbol{w}) \propto \prod_{d=1}^{D} w_j^{A^1_{nd}-1}(1-w_j)^{B^1_{nd}-1} + \prod_{d=1}^{D} w_j^{A^2_{nd}-1}(1-w_j)^{B^2_{nd}-1},$$

$$A^1_{nd} = \sum_{i=1}^{N} a_{id} - N + a_{nd} + (1-t_n)x_{nd}, \quad B^1_{nd} = \sum_{i=1}^{N} b_{id} - N + b_{nj} - (1-t_n)x_{nd}$$

$$A^2_{nd} = \sum_{i=1}^{N} a_{id} - N + a_{nd} - t_n x_{nd}, \quad B^2_{nd} = \sum_{i=1}^{N} b_{id} - N + b_{nj} + t_n x_{nd}$$

Тогда операция приравнивания достаточных статистик оказывается эквивалентна решению следующих нелинейных систем:

$$\begin{cases} \psi(x) - \psi(x+y) = b^1_{nd} \\ \psi(y) - \psi(x+y) = b^2_{nd} \end{cases}$$

$$Z^1_n = \frac{\prod_{d=1}^{D} \mathrm{B}(A^1_{nd}, B^1_{nd})}{\mathrm{B}(A^1_{nd}, B^1_{nd}) + \mathrm{B}(A^2_{nd}, B^2_{nd})}$$

$$Z^2_n = 1 - Z^1_n$$

$$b^1_{nd} = Z^1_n(\psi(A^1_{nd}) - \psi(A^1_{nd} + B^1_{nd})) + Z^2_n(\psi(A^2_{nd}) - \psi(A^2_{nd} + B^2_{nd}))$$

$$b^2_{nd} = Z^1_n(\psi(B^1_{nd}) - \psi(A^1_{nd} + B^1_{nd})) + Z^2_n(\psi(B^2_{nd}) - \psi(A^2_{nd} + B^2_{nd}))$$

где $\psi(x)$ — дигамма-функция, а $\mathrm{B}(a, b)$ — бета-функция. Оказывается, что для решения систем такого вида известен очень эффективный метод решения, основанный на методе Ньютона [13]. Решение этой системы даст параметры результата соответствующей проекции. Обозначим результат решения системы, соответствующей паре (n, d) за $\hat{A}_{nd} = x, \hat{B}_{nd} = y$. Тогда новое значение параметров для приближенного фактора может быть подсчитано по следующей формуле:

$$\tilde{t}^{new}_n = \prod_{d=1}^{D} \mathrm{Beta}\left(\sum_{i=1}^{N} a_{id} - N + a_{nd} + 1 - \hat{A}_{nd}, \sum_{i=1}^{N} b_{id} - N + b_{nd} + 1 - \hat{B}_{nd}\right)$$

Остается заметить, что достаточно обрабатывать только те пары (n, d), для которых $x_{nd} = 1$. В противном случае можно при инициализации выставить $a_{nd} = b_{nd} = 1$ и считать соответствующий «субфактор» фиксированным. В таком случае сложность одной итерации будет пропорциональна количеству единиц в матрице $\mathbf{X}$.

После сходимости Power EP оптимальное значение гиперпараметров может быть найдено путем максимизации аппроксимированной обоснованности модели:

$$\log \int \prod_{n=1}^{N} \mathrm{Beta}(w_d \,|\, a_{nd}, b_{nd}) \mathrm{Beta}(w_d \,|\, \alpha_d + 1, \alpha_d + 1) dw_d \to \max_{\alpha_d \geq 0}$$

Этот интеграл может быть вычислен аналитически, а оптимизационная задача может быть решена любым подходящим методом оптимизации. Время работы этой части метода будет в любом случае пренебрежимо мало по сравнению с временем работы Power EP.

Для того, чтобы метод работал более стабильно, быстро и точно, оказалось полезным применить несколько дополнительных приемов. Во-первых, это демпфирование параметров:

$$\tilde{t}_n := (\tilde{t}^{old}_n)^{\gamma} \cdot (\tilde{t}^{new}_n)^{(1-\gamma)}$$

Во-вторых, это последовательная схема пересчета факторов. Вместо того, чтобы на каждой итерации метода фиксировать контексты всех факторов, а затем одновременно обновлять все факторы, факторы обновляются по очереди, а текущий контекст строится с учетом уже обновленных на этой итерации факторов. Наконец, добавление фиксиро-

ванных факторов, соответствующих априорному распределению с фиксированным небольшим значением гиперпараметров, также позволило улучшить работу метода.

8 Эксперименты

8.1 Линейная регрессия

Методы Стохастический RVR, VD-RVR и G-KL RVR были протестированы на синтетических данных. Выборка имела размер 100000 объектов, 100 признаков. 90 признаков были нерелевантны. Методы были обучены при различном размере минибатча (0.1%, 1% и 10% выборки). Точность на тестовой выборке, время работы методов и точность отбора признаков представлены на таблицах.

Среднеквадратичная ошибка на тестовой выборке:

Размер минибатча	0.1%	1%	10%
Стохастический RVR	0.10196	0.10198	0.10198
VD-RVR	0.10257	0.10195	0.10188
G-KL RVR	0.10411	0.10192	0.10185

Время работы:

Размер минибатча	0.1%	1%	10%
Стохастический RVR	1.86	5.43	58.4
VD-RVR	14.2	15.4	13
G-KL RVR	17.1	11.2	10.9

Убрано нерелевантных признаков:

Размер минибатча	0.1%	1%	10%
Стохастический RVR	0%	100%	100%
VD-RVR	47%	66%	69%
G-KL RVR	100%	100%	100%

Основной результат — видно, что ARD-эффект наблюдается на всех методах. Видно, что время работы стохастического RVR почти прямо пропорционально размеру минибатча, в то время как время работы методов VD-RVR и G-KL RVR слабо зависит от размера минибатча. При маленьком размере минибатча лучшую точность показывает стохастический RVR, даже несмотря на то, что при этом ему не удается решить задачу отбора признаков. При более большом размере минибатча лучшую

точность показывает метод G-KL RVR. Метод, основанный на вариационном дропауте же занимает по точности работы второе место. С задачей отбора признаков же он справляется хуже всех остальных методов. Так как число признаков в этом эксперименте невелико, при небольшом размере минибатча методу стохастический RVR удалось сойтись значительно быстрее остальных методов. В связи с более высокой сложностью итерации этого метода, с ростом числа признаков он бы быстро потерял эффективность.

При обучении методов по полной выборке, их поведение сохраняется. Так, например, все методы, кроме VD-RVR, успешно находят все нерелевантные признаки, в то время как VD-RVR находит порядка 75%.

8.2 Логистическая регрессия

Методы Стохастический JJ, VD-RVC и G-KL RVC были протестированы на синтетических данных. Выборка имела размер 100000 объектов, 100 признаков. 90 признаков были нерелевантны. Методы были обучены при различном размере минибатча (0.1%, 1% и 10% выборки). Точность на тестовой выборке, время работы методов и точность отбора признаков представлены на таблицах.

Точность классификации на тестовой выборке:

Размер минибатча	0.1%	1%	10%
Стохастический JJ	0.725	0.915	0.909
VD-RVC	0.734	0.905	0.908
G-KL RVC	0.83	0.91	0.91

Время работы:

Размер минибатча	0.1%	1%	10%
Стохастический JJ	20.1	41.3	189.3
VD-RVC	462.64	238.9	193.6
G-KL RVC	15.8	7.6	47.0

Убрано нерелевантных признаков:

Размер минибатча	0.1%	1%	10%
Стохастический JJ	0%	37%	96%
VD-RVC	1%	19%	56%
G-KL RVC	100%	100%	100%

Здесь также видно, что ARD-эффект наблюдается на всех методах. Метод G-KL RVC при маленьком размере батча помимо нерелевантных

признаков удалил также и 4 релевантных признака. Метод VD-RVC достиг своей точности классификации тестовой выборки за время, сравнимое со временем работы метода G-KL, а оставшееся время потребовалось для настройки параметров $\boldsymbol{\alpha}$. Значения α_d для большинства нерелевантных признаков ушло в большие значения только к концу работы метода. Это может означать, что для этого метода требуется более аккуратный выбор метода оптимизации и более тщательная настройка его параметров.

Здесь же при обучении по полной выборке все методы, включая метод VD-RVC, успешно удаляют все нерелевантные признаки.

8.3 Машина релевантных тегов

В этом разделе сравниваются методы RTM-PEP, RTM-DSVI-1 и RTM-DSVI-2, а также ранее предложенные методы RTM-EM и RTM-EP. Эти методы сравнивались на двух реальных датасетах. Решается задача классификации предложений на позитивные и негативные. Объекты представляются мешком слов, теги каждого объекта — слова предложения, ему соответствующего. В качестве предобработки была проведена лемматизация и стемминг. Также были удалены слова, встречающиеся менее чем в пяти предложениях. Итоговое качество классификации тестовой выборки представлено на таблице.

Точность классификации тестовой выборки:

	RTM-PEP	RTM-EM	RTM-EP
UMICH Sentiment :	0.9708	0.9659	0.9683
Sentiment Treebank:	0.7523	0.7294	0.7398

	RTM-DSVI-1	RTM-DSVI-2
UMICH Sentiment :	0.944	0.956
Sentiment Treebank:	0.7380	0.6638

Размер первого датасета — около 2000 объектов и 800 признаков, второго — 5000 объектов и 2000 признаков. При этом на втором датасете время работы метода RTM-EM составило несколько часов, RTM-EP — порядка часа, RTM-PEP — порядка 15ти минут. Методы RTM-DSVI-1 и RTM-DSVI-2 отработали, соответственно, за 31 минуту и 8 минут. Такое большое время работы последних методов связано с тем, что их реализация была не оптимальна для данной задачи. В этой задаче данные разреженные (только 0.6% элементов матрицы $\mathbf{X}$ являются ненулевыми), а реализация этих методов не использовала разреженность данных.

Видно, что RTM-PEP превосходит все остальные методы по точности. Семейство методов RTM-DSVI же показывает худшую точность. При этом у методов RTM-PEP, RTM-EM и RTM-EP итоговое решение оказалось разреженным, из него было убрано около 70% признаков. Методы же, основанные на DSVI, убрали меньше 5% признаков.

Также методы RTM-DSVI сравнивались с методом RTM-PEP на синтетических данных. Выборка состоит из 10000 объектов и 100 признаков, 90 из которых нерелевантны. Методы RTM-DSVI сошлись примерно за 10 секунд. При этом методу RTM-PEP требуется 11 минут для сходимости. По точности классификации методы получаются сравнивыми: 0.811 у RTM-DSVI-1, 0.801 у RTM-DSVI-2 и 0.821 у RTM-PEP.

Что касается отбора признаков, RTM-PEP успешно находит все нерелевантные признаки и выставляет значение гиперпараметров для них в значения выше 100, что позволяет судить об их нерелевантности. Метод RTM-DSVI-2 не смог настроить ковариационную матрицу вариационного приближения и, соответственно, не смог найти нерелевантные признаки. Метод RTM-DSVI-1 половине нерелевантных признаков выставляет значение гиперпараметра 20, что позволяет судить о плохом качестве признаков, но чего недостаточно для их удаления из модели.

Таким образом, метод RTM-DSVI-2 в целом плохо справился с настройкой модели. Метод RTM-DSVI-1 работает лучше, однако задача отбора признаков им решается довольно плохо. Наконец, метод RTM-PEP работает медленней методов, основанных на дважды стохастической процедуре, однако значительно превосходит их по качеству отбора признаков и точности классификации.

8.4 Общие выводы и замечания

На всех рассмотренных методах и моделях наблюдается ARD-эффект. Особенно интересно наличие ARD-эффекта в методах, основанных на вариационном дропауте. Тем не менее, на линейной регрессии этот эффект оказался менее выражен, чем у остальных методов, а на логистической регрессии этот эффект оказался сильно чувствительным к параметрам метода оптимизации.

Интересным результатом является то, что в экспериментах хорошо себя показали методы стохастический RVR и стохастический JJ. Эти методы отличаются от большинства существующих стохастических методов обучения тем, что целевая функция в этих методах не распадается в сумму по объектам. Так, в случае линейной регрессии в качестве функционала использовался логарифм обоснованности модели, а вариационная оценка не него не строилась. В случае логистической регрессии

использовалась локальная вариационная нижняя оценка на обоснованность, которая также не факторизовалась по объектам. Это должно было затруднить применение стохастической оптимизации в этих методах, однако на практике они показали себя хорошо.

9 Результаты

В этой работе были получены следующие результаты:

- Предложено пять новых методов обучения моделей с возможностью автоматического определения значимости признаков;
- Рассмотрено применение существующих методов к обучению таких моделей;
- Предложенные методы могут быть использованы на больших данных и работают не хуже существующих аналогов или даже превосходят их по точности и/или скорости работы;
- Предложен новый подход к получению ARD-эффекта, основанный на использовании вариационного дропаута;
- Экспериментально подтверждено, что ARD-эффект имеет место и в случае больших выборок, а также сложных нелинейных моделях с априорными распределениями, отличными от нормальных.

10 Дальнейшие планы

В дальнейшей работе планируется более подробно изучить новый подход к получению ARD-эффекта, основанном на вариационном дропауте. В частности, планируется применить этот подход к более сложным моделям, таким как нейронные сети. Также планируется исследовать возможность применения обычной ARD-регуляризации на таких моделях. Ожидается, что работа в данном направлении позволит получить новые способы регуляризации нейронных сетей, позволяющие справиться с переобучением и получить более разреженное решение. Это позволит улучшить как качество работы таких сетей, так и скорость работы и объем требуемой памяти. Также планируется исследование по сочетанию вариационного дропаута с другими априорными распределениями.

Список литературы

[1] Beaver and Clark. Stochastic Variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2012.

[2] James Bergstra, Olivier Breuleux, Frederic Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math compiler in Python. *Proceedings of the Python for Scientific Computing Conference (SciPy)*, (Scipy):1–7, 2010.

[3] Christopher M Bishop. *Pattern Recognition and Machine Learning*, volume 4. 2006.

[4] E Challis and D Barber. Gaussian Kullback-Leibler Approximate Inference. *Journal of Machine Learning Research*, 14:2239–2286, 2013.

[5] Andreas Griewank. On automatic differentiation. *Mathematical Programming: recent developments ...*, (November):1–28, 1989.

[6] Tommi S Jaakkola and Michael I Jordan. A variational approach to Bayesian logistic regression models and their extensions. *Aistats*, (AUGUST 2001), 1996.

[7] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[8] Theofanis Karaletsos and Gunnar Rätsch. Automatic Relevance Determination For Deep Generative Models. may 2015.

[9] Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations*, pages 1–13, 2015.

[10] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint*, (Ml):1–14, 2013.

[11] Christos Louizos. Smart Regularization of Deep Architectures. Master's thesis, University of Amsterdam, 2015.

[12] Thomas P Minka. Expectation Propagation for approximate Bayesian inference. *Uncertainty in Artificial Intelligence, Proc. of (UAI)*, pages 362–369, 2001.

[13] Thomas P. Minka. Estimating a Dirichlet distribution. *Annals of Physics*, 2000(8):1–13, 2003.

[14] Thomas P. Minka. Power EP. Technical report, Microsoft Research, Cambridge, 2004.

[15] Ruslan Salakhutdinov, Sam T Roweis, and Zoubin Ghahramani. On the Convergence of Bound Optimization Algorithms. In *UAI*, number 8, pages 509–516, 2003.

[16] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)*, 15:1929–1958, 2014.

[17] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 2012.

[18] Michael E Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 1:211–245, 2001.

[19] Michalis Titsias and Miguel Lázaro-Gredilla. Doubly Stochastic Variational Bayes for non-Conjugate Inference. *Proceedings of The 31st International Conference on Machine Learning*, 32:1971–1979, 2014.