МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

ИСПОЛЬЗОВАНИЕ РЕГУЛЯРИЗАЦИИ ДЛЯ РЕШЕНИЯ ПРОБЛЕМЫ ПЕРЕОБУЧЕНИЯ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 412 группы
направления 01.03.02 — Прикладная математика и информатика

механико-математического факультета
Георгиева Ивана Владимировича

Научный руководитель
д. ф.-м. н., доцент _____ С. П. Сидоров

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2021

ВВЕДЕНИЕ

Актуальность темы. В современном мире машинное обучение приобретает популярность при решении обширного класса задач. Практически в каждой из таких задач возникает проблема переобучения. При переобучении построенная для решения поставленной задачи модель может давать ответы, близкие к реальным, на данных, используемых для обучения. Но при этом на остальных данных результаты могут значительно отличаться от ожидаемых.

Для решения этой проблемы используется много различных методов. Одним из наиболее актуальных является регуляризация. Так, в популярных библиотеках, предназначенных для машинного обучения, некоторые модели используют регуляризацию автоматически, при условии, что значение соответствующего параметра не предполагает обратного. Переобучение часто наступает при слишком больших значениях некоторых коэффициентов модели. Регуляризация не позволяет модели чересчур увеличивать коэффициенты, то есть снижает уровень переобучения.

Целью бакалаврской работы является исследование регуляризации для решения проблемы переобучения и сравнение результатов обучения модели с использованием и без применения регуляризации.

Объект исследования – виды регуляризации.

Предмет исследования – L_1 - регуляризация, L_2 - регуляризация.

Для достижения поставленных целей в работе необходимо решить следующие **задачи**:

— определить основные понятия машинного обучения
— показать случаи переобучения на примерах
— рассмотреть основные виды регуляризации
— собрать данные для обучения модели из открытых источников
— обучить модель с использованием и без использования регуляризации
— провести сравнение результатов обучения

Практическая значимость данной работы состоит в том, что модель, обученная с применением регуляризации, даёт более точный результат чем модель, которую обучали без использования данного метода. Кроме того полученная модель может быть использована для предсказания диагнозов

пациентов по их симптомам. Полученные с её помощью предсказания могут дать направление для дальнейших исследований здоровья больного.

Структура и содержание бакалаврской работы. Работа состоит из введения, трёх разделов, заключения, списка использованных источников, содержащего 20 наименований и приложения. Общий объём работы составляет 45 страниц.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность темы работы, формулируется цель работы и решаемые задачи, отмечается практическая значимость полученных результатов.

В **первом разделе** подробно описываются гребневая регрессия и регрессия лассо.

Регрессия Лассо, или регуляризация через манхэттенское расстояние, или L_1-регуляризация:

$$L_1 = \sum_i (y_i - y(t_i))^2 + \lambda \sum_i |a_i|$$

Гребневая регрессия, или регуляризация Тихонова, или L_2-регуляризация:

$$L_2 = \sum_i (y_i - y(t_i))^2 + \lambda \sum_i a_i^2$$

Добавление многочленов более высокой степени к уравнению регрессии приводит к переобучению. Переобучение происходит, когда модель слишком хорошо подходит для обучающих данных и не обобщается на ранее неизвестные данные.

Переобучение также может произойти, если в уравнении регрессии слишком много независимых переменных или, если наблюдений слишком мало. Переобучение также связано с очень большими оценочными параметрами (весами) $\hat{w}$. Поэтому мы хотим найти баланс между

— Насколько хорошо наша модель соответствует данным (мерой соответствия)

— Величиной коэффициентов

Таким образом, общая стоимость модели представляет собой комби-

нацию меры соответствия и величиной коэффициентов. Мера соответствия представлена суммой квадратов разностей(RSS). Небольшая указывает на хорошее соответствие. Мера величины коэффициентов - это сумма абсолютных значений коэффициентов l_1 норма или сумма квадратов значений коэффициентов l_2 норм. Они представлены следующим образом:

$$\|w_0\| + \|w_1\| + \cdots + \|w_n\| = \sum_{j=0}^{n} \|w_j\| = \|w\|_1 \, (l_1 \text{ норма})$$

$$w_0^2 + w_1^2 + \cdots + w_n^2 = \sum_{j=0}^{n} w_j^2 = \|w\|_2^2 \, (l_2 \text{ норма})$$

В гребневой регрессии мы считаем l_2 норму как меру величины коэффициентов. Таким образом, общая стоимость

$$Total\ Cost = RSS(w) + \|w\|_2^2 \tag{1}$$

Наша цель в гребневой регрессии состоит в том, чтобы найти $\hat{w}$, чтобы минимизировать общие затраты в уравнении 1. Баланс между мерой соответствия и величиной коэффициентов достигается путем введения параметра настройки λ так, чтобы

$$Total\ Cost = RSS(w) + \lambda \|w\|_2^2 \tag{2}$$

Обобщая,

— Для регрессии методом наименьших квадратов: $w_j^{(t+1)} \leftarrow w_j^{(t)} - \eta *$ (коэффициент обновления)

— Для гребневой регрессии: $w_j^{(t+1)} \leftarrow (1 - 2\eta\lambda) w_j^{(t)} - \eta *$ (коэффициент обновления)

Мы всегда можем лучше подобрать обучающую выборку со сложной моделью и настройкой спада веса $\lambda = 0$, чем мы могли бы с менее сложной моделью и положительным спадом веса. Это подводит нас к вопросу о том, как выбрать параметр настройки λ? Чтобы найти λ, мы используем k-кратную перекрестную проверку.

Процесс включает подгонку $\hat{w}_\lambda$ к обучающему набору, тестирование

производительности модели с $\hat{w}_\lambda$ на проверочном наборе для выбора λ^* и, наконец, оценку ошибки обобщения модели с $\hat{w}_{\lambda^*}$. Средняя ошибка вычисляется следующим образом

$$\text{Средняя ошибка } CV(\lambda) = \frac{1}{k}\sum_{k=1}^{k} error_k(\lambda) \tag{3}$$

Обобщая,

— Для регрессии методом наименьших квадратов: $w_j^{(t+1)} \leftarrow w_j^{(t)} - \eta*$ (коэффициент обновления)

— Для гребневой регрессии: $w_j^{(t+1)} \leftarrow (1 - 2\eta\lambda)w_j^{(t)} - \eta*$ (коэффициент обновления)

Мы всегда можем лучше подобрать обучающую выборку со сложной моделью и настройкой спада веса $\lambda = 0$, чем мы могли бы с менее сложной моделью и положительным спадом веса. Это подводит нас к вопросу о том, как выбрать параметр настройки λ? Чтобы найти λ, мы используем k-кратную перекрестную проверку (см. предыдущие разделы).

Процесс включает подгонку $\hat{w}_\lambda$ к обучающему набору, тестирование производительности модели с $\hat{w}_\lambda$ на проверочном наборе для выбора λ^* и, наконец, оценку ошибки обобщения модели с $\hat{w}_{\lambda^*}$. Средняя ошибка вычисляется следующим образом

$$\text{Средняя ошибка } CV(\lambda) = \frac{1}{k}\sum_{k=1}^{k} error_k(\lambda) \tag{4}$$

Если у нас есть «широкий» набор функций (скажем, $1e + 10$), гребневая регуляризация может создать вычислительные проблемы, поскольку выбраны все функции. Алгоритм лассо отбрасывает менее важные / избыточные характеристики, переводя их коэффициенты в ноль. Это позволяет нам интерпретировать функции, а также сокращает время вычислений. Лассо (регуляризованная регрессия l_1) использует норму l_1 в качестве штрафа, вместо нормы l_2 в гребневой регрессии.

Прежде чем мы перейдем к целевой функции лассо, давайте вернемся к алгоритму гребневой регрессии. Гребневая регрессия выбирает параметры β с минимальной RSS при условии, что норма l_2 параметров $\beta_1^2+\beta_2^2+\cdots+\beta_n^2 \leq t$,

ограничение гребня.

В случае гребневой регрессии $\beta_1^2 + \beta_2^2 \leq t$, ограничение гребня. Ограничение принимает форму круга для двух параметров и становится сферой с большим количеством параметров. Первая точка соприкосновения контура RSS с окружностью - это точка, описывающая параметры гребня β_1 и β_2. Значения β в большинстве случаев оказываются ненулевыми. Если t мало, параметры будут маленькими, а если велико, будет стремиться к решению по методу наименьших квадратов.

В случае регрессии лассо параметры β выбираются такими, что $|\beta_1| + |\beta_2| \leq t$, ограничение лассо, для минимального RSS.

Мы можем переписать общую стоимость в формуле **??**, для регрессии лассо как

$$Total\ Cost = RSS(w) + \lambda \left\|w\right\|_1 \tag{5}$$

Если $\lambda = 0 \rightarrow \hat{w}^{lasso} = \hat{w}^{LeastSquares}$ Если $\lambda = \infty \rightarrow \hat{w}^{lasso} = 0$ Если λ между $\rightarrow 0 \leq \hat{w}^{lasso} \leq \hat{w}^{LeastSquares}$

Лассо выбирает параметры β с минимальным RSS, при условии, что l_1 норма параметров $(|\beta_1| + |\beta_2| + \cdots + |\beta_n|) \leq tolerance$. Ранее мы видели, что оптимальное решение задачи минимизации лассо находится в начале координат, и поэтому мы не можем вычислить градиент. Поэтому решением является выпуклый алгоритм оптимизации, называемый Координатным спуском. Этот алгоритм пытается минимизировать

$$\begin{gathered} f(w) = f(w_0, w_1, \cdots, w_n) \\ find \underset{min}{f}\ (w) \end{gathered} \tag{6}$$

Алгоритм координатного спуска можно описать следующим образом:

$$\begin{gathered} \text{Initialize } \hat{w} \\ \text{while not converged, pick a coordinate j} \\ \hat{w}_j \leftarrow \underset{min}{f}\ (\hat{w}_0.\hat{w}_1, \cdots, w, \hat{w}_{j+1}, \cdots, \hat{w}_n) \end{gathered} \tag{7}$$

Если мы выберем следующую координату случайным образом, это ста-

нет стохастическим координатным спуском. При координатном спуске нам не нужно выбирать размер шага.

Ниже приведен алгоритм спуска координат для гребневой регрессии, по одной координате за раз.

$$RSS(w) = \sum_{i=1}^{n}(y_i - \sum_{j=0}^{n} w_j h_j(x_i))^2$$

Зафиксируем все координаты w_{-j} и возьмем частную производную по w_j

$$\frac{\partial}{\partial w_j} = -2\sum_{i=1}^{n} h_j(x_i)(y_i - \sum_{j=0}^{n} w_j h_j(x_i))^2$$

$$= -2\sum_{i=1}^{n} h_j(x_i)(y_i - \sum_{k\neq j} w_k h_k(x_i) - w_j h_j(x_i))$$

$$= -2\sum_{i=1}^{m} h_j(x_i)(y_i - \sum_{k\neq j} w_k h_k(x_i)) + 2w_j \sum_{i=1}^{n} h_j(x_i)^2$$

Пояснения:

(i) по определению нормированных функций, $\sum_{i=1}^{n} h_j(x_i)^2 = 1$

(ii) мы будем обозначать$(\sum_{i=1}^{n} h_j(x_i)(y_i - \sum_{k\neq j} w_k h_k(x_i)) = \rho_j$

$$= -2\rho + 2w_j$$

приравняв частичные производные к 0, получим

$$w_j = \rho_j \tag{8}$$

Ниже приведен псевдокод спуска координат для регрессии лассо, по

одной координате за раз.

$$\begin{gathered}\text{Initialize } \hat{w} \\ \text{while not converged} \\ \text{for j in } 0, 1, 2, \cdots, n \\ \text{compute: } \rho_j = \sum_{i=1}^{n} h_j(x_i)(y_i ? \hat{y}_i(\hat{w}_j)) \\ \text{set: } w_j = \begin{cases} \rho_j + \frac{\lambda}{2} \text{ if } \rho_j < -\frac{\lambda}{2} \\ 0 \text{ if } \rho_j \text{ in } \left[-\frac{\lambda}{2}, \frac{\lambda}{2}\right] \\ \rho_j - \frac{\lambda}{2} \text{ if } \rho_j > \frac{\lambda}{2} \end{cases}\end{gathered} \tag{9}$$

Во **втором разделе** рассматриваются основные понятия машинного обучения. Приводится описание линейной модели.

Общая постановка задачи машинного обучения. Имеется множество объектов и множество ответов. Между ними существует некоторая неизвестная зависимость. Известно лишь конечное число пар "объект-ответ"составляющее обучающую выборку. По имеющимся данным следует построить алгоритм, который достаточно точно отображает неизвестную зависимость. То есть, способный для любых данных выдать ответ близкий к реальному.

Классы задач машинного обучения:

— обучение с учителем - восстановление зависимости по известным примерам и ответам.

— обучение без учителя - известно лишь множество объектов. Множество ответов отсутствует. Требуется, например, найти закономерности

Классические задачи машинного обучения:

— задача классификации. Множество объектов разделено некоторым образом на классы. Имеется обучающая выборка, содержащая пары "объект, класс". Требуется для произвольного объекта определить, к какому классу он мог бы принадлежать

— задачи кластеризации. Предназначены как для разработки типологии, так и для проверки гипотез на основе исследования данных.

— задача регрессии. И множество объектов, и множество ответов явля-

ются численными данными. Требуется по конечному числу имеющихся точек восстановить исходную зависимость

Одна из основных проблем машинного обучения - переобучение. Это явление, при котором для элементов обучающей выборки модель показывает результат, близкий к корректному, а для любого другого работает гораздо хуже. Это связано с тем, что при обучении модель обнаруживает в выборке случайные закономерности и в итоге "запоминает"все ответы.

Так как обучающая выборка конечна и неполна, а так же достоверно отличить случайные флуктуации от закономерностей невозможно, переобучение будет присутствовать практически всегда.

Один из факторов, способствующих переобучению, - чрезмерная сложность модели. Зачастую большое количество параметров поощряет подгонку под обучающее множество.

Методы борьбы с переобучением:

— перекрестная проверка - данные разбиваются на k частей. Обучение модели проходит на $k-1$, тестирование на одной.
— ранняя остановка - как только значение ошибки на тестовых данных начало превышать значение на обучающей выборке, прекращаем обучение
— увеличение количества обучающих данных - либо искусственно, специальным образом обрабатывая данные, либо путём сбора дополнительной информации
— ансамбли моделей - использование нескольких однотипных моделей параллельным или последовательным образом
— регуляризация

Регуляризация - добавление дополнительных условий к задаче с целью предотвратить переобучение.

В данной работе рассматривается применение L_1, L_2 - регуляризации, про которые было подробно рассказано в 1 главе. Кроме того существует техника регуляризации исключением.

В отличии от вышесказанных техник, регуляризация исключением не может применяться к линейным моделям и вместо изменения функции ошибки, будет меняться сама сеть.

Во время обучения нейросети часть её нейронов, за исключением входных и выходных, случайным образом временно удаляются. В результате чего получается изменённая сеть с меньшим числом нейронов. Далее берётся небольшое число примеров, на них происходит обучение сети. Обновляются соответствующие веса и смещения. Затем восстанавливаются удалённые нейроны и случайным образом выбирается новая группа для удаления. Одна из возможных реализаций этого алгоритма - вместо удаления фиксированного числа нейронов каждый раз, для каждого нейрона задать вероятность с которым он будет удаляться на новом шаге.

Получаем, когда удаляются разные нейроны, процесс обучения становится похож на обучение сразу нескольких различных нейросетей. И процедура исключения имеет схожий эффект с усреднением по большому числу нейросетей. Разные сети будут переобучаться по-разному и их общий результат может иметь более низкий уровень переобучения.

В **третьем разделе** находится описание алгоритма применения регуляризации при построении модели, показывающей диагнозы пациентов.

В алгоритме используется SGDClassifier. SGDClassifier - линейный классификатор из библиотеки sklearn. Реализует обучение с помощью стохастического градиентного спуска. Поддерживает L_1 и L_2 регуляризации.

1. Импортируем нужные библиотеки
2. Поиск в открытых источниках данных по диагнозам. В работе используются данные с платформы kaggle.
3. Предварительная подготовка данных - чтение всех симптомов и диагнозов. Для каждого диагноза набор симптомов формируется следующим образом: создаётся словарь, где ключом является симптом, а значением - 0 или 1 в зависимости от наличия симптома для соответствующего диагноза
4. Обучение модели без регуляризации и двух моделей с использованием регуляризации с помощью SGDClassifier
5. Сравнение полученных результатов

Обе модели, использующие регуляризацию, показали лучший результат чем модель, которая её не использует.

Основные результаты

1. Изучены основные понятия машинного обучения
2. Определены методы борьбы с переобучением
3. Рассмотрены основные виды регуляризации
4. Обучена модель с использованием и без использования регуляризации
5. В результате сравнения результатов обучения выяснилось, что регуляризация снижает уровень переобучения и точность модели повышается