Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

РЕФЕРАТ

на тему

МЕТРИКИ ДЛЯ ОЦЕНКИ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ. ВЫБОР ГИПЕРПАРАМЕТРОВ

Магистрант: А.С. Долматович

МИНСК 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Какие задачи чаще всего решаются с помощью машинного обучения? В первую очередь это регрессия, классификация и кластеризация.

Первые две — так называемое обучение с учителем: есть набор размеченных данных, на основе какого-то опыта нужно предсказать заданное значение.

Регрессия — это предсказание какого-то значения: например, на какую сумму купит клиент, какова износостойкость материала, сколько километров проедет автомобиль до первой поломки.

Кластеризация — это определение структуры данных с помощью выделения кластеров (например, категорий клиентов).

Метрики машинного обучения весьма специфичны и часто вводят в заблуждение, показывая хороший результат для плохих моделей. Для проверки моделей и их совершенствования нужно выбрать метрику, которая адекватно отражает качество модели, и способы её измерения. Обычно для оценки качества модели используют отдельный тестовый набор данных. И выбор правильной метрики — задача сложная.

1 Метрики качества регрессии

Общая идея: насколько хорошо вписывается в данные линия регрессии.

1.1 MAE/MAD

MAE/MAD (Mean Absolute Error, Mean Absolute Deviation) — средний модуль ошибки

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|e_t|$$

Средняя абсолютная ошибка (MAE) - это среднее вертикальное расстояние между каждой точкой и единичной линией или среднее горизонтальное расстояние между каждой точкой и идентификационной линией.

1.2 MSE/MSD

MSE/MSD (Mean Squared Error/Deviation) — среднеквадратическая ошибка

$$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n}e_t^2$$

Среднеквадратичное отклонение (MSD) измеряет среднее значение квадратов ошибок, то есть среднеквадратичную разницу между оцененными значениями и фактической стоимостью.

MSE — это функция риска, соответствующая ожидаемому значению квадрата потери ошибок. Тот факт, что MSE почти всегда является строго положительным (а не нулевым), объясняется

случайностью или тем, что оценщик не учитывает информацию, которая может дать более точную оценку.

1.3 RMSE/RMSD

RMSE/RMSD (Root Mean Squared Error) — лучше, чем MSE, потому что выражается в тех же единицах, что и измеряемая величина

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$$

RMSD представляет собой квадратный корень из второго момента выборки различий между предсказанными значениями и наблюдаемыми значениями. Среднеквадратичное отклонение служит для объединения величин ошибок в предсказаниях. RMSD - это показатель точности, позволяющий сравнивать ошибки прогнозирования разных моделей для конкретного набора данных, а не между наборами данных.

RMSD всегда неотрицательна, и значение 0 (практически никогда не достигается на практике) будет указывать на идеальное соответствие данных. В целом, более низкое RMSD лучше, чем более высокое. Однако сравнение данных разных типов будет недопустимым, поскольку мера зависит от масштаба используемых чисел.

RMSD - это квадратный корень из среднего квадрата ошибок. Влияние каждой ошибки на RMSD пропорционально размеру квадрата ошибки; таким образом, большие ошибки оказывают непропорционально большое влияние на RMSD. Следовательно, RMSD чувствительна к выбросам.

1.4 MAPE/MAPD

MAPE/MAPD (Mean Absolute Percentage Error) — ошибка в процентах от самой величины

$$\mathrm{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{e_t}{y_t} \right|$$

Средняя абсолютная процентная ошибка (MAPE) - это статистическая мера точности системы прогноза. Он измеряет эту точность в процентах и может быть рассчитан как средняя абсолютная процентная ошибка для каждого периода времени минус фактические значения, деленные на фактические значения.

2 Метрики качества классификации

Общая идея: насколько хорошо мы угадываем классы. Важно, что цена ошибки может быть разной для разных случаев.

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

При классификации на два класса фактически есть 4 различных исхода:

- True Positive (TP) — истинное значение “да”, предсказано “да”
- True Negative (TN) — истинное значение “нет”, предсказано “нет”
- False Positive (FP) — истинное значение “нет”, предсказано “да”. Ложное срабатывание, ошибка I рода.
- False Negatie (FN) — истинное значение “да”, предсказано “нет”. Пропуск цели, ошибка II рода.

2.1 Accuracy

Интуитивно понятной, очевидной и почти неиспользуемой метрикой является accuracy — доля правильных ответов алгоритма:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2.2 Precision, recall

Для оценки качества работы алгоритма на каждом из классов по отдельности введем метрики precision (точность) и recall (полнота).

Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными.

$$precision = \frac{TP}{TP + FP}$$

Recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

$$recall = \frac{TP}{TP + FN}$$

Именно введение precision не позволяет нам записывать все объекты в один класс, так как в этом случае мы получаем рост уровня False Positive. Recall демонстрирует способность алгоритма обнаруживать данный класс вообще, а precision — способность отличать этот класс от других классов.

Precision и recall не зависят, в отличие от accuracy, от соотношения классов и потому применимы в условиях несбалансированных выборок.

2.3 F-мера

Существует несколько различных способов объединить precision и recall в агрегированный критерий качества. F-мера (в общем случае — среднее гармоническое precision и recall):

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

β в данном случае определяет вес точности в метрике.

F-мера достигает максимума при полноте и точности, равными единице, и близка к нулю, если один из аргументов близок к нулю.

2.4 AUC-ROC и AUC-PR

При конвертации вещественного ответа алгоритма (как правило, вероятности принадлежности к классу) в бинарную метку, необходимо выбрать какой-либо порог, при котором 0 становится 1. Естественным и близким кажется порог, равный 0.5, но он не всегда оказывается оптимальным, например, при вышеупомянутом отсутствии баланса классов.

Одним из способов оценить модель в целом, не привязываясь к конкретному порогу, является AUC-ROC (или ROC AUC) — площадь (Area Under Curve) под кривой ошибок (Receiver Operating Characteristic curve). Данная кривая представляет из себя линию от (0,0) до (1,1) в координатах True Positive Rate (TPR) и False Positive Rate (FPR):

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

TPR — это полнота, а FPR показывает, какую долю из объектов negative класса алгоритм предсказал неверно. В идеальном случае, когда классификатор не делает ошибок (FPR = 0, TPR = 1) должна получится площадь под кривой, равная единице; в противном случае, когда классификатор случайно выдает вероятности классов, AUC-ROC будет стремиться к 0.5, так как классификатор будет выдавать одинаковое количество TP и FP (Рисунок 1).

Каждая точка на графике соответствует выбору некоторого порога. Площадь под кривой в данном случае показывает качество алгоритма (больше — лучше), кроме этого, важной является

крутизна самой кривой (максимизировать TPR, минимизируя FPR — значит, кривая в идеале должна стремиться к точке (0,1)).

Критерий AUC-ROC устойчив к несбалансированным классам и может быть интерпретирован как вероятность того, что случайно выбранный positive объект будет проранжирован классификатором выше (будет иметь более высокую вероятность быть positive), чем случайно выбранный negative объект.

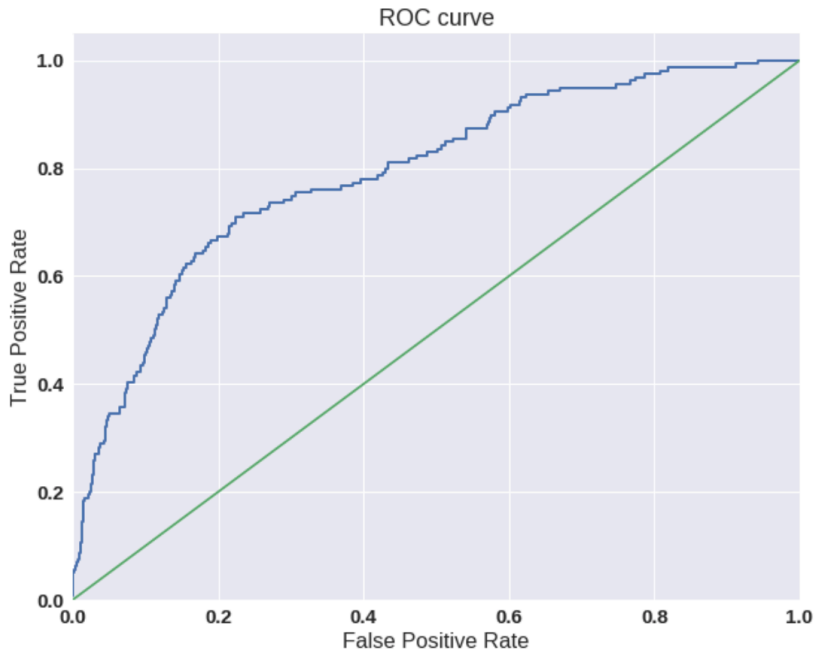

Рисунок 1 — ROC-кривая

2.5 Logistic Loss

Особняком стоит логистическая функция потерь, определяемая как:

$$logloss = -\frac{1}{l} \cdot \sum_{i=1}^{l} (y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i))$$

Здесь y’ — это ответ алгоритма на i-ом объекте, y — истинная метка класса на i-ом объекте, а l размер выборки.

Интуитивно можно представить минимизацию logloss как задачу максимизации accuracy путем штрафа за неверные предсказания. Однако необходимо отметить, что logloss крайне сильно штрафует за уверенность классификатора в неверном ответе.

3 Метрики качества кластеризации

Проблема оценки качества в задаче кластеризации трудноразрешима, как минимум, по двум причинам:

- Не существует оптимального алгоритма кластеризации.
- Многие алгоритмы кластеризации не способны определить настоящее количество кластеров в данных. Чаще всего количество кластеров подается на вход алгоритма и подбирается несколькими запусками алгоритма.

3.1 Внешние метрики оценки качества

Внешние метрики основаны на сравнении результата кластеризации с априори известным разделением на классы.

3.1.1 Rand Index

Индекс Rand оценивает, насколько много из тех пар элементов, которые находились в одном классе, и тех пар элементов, которые находились в разных классах, сохранили это состояние после кластеризации алгоритмом.

$$Rand = \frac{TP + FN}{TP + TN + FP + FN}$$

Имеет область определения от 0 до 1, где 1 — полное совпадение кластеров с заданными классами, а 0 — отсутствие совпадений.

3.1.2 Jaccard Index

Индекс Жаккара похож на Rand Index, только не учитывает пары элементов находящиеся в разные классах и разных кластерах.

$$Jaccard = \frac{TP}{TP + TN + FP}$$

Имеет область определения от 0 до 1, где 1 — полное совпадение кластеров с заданными классами, а 0 — отсутствие совпадений.

3.1.3 Folkes and Mallows Index

Индекс Fowlkes-Mallows используется для определения сходства между двумя кластерами.

$$FM = \sqrt{\frac{TP}{TP + TN} \cdot \frac{TP}{TP + FP}}$$

Более высокое значение индекса означает большее сходство между кластерами. Этот индекс также хорошо работает на зашумленных данных.

3.1.4 Purity

Чистота ставит в соответствие кластеру самый многочисленный в этом кластере класс.

$$P = \sum_i p_i (\max_j \frac{p_{ij}}{p_i})$$

Чистота находится в интервале [0, 1], причём значение = 1 отвечает оптимальной кластеризации.

3.2 Внутренние метрики оценки качества

Внутренние метрики отображают качество кластеризации только по информации в данных.

3.2.1 Компактность кластеров (Cluster Cohesion)

Идея данного метода в том, что чем ближе друг к другу находятся объекты внутри кластеров, тем лучше разделение.

$$WSS = \sum_{j=1}^{M} \sum_{i=1}^{|C_j|} (x_{ij} - \overline{x_j})^2, \text{ где } M - \text{количество кластеров.}$$

Таким образом, необходимо минимизировать внутриклассовое расстояние, например, сумму квадратов отклонений.

3.2.2 Отделимость кластеров (Cluster Separation)

В данном случае идея противоположная — чем дальше друг от друга находятся объекты разных кластеров, тем лучше.
Поэтому здесь стоит задача максимизации суммы квадратов отклонений:

$$BSS = n \cdot \sum_{j=1}^{M} (\overline{x_j} - \overline{x})^2, \text{ где } M - \text{количество кластеров.}$$

4 Понятие гиперпараметров модели

Гиперпараметры модели — параметры, значения которых задается до начала обучения модели и не изменяется в процессе обучения. У модели может не быть гиперпараметров.

Параметры модели — параметры, которые изменяются и оптимизируются в процессе обучения модели и итоговые значения этих параметров являются результатом обучения модели.

Примерами гиперпараметров могут служить количество слоев нейронной сети, а также количество нейронов на каждом слое. Примерами параметров могут служить веса ребер нейронной сети.

4.1 Подзадача оптимизации гиперпараметров

Оптимизация гиперпараметров — задача машинного обучения по выбору набора оптимальных гиперпараметров для обучающего алгоритма. Одни и те же виды моделей машинного обучения могут требовать различные предположения, веса или скорости обучения для различных видов данных. Эти параметры называются гиперпараметрами и их следует настраивать так, чтобы модель могла оптимально решить задачу обучения. Для этого находится кортеж гиперпараметров, который даёт оптимальную модель, оптимизирующую заданную функцию потерь на заданных независимых данных. Целевая функция берёт кортеж гиперпараметров и возвращает связанные с ними потери. Часто используется перекрёстная проверка для оценки этой обобщающей способности.

4.2 Виды алгоритмов оптимизации гиперпараметров

4.2.1 Поиск по решётке

Традиционным методом осуществления оптимизации гиперпарамеров является поиск по решётке (или вариация параметров), который просто делает полный перебор по заданному вручную подмножеству пространства гиперпараметров обучающего алгоритма. Поиск по решётке должен сопровождаться некоторым измерением

производительности, обычно измеряемой посредством перекрёстной проверки на тренировочном множестве, или прогонкой алгоритма на устоявшемся проверочном наборе.

Поскольку пространство параметров алгоритма машинного обучения для некоторых параметров может включать пространства с вещественными или неограниченными значениями, вручную установить границу и дискретизацию может оказаться необходимым до применения поиска по решётке.

4.2.2 Случайный поиск

Случайный поиск заменяет полный перебор всех комбинаций на выборку их случайным образом. Это можно легко применить к дискретным установкам, приведённым выше, но метод может быть также обобщен к непрерывным и смешанным пространствам. Случайный поиск может превзойти поиск по решётке, особенно в случае, если только малое число гиперпараметров оказывает влияние на производительность алгоритма обучения машины. В этом случае говорят, что задача оптимизации имеет низкую внутреннюю размерность. Случайный поиск также легко параллелизуем и, кроме того, позволяют использовать предварительные данные путём указания распределения для выборки случайных параметров.

4.2.3 Байесовская оптимизация

Байесовская оптимизация — это метод глобальной оптимизации для неизвестной функции (чёрного ящика) с шумом. Применённая к гиперпараметрической оптимизации, байесовская оптимизация строит стохастическую модель функции отображения из значений гиперпараметра в целевую функцию, применённую на множестве проверки. Путём итеративного применения перспективной конфигурации гиперпараметров, основанной на текущей модели, а затем её обновления, байесовская оптимизация стремится собрать как можно больше информации об этой функции и, в частности, место оптимума. Метод пытается сбалансировать зондирование (гиперпараметры, для которых изменение наименее

достоверно известно) и использование (гиперпараметры, которые, как ожидается, наиболее близки к оптимуму). На практике байесовская оптимизация показала. Лучшие результаты с меньшими вычислениями по сравнению с поиском по решётке и случайным поиском ввиду возможности суждения о качестве экспериментов ещё до их выполнения.

4.2.4 Оптимизация на основе градиентов

Для конкретных алгоритмов обучения можно вычислить градиент гиперпараметров и оптимизировать их с помощью градиентного спуска.

Первое использование этих техник фокусировалось на нейронных сетях. Затем эти методы были распространены на другие модели, такие как методы опорных векторов или логистическая регрессия.

Другой подход использования градиентов гиперпараметров состоит в дифференцировании шагов алгоритма итеративной оптимизации с помощью автоматического дифференцирования

Эволюционная оптимизация — это методология для глобальной оптимизации неизвестных функций с шумом. При оптимизации гиперпараметров эволюционная оптимизация использует эволюционные алгоритмы для поиска гиперпараметров для данного алгоритма.

ЗАКЛЮЧЕНИЕ

Приведенные метрики относятся только к оценке качества алгоритма машинного обучения уже после его обучения. Дело в том, что нельзя использовать критерий оценки качества и при решении оптимизационной задачи в процессе обучения — такой способ на обучающей выборке безусловно даст хорошие показатели качества, однако на тестовой выборке данные метрики, скорее всего, дадут низкие показатели качества. Чтобы честно оценить качество работы алгоритма машинного обучения, критерий его оценки должен быть независим от оптимизационной задачи.

Но если имеется необходимость оценить модель, то по возможности, стоит использовать одну и ту же метрику для обучения и оценки качества модели.

Выбор метрики нужно делать с фокусом на предметную область, предварительно обрабатывая данные и, возможно, сегментируя их.

Необходимо использовать устойчивую метрику (например, медиана устойчивее арифметического среднего) или фильтровать выбросы, поскольку они могут сильно исказить результат метрики.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

HABR [Электронный ресурс]. - Электронные данные. - Режим доступа: https://habr.com/ru/company/ods/blog/328372/ - Дата доступа: 04.11.2019.

HABR [Электронный ресурс]. - Электронные данные. - Режим доступа: https://habr.com/ru/company/jetinfosystems/blog/420261/ - Дата доступа: 05.11.2019.

WIKIPEDIA [Электронный ресурс]. - Электронные данные. - Режим доступа: https://en.wikipedia.org/wiki/Mean_squared_error - Дата доступа: 07.11.2019.

WIKIPEDIA [Электронный ресурс]. - Электронные данные. - Режим доступа: https://en.wikipedia.org/wiki/Mean_absolute_error - Дата доступа: 07.11.2019.

WIKIPEDIA [Электронный ресурс]. - Электронные данные. - Режим доступа: https://en.wikipedia.org/wiki/Root-mean-square_deviation - Дата доступа: 07.11.2019.

BIOINFORMATICSINSTITUTE [Электронный ресурс]. - Электронные данные. - Режим доступа: http://bioinformaticsinstitute.ru/sites/default/files/vvedenie_v_mashinnoe_obuchenie.pdf - Дата доступа: 09.11.2019.

IFMO [Электронный ресурс]. - Электронные данные. - Режим доступа: http://neerc.ifmo.ru/wiki/index.php?title=Оценка_качества_в_задаче_кластеризации - Дата доступа: 12.11.2019.