03.04.20

Ensemble learning and random forest
Ансамблевое обучение и случайные леса

Ансамбль – группа прогнозаторов

```
VotingClassifier(estimators=[(…,…),()],voting=['hard, soft'])
```

Бэггинг и вставка

`BaggingClassifier`:

`BaggingRegressor`:

Bagging (bootstrap aggregation) - этот подход предусматривает использование для каждого прогнозатора одного и того же алгоритма обучения, но обучение прогнозаторов на разных случайных поднаборах обучающего набора. Когда выборка осуществляется с заменой, такой метод называется бэггингом.

Вставка (pasting) – выборка выполняется без замены.

После того, как все прогнозаторы обучены, ансамбль может вырабатывать прогноз для нового образца, просто агрегируя прогнозы всех прогнозаторов.

Обычно совокупный результат состоит в том, что ансамбль имеет похожее смещение, но меньшую дисперсию, чем одиночный прогнозатор, обученный на исходном обучающем наборе.

В целом бэггинг часто приводит к лучшим моделям, что и является причиной, по которой ему обычно отдают предпочтение.

Random forest

RandomForestClassifier:

RandomForestRegressor:

Случайный лес (random forest) - это ансамбль деревьев принятия решений, которые обычно обучены посредством метода бэггинга (либо иногда вставки), как правило, с параметром max_ samples, установленным в размер обучающего набора.

Случайные леса очень удобны для быстрого понимания того, какие признаки действительно имеют значение, в особенности, если вам необходимо осуществлять выбор признаков.

Бустинг

Бустинг (первоначально называемый усилением гипотезы (hypothesis boosting)) относится к любому ансамблевому методу, который способен комбинировать нескольких слабых учеников в одного сильного ученика. Основная идея большинства методов бустинга предусматривает последовательное обучение прогнозаторов, причем каждый из них старается исправить своего предшественника.

Доступно много методов бустинга, но безоговорочно самыми популярными являются **AdaBoost** (сокращение от Adaptive Boosting - адаптивный бустинг) и **градиентный бустинг** (gradient boosting).

AdaBoost

AdaBoostClassifier

AdaBoostRegressor

Один из способов, которым новый прогнозатор может исправлять своего предшественника, заключается в том, что он уделяет чуть больше внимания обучающим образцам, на которых у предшественника было недообучение. В результате новые прогнозаторы все больше и больше концентрируются на трудных случаях. Именно такой прием применяет метод AdaBoost

У этого приема последовательного обучения есть один важный недостаток: он не допускает распараллеливания (или только частично), поскольку каждый прогнозатор можно обучать лишь после того, как был обучен и оценен предыдущий прогнозатор. В результате он не масштабируется настолько хорошо, как бэггинг или вставка.

Градиентный бустинг

GradientBoostingRegressor

GradientBoostingClassifier

Подобно AdaBoost градиентный бустинг работает, последовательно добавляя в ансамбль прогнозаторы, каждый из которых корректирует своего предшественника. Тем не менее, вместо подстройки весов образцов на каждой итерации, как делает AdaBoost, этот метод старается подогнать новый прогнозатор к остаточным ошибкам (residual error), допущенным предыдущим прогнозатором.

Стекинг

Он основан на простой идее: вместо того, чтобы использовать тривиальные функции (такие как с жестким голосованием) для агрегирования прогнозов всех прогнозаторов в ансамбле, почему бы нам ни научить какую-то модель делать это агрегирование? На рис. 7.12 демонстрируется ансамбль такого рода, выполняющий задачу регрессии на новом образце.

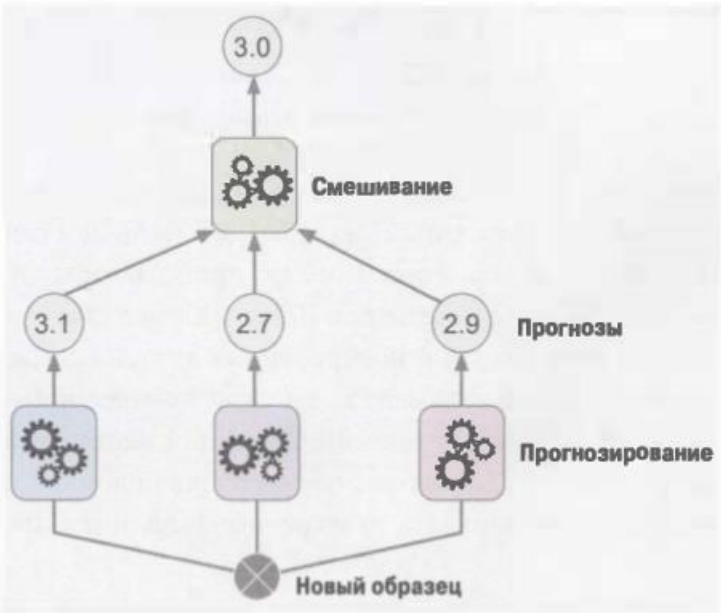

Рис. 7.12. Агрегирование прогнозов с применением смешивающего прогнозатора

финальный прогнозатор (называемый смесителем (blender) или мета-учеником (meta learner)) получает на входе такие прогнозы и вырабатывает окончательный прогноз (3.0).

Ответы на вопросы

1. **Если вы обучили пять разных моделей на точно тех же обучающих данных, и все они достигают точности 95%, то можно ли как-нибудь скомбинировать эти модели, чтобы получить лучшие результаты? Если да, то как? Если нет, то почему?**

Если вы обучили пять разных моделей, и все они достигают точности 95%, то можете попытаться скомбинировать их в ансамбль с голосованием, который часто будет давать даже лучшие результаты. Он лучше работает, если модели сильно отличаются (например, классификатор SVM, классификатор на базе дерева принятия решений, классификатор на основе логистической регрессии и т.д.). Еще лучше, когда модели обучаются на разных обучающих образцах (в этом заключается весь смысл ансамблей с бэггингом и вставкой), но если это не так, то ансамбль попрежнему будет работать при условии, что модели отличаются.

2. **В чем разница между классификаторами с жестким и с мягким голосованием?**

Классификатор с жестким голосованием просто подсчитывает голоса каждого классификатора в ансамбле и выбирает класс, получивший большинство голосов. Классификатор с мягким голосованием вычисляет среднюю оценочную вероятность для каждого класса и выбирает класс с наивысшей вероятностью. Это придает голосам с высоким доверием больший вес и часто работает лучше, но только в случае, если каждый классификатор способен оценивать вероятности классов (например, для классификатора SVM потребуется установить `probability=True`)

3. **Можно ли ускорить обучение ансамбля с бэггингом, распределив его по множеству серверов? Как насчет ансамблей с вставкой, ансамблей с бустингом, случайных лесов или ансамблей со стекингом?**

Обучение ансамбля с бэггингом вполне можно ускорить, распределив его между множеством серверов, т.к. каждый прогнозатор в ансамбле не зависит от остальных. То же самое касается ансамблей с вставкой и случайных лесов, по той же причине. Тем не менее, каждый прогнозатор в ансамбле с бустингом построен на основе предыдущего прогнозатора, поэтому обучение обязано быть последовательным, и вы не получите какой-либо выгоды, распределив обучение между несколькими серверами. Что касается ансамблей со стекингом, то все

прогнозаторы в заданном слое не зависят друг от друга, а потому их можно обучать параллельно на множестве серверов. Однако прогнозаторы в одном слое могут обучаться только после того, как обучены все прогнозаторы из предыдущего слоя.

4. **В чем преимущество оценки с помощью неиспользуемых образцов (oob)?**

При оценке с помощью неиспользуемых образцов каждый прогнозатор в ансамбле с бэггингом оценивается с применением образцов, на которых он не обучался (они удерживались в стороне). Это позволяет получить достаточно объективную оценку ансамбля без необходимости в наличии дополнительного проверочного набора. Таким образом, для обучения доступно больше образцов и ансамбль может работать чуть лучше.

5. **Что делает особо случайные деревья (Extra-Trees) в большей степени случайными, чем обыкновенные случайные леса? Чем может помочь такая добавочная случайность? Особо случайные деревья медленнее или быстрее обыкновенных случайных лесов?**

При выращивании дерева в случайном лесе для каждого узла, подлежащего расщеплению, рассматривается только случайный поднабор признаков. Это справедливо также для особо случайных деревьев, но они продвигаются на шаг дальше: вместо поиска наилучшего возможного порога, как поступает обыкновенное дерево принятия решений, они используют для каждого признака случайные пороги. Такая дополнительная случайность действует подобно форме регуляризации: если случайный лес переобучается обучающими данными, то особо случайные деревья могут работать лучше. Кроме того, поскольку особо случайные деревья не ищут наилучшие возможные пороги, в обучении они гораздо быстрее, чем случайные леса. Тем не менее, при выработке прогнозов особо случайные деревья ни быстрее, ни медленнее случайных деревьев.

6. **Если ваш ансамбль AdaBoost недообучается на обучающих данных, то какие гиперпараметры вы должны подстраивать и каким образом?**

Если ваш ансамбль AdaBoost недообучается на обучающих данных, тогда можете попробовать увеличить количество оценщиков или уменьшить гиперпараметры регуляризации базового оценщика. Можно также попробовать слегка повысить скорость обучения.

7. **Если ваш ансамбль с градиентным бустингом переобучается обучающим набором, то вы должны увеличить или уменьшить скорость обучения?**

Если ваш ансамбль с градиентным бустингом переобучается обучающим набором, то вы должны попробовать уменьшить скорость обучения. Вы также можете воспользоваться ранним прекращением, чтобы найти правильное количество прогнозаторов.