

Введение в искусственный интеллект. Машинное обучение

Тема: Кросс-валидация. Дилемма смещения-разброса

Бабин Д.Н., Иванов И.Е.

кафедра Математической Теории Интеллектуальных Систем



1 Тестирование моделей, кросс-валидация

- 1 Тестирование моделей, кросс-валидация
- 2 Декомпозиция ошибки, недообучение и переобучение

Как понять, что одна модель лучше другой?

Для этого используют независимое от **обучающего** множества множество, которое называется **ТЕСТОВЫМ**

Как понять, что одна модель лучше другой?

Для этого используют независимое от **обучающего** множества множество, которое называется **тестовым**

Зачем вообще это понимать?

- Существует множество алгоритмов машинного обучения и важно понимать, какой из них более применим в конкретной задаче
- Даже в рамках одной модели может существовать множество параметров

Как выбирать лучшую модель

Наивный подход

Обучить модели с различными параметрами и выбрать лучшую на тесте

Как выбирать лучшую модель

Наивный подход

Обучить модели с различными параметрами и выбрать лучшую на тесте

Минусы наивного подхода

- Так как тест обычно состоит из случайной подвыборки исходной выборки, то результат на тесте тоже является некоторым приближением случайной величины
- Если все модели тестировать на тестовом датасете и таким образом выбирать лучшую, то будет происходить неявное обучение на тесте, а на другом независимом тесте возможны сюрпризы

Как выбирать лучшую модель

Наивный подход

Обучить модели с различными параметрами и выбрать лучшую на тесте

Минусы наивного подхода

- Так как тест обычно состоит из случайной подвыборки исходной выборки, то результат на тесте тоже является некоторым приближением случайной величины
- Если все модели тестировать на тестовом датасете и таким образом выбирать лучшую, то будет происходить неявное обучение на тесте, а на другом независимом тесте возможны сюрпризы

Что же делать?

Чтобы неявно не обучиться на тестовых данных — надо использовать кросс-валидацию, или скользящий контроль (cross-validation)

Общая идея

Основная идея кросс-валидации состоит в разбиении обучающего множества на два непересекающихся множества (возможно многократном):

$$X^{learn} = X^{train} \sqcup X^{val}$$

На одном из них происходит обучение, а на другом происходит валидация модели.

Общая идея

Основная идея кросс-валидации состоит в разбиении обучающего множества на два непересекающихся множества (возможно многократном):

$$X^{learn} = X^{train} \sqcup X^{val}$$

На одном из них происходит обучение, а на другом происходит валидация модели.

Зачем нужна валидация?

Обычно любой алгоритм машинного обучения содержит целый набор т.н.

“**гиперпараметров**” (т.е. параметров, которые не обучаются, а задаются изначально): размерность, различные весовые коэффициенты и т.п.

И для того, чтобы подбирать эти параметры “по-честному”, не используя вообще тестовые данные, и проводят процедуру валидации.

Частные случаи

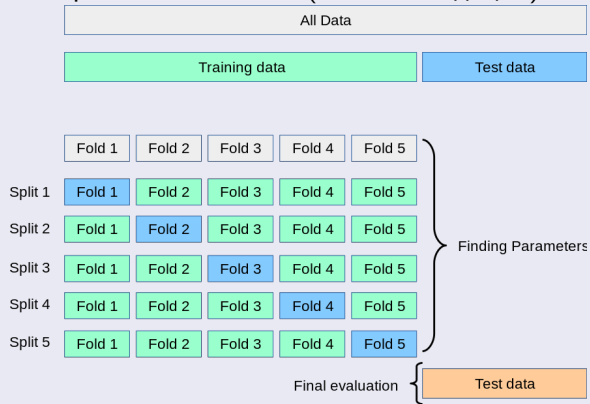
- 1 Простейшая кросс-валидация — это контроль на отложенном множестве (hold-out), при котором происходит однократное разделение множества:

Train

Validation

Частные случаи

2 Контроль по k блокам (k-fold валидация)¹:



¹Image source: <https://scikit-learn.org/>

Частные случаи

- 3 Контроль по отдельным объектам (leave-one-out, или LOO валидация) — частный случай k -fold валидации, если k равно мощности обучающего множества

Частные случаи

- ③ Контроль по отдельным объектам (leave-one-out, или LOO валидация) — частный случай k -fold валидации, если k равно мощности обучающего множества
- ④ Многократная k -fold валидация — повторение k -fold валидации несколько раз с разными разбиениями.



Определение

Переобучение (overfitting) — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке.

Переобучение возникает при использовании избыточно сложной модели

Определение

Переобучение (overfitting) — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке.

Переобучение возникает при использовании избыточно сложной модели

Одна из основных причин возникновения

Избыточная размерность пространства параметров модели, “лишние” степени свободы используются для точной настройки на обучающую выборку

Переобучение

Определение

Переобучение (overfitting) — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке.

Переобучение возникает при использовании избыточно сложной модели

Одна из основных причин возникновения

Избыточная размерность пространства параметров модели, “лишние” степени свободы используются для точной настройки на обучающую выборку

Один из основных методов обнаружения

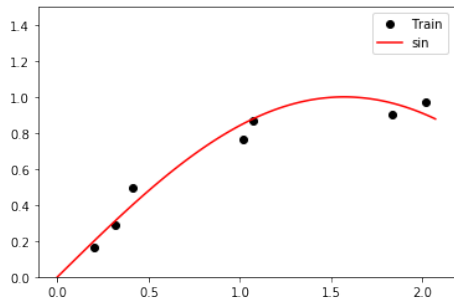
Использование кросс-валидации

Определение

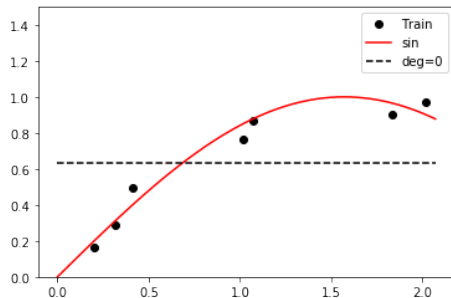
Недообучение (underfitting) – нежелательное явление, возникающее при решении задач обучения по прецедентам, когда алгоритм обучения не обеспечивает достаточно малой величины средней ошибки на обучающей выборке.

Недообучение возникает при использовании недостаточно сложных моделей

Примеры недообучения и переобучения

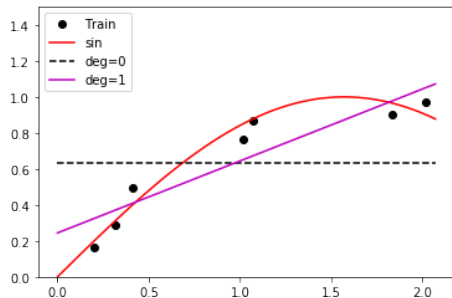


Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели

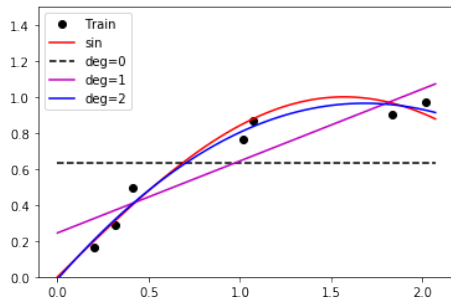
Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность

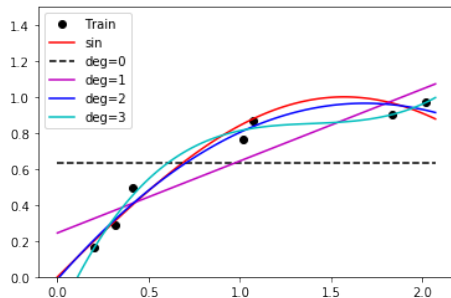


Примеры недообучения и переобучения



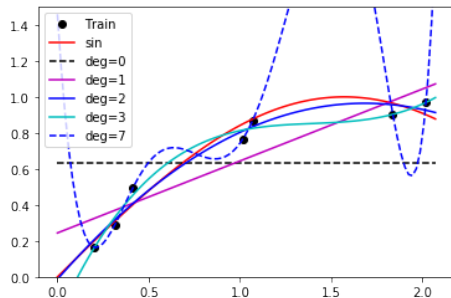
- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность
- Полиномы высоких степеней могут в точности пройти через точки обучающей выборки

Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность
- Полиномы высоких степеней могут в точности пройти через точки обучающей выборки

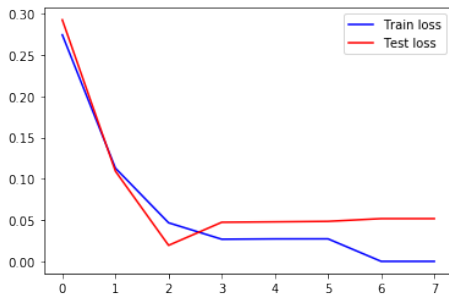
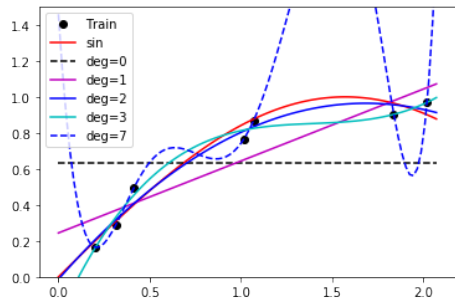
Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность
- Полиномы высоких степеней могут в точности пройти через точки обучающей выборки



Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность
- Полиномы высоких степеней могут в точности пройти через точки обучающей выборки

О параметрах и гиперпараметрах

В примере с приближением неизвестной зависимости полиномом

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0:$$

- **Параметры:** коэффициенты $a_n, a_{n-1}, \dots, a_1, a_0$, и они настраиваются во время обучения модели
- **Гиперпараметры:** степень многочлена n , которая выбирается до начала обучения и затем выбирается из множества проверенных на валидационном множестве



Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.



Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Полагаем, что ε и a — независимые ($Ea\varepsilon = EaE\varepsilon$). $Ey = Ef$, $Dy = D\varepsilon = \sigma^2$.

Разложение квадрата ошибки

$$E(y - a)^2 = E(y^2 + a^2 - 2ya) = Ey^2 + Ea^2 - 2Eya =$$

Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Полагаем, что ε и a — независимые ($Ea\varepsilon = EaE\varepsilon$). $Ey = Ef$, $Dy = D\varepsilon = \sigma^2$.

Разложение квадрата ошибки

$$\begin{aligned} E(y - a)^2 &= E(y^2 + a^2 - 2ya) = Ey^2 + Ea^2 - 2Eya = \\ &= Ey^2 + Ea^2 - 2E(f + \varepsilon)a = Ey^2 + Ea^2 - 2Efa - 2E\varepsilon a = \end{aligned}$$



Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Полагаем, что ε и a — независимые ($Ea\varepsilon = EaE\varepsilon$). $Ey = Ef$, $Dy = D\varepsilon = \sigma^2$.

Разложение квадрата ошибки

$$\begin{aligned} E(y - a)^2 &= E(y^2 + a^2 - 2ya) = Ey^2 + Ea^2 - 2Eya = \\ &= Ey^2 + Ea^2 - 2E(f + \varepsilon)a = Ey^2 + Ea^2 - 2Efa - 2E\varepsilon a = \\ &= Ey^2 - (Ey)^2 + (Ey)^2 + Ea^2 - (Ea)^2 + (Ea)^2 - 2fEa = \end{aligned}$$

Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Полагаем, что ε и a — независимые ($Ea\varepsilon = EaE\varepsilon$). $Ey = Ef$, $Dy = D\varepsilon = \sigma^2$.

Разложение квадрата ошибки

$$\begin{aligned} E(y - a)^2 &= E(y^2 + a^2 - 2ya) = Ey^2 + Ea^2 - 2Eya = \\ &= Ey^2 + Ea^2 - 2E(f + \varepsilon)a = Ey^2 + Ea^2 - 2Efa - 2E\varepsilon a = \\ &= Ey^2 - (Ey)^2 + (Ey)^2 + Ea^2 - (Ea)^2 + (Ea)^2 - 2fEa = \\ &= Dy + Da + (Ey)^2 + (Ea)^2 - 2fEa = \end{aligned}$$

Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Полагаем, что ε и a — независимые ($Ea\varepsilon = EaE\varepsilon$). $Ey = Ef$, $Dy = D\varepsilon = \sigma^2$.

Разложение квадрата ошибки

$$\begin{aligned} E(y - a)^2 &= E(y^2 + a^2 - 2ya) = Ey^2 + Ea^2 - 2Eya = \\ &= Ey^2 + Ea^2 - 2E(f + \varepsilon)a = Ey^2 + Ea^2 - 2Efa - 2E\varepsilon a = \\ &= Ey^2 - (Ey)^2 + (Ey)^2 + Ea^2 - (Ea)^2 + (Ea)^2 - 2fEa = \\ &= Dy + Da + (Ey)^2 + (Ea)^2 - 2fEa = \\ &= Dy + Da + (Ef)^2 - 2fEa + (Ea)^2 = \end{aligned}$$

Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Полагаем, что ε и a — независимые ($Ea\varepsilon = EaE\varepsilon$). $Ey = Ef$, $Dy = D\varepsilon = \sigma^2$.

Разложение квадрата ошибки

$$\begin{aligned} E(y - a)^2 &= E(y^2 + a^2 - 2ya) = Ey^2 + Ea^2 - 2Eya = \\ &= Ey^2 + Ea^2 - 2E(f + \varepsilon)a = Ey^2 + Ea^2 - 2Efa - 2E\varepsilon a = \\ &= Ey^2 - (Ey)^2 + (Ey)^2 + Ea^2 - (Ea)^2 + (Ea)^2 - 2fEa = \\ &= Dy + Da + (Ey)^2 + (Ea)^2 - 2fEa = \\ &= Dy + Da + (Ef)^2 - 2fEa + (Ea)^2 = \\ &= Dy + Da + (E(f - a))^2 = \sigma^2 + \text{variance}(a) + \text{bias}^2(f, a) \end{aligned}$$

Определение

Разброс (variance) — дисперсия ответов алгоритмов $a(x)$.

Характеризует разнообразие алгоритмов (из-за случайности обучающей выборки, шума, стохастичности обучения и т.д.)



Дополнительные определения

Определение

Разброс (variance) — дисперсия ответов алгоритмов $a(x)$.

Характеризует разнообразие алгоритмов (из-за случайности обучающей выборки, шума, стохастичности обучения и т.д.)

Определение

Смещение (bias) — матожидание разности между истинным ответом и выбранным алгоритмом.

В примере выше — это $E(f - a)$.

Характеризует способность модели настраиваться на целевую зависимость

Дополнительные определения

Определение

Разброс (variance) — дисперсия ответов алгоритмов $a(x)$.

Характеризует разнообразие алгоритмов (из-за случайности обучающей выборки, шума, стохастичности обучения и т.д.)

Определение

Смещение (bias) — матожидание разности между истинным ответом и выбранным алгоритмом.

В примере выше — это $E(f - a)$.

Характеризует способность модели настраиваться на целевую зависимость

Определение

Разложение квадрата ошибки в примере выше называется **дилеммой смещения-разброса** (bias-variance tradeoff)

Модель оптимальной сложности: классический взгляд

- Для простых моделей характерно недообучение



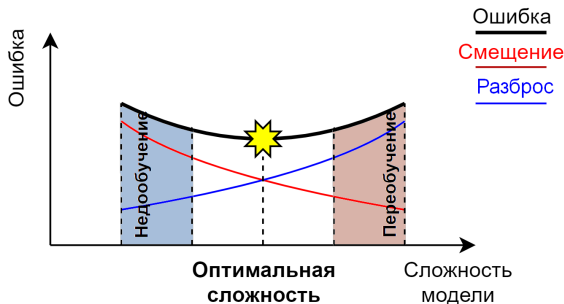
Модель оптимальной сложности: классический взгляд

- Для простых моделей характерно недообучение
- Для сложных моделей характерно переобучение



Модель оптимальной сложности: классический взгляд

- Для простых моделей характерно недообучение
- Для сложных моделей характерно переобучение
- Оптимальная сложность модели где-то между



Модель оптимальной сложности: современные эмпирические данные

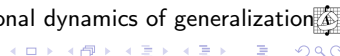
- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности

²Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky. "High-dimensional dynamics of generalization error in neural networks." 2017

Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности
- С развитием техники стало возможным обучать модели с миллионами и даже миллиардами параметров

²Advani, Madhu S., Andrew M. Saxe, and Haim Sompolsky. "High-dimensional dynamics of generalization error in neural networks." 2017



Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности
- С развитием техники стало возможным обучать модели с миллионами и даже миллиардами параметров
- Оказалось, что с увеличением сложности ошибка ведет себя сначала как предсказывает дилемма смещения-разброса, а затем неожиданно начинает снижаться² и выходит даже на меньший уровень ошибки!

²Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky. "High-dimensional dynamics of generalization error in neural networks." 2017

Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности
- С развитием техники стало возможным обучать модели с миллионами и даже миллиардами параметров
- Оказалось, что с увеличением сложности ошибка ведет себя сначала как предсказывает дилемма смещения-разброса, а затем неожиданно начинает снижаться² и выходит даже на меньший уровень ошибки!
- Переломный момент — точка, в которой сложность модели сопоставима с мощностью обучающей выборки (interpolation threshold)

²Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky. “High-dimensional dynamics of generalization error in neural networks.” 2017

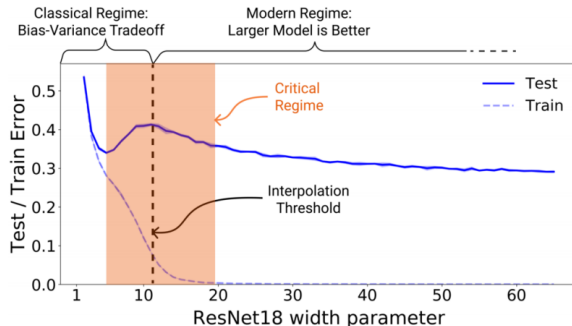
Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности
- С развитием техники стало возможным обучать модели с миллионами и даже миллиардами параметров
- Оказалось, что с увеличением сложности ошибка ведет себя сначала как предсказывает дилемма смещения-разброса, а затем неожиданно начинает снижаться² и выходит даже на меньший уровень ошибки!
- Переломный момент — точка, в которой сложность модели сопоставима с мощностью обучающей выборки (interpolation threshold)
- Такое поведение называется **двойным спуском** (double descent)

²Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky. “High-dimensional dynamics of generalization error in neural networks.” 2017

Модель оптимальной сложности: двойной спуск

- Пример двойного спуска на практике³:



³Image source: <https://arxiv.org/pdf/1912.02292.pdf>

- 1 Нужно сразу делить имеющиеся данные на обучающую, валидационную и тестовую выборки

- 1 Нужно сразу делить имеющиеся данные на обучающую, валидационную и тестовую выборки
- 2 Необходимо следить за сложностью модели — слишком большая приведет к переобучению, слишком маленькая — к недообучению.

- 1 Нужно сразу делить имеющиеся данные на обучающую, валидационную и тестовую выборки
- 2 Необходимо следить за сложностью модели — слишком большая приведет к переобучению, слишком маленькая — к недообучению.
 - И то, и другое приведет к увеличению ошибки на тесте

- 1 Нужно сразу делить имеющиеся данные на обучающую, валидационную и тестовую выборки
- 2 Необходимо следить за сложностью модели — слишком большая приведет к переобучению, слишком маленькая — к недообучению.
 - И то, и другое приведет к увеличению ошибки на тесте
- 3 В случае огромного количества данных и параметров (миллиарды) классические оценки перестают работать





Спасибо за внимание!