1. Объект воспринимаем как строку матрицы X (l x d). **Объектом** (x_i, y_i) будем считать вектор длины d+1, у которого есть d значений отвечающих за его признаки и 1 значение, отвечающее за его целевое значение. За **признак** считаем переменную, значение которой, как правило, можно наблюдать. **Целевой переменной** назовём переменную, значение которой мы хотим научиться предсказывать. Функция потерь -- это отображение L(a(x_i), y_i): YxY -> R, где a(x_i) -- наше предсказание, y_i -- истинное целевое значение объекта. **Функция потерь** призвана показывать, насколько мы ошиблись на конкретном объекте. **Модель** -- это отображение a(x_i) **X** -> **Y**, которое сопоставляет набору признаков предсказание целевого значения. **Функционал ошибки** -- это, как правило, взвешенная сумма функций ошибок по всей обучающей выборке. **Обучение** -- минимизация функционала ошибки.
2. Признаки бывают количественными и категориальными. Количественный признак может принимать произвольное вещественное значение, категориальный принимает значения из конечного множества.
3. **Недообучение** - ситуация при которой либо модель слишком проста, либо её параметры далеки от оптимальных. Как следствие, недообученная модель обычно будет показывать низкие показатели качества как на обучающей, так и на тестовой выборке. **Переобучение** - ситуация при которой модель слишком сильно настраивается на обучающую выборку, тем самым, теряя свою обобщающую способность. Переобученная модель зачастую показывает аномально высокие показатели качества на обучающей выборке и низкие показатели качества на тестовой выборке.
4. **Кросс-валидация** - техника обучения(?), при которой мы несколько раз случайным образом разбиваем выборку на тестовую и обучающую, после чего усредняем полученное качество. Используется для того, чтобы избежать переобучения. Лучше тем, что всевозможные смещения усредняются и выметаются.
5. **Параметры** подбираются по ходу обучения модели. **Гиперпараметры** либо задаются извне, либо перебираются по сетке и не могут быть подобраны на обучении.

Продолжение на след. странице.

№6.

$a(x_i) = \langle w, x_i \rangle$

$x_i^{(1)} \equiv 1$, чтобы учесть сдвиг.

$$\overset{MSE}{Q}(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\langle w, x_i \rangle - y_i \right)^2$$

$$\overset{MSE}{Q}(w) = (Xw - y)^T (Xw - y)$$

№7.

$$f: \mathbb{R}^n \to \mathbb{R} \quad \nabla_v f(v_0) = \left(\frac{\partial f}{\partial v_1}(v_0), \dots, \frac{\partial f}{\partial v_n}(v_0) \right)$$

$$f: \mathbb{R}^{n \times m} \to \mathbb{R} \quad \nabla_A f(A) = \left(\frac{\partial f(A)}{\partial a_{ij}} \right)_{i,j=1}^{n,m}$$

• Запишем производную по направлению вектора $h \in \mathbb{R}^n$

$$\frac{\partial f(\bar{x}_0)}{\partial h} = \left(\frac{\partial f(\bar{x}_0)}{\partial x_1} \cdot h_1, \dots, \frac{\partial f(\bar{x}_0)}{\partial x_n} \cdot h_n \right) = \langle \nabla_x f(\bar{x}_0), h \rangle =$$

$$= |\nabla_x f(\bar{x})| \cdot |h| \cdot \cos\varphi \to \max \Rightarrow \cos\varphi \to 1 \Rightarrow \varphi \to 0$$

$\Rightarrow h$ сонаправлен с $\nabla_x f(\bar{x})$

$\Rightarrow$ Наибольшее значение производной по градиенту направлению.

№8.

$$w^{(k)} = w^{(k-1)} - \eta \cdot \nabla_w f(w^{(k-1)}), \quad \eta \text{ - размер шага.}$$

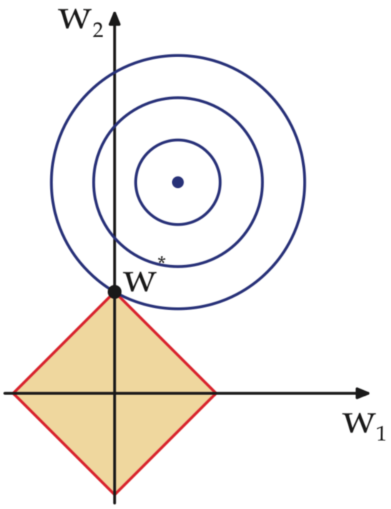
В обычном случае градиент считается по всей выборке.

В SGD: мы случайно выбираем элемент (объект) на каждой итерации, чтобы <u>экономить время.</u>

9. Данные нормируются, чтобы не испытывать проблем при значительном отличии масштаба признаков. В противном случае, градиент может проскакивать оптимальные значения. Способы нормировки:

(1) **стандартизация:** $x_{ij} \to \dfrac{x_{ij} - \mu_{ij}}{\sigma_{ij}}, \mu_{ij} = \dfrac{1}{n}\sum_{i=1}^{n} x_{ij},\ \sigma_{ij} = \dfrac{1}{n}\sum_{i=1}^{n}(x_{ij} - \mu_{ji})^2$

(2) на отрезок: $x_{ij} \to \dfrac{x_{ij} - \min\limits_i(x_{ij})}{\max\limits_i(x_{ij}) - \min\limits_i(x_{ij})}$



10. **Регуляризация** -- это добавление к функционалу Q(w) регуляризатора R(w), который призван сдерживать размер получающихся весов. В линейных моделях регуляризацию используют для того, чтобы бороться с переобучением, так как аномально большие веса -- симптом переобучения.

$R_p(w) = \left(\sum_{j=1}^{d} |w_j|^p\right)^{1/p}$ Для L1, L2 параметр p = 1, 2 соответственно.

L1 регуляризация отбирает признаки так как задача оптимизации функционала с регуляризатором сводится к задаче условной оптимизации. В случае с L1 условие получается "угловатым", поэтому велика вероятность углового касания линии уровня функционала. При угловом касании зануляется часть переменных.

Продолжение на след. странице.

№11.

Бинарный случай: $a(x) = \langle w, x \rangle$

$M_i = y_i \langle w, x_i \rangle$ – маржин (отступ)

$$Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} [y_i \langle w, x_i \rangle < 0]$$

- Знак отступа говорит о правильности предсказания.
- Значение отступа – о "расстоянии" от разделяющей поверхности.

$[y_i \langle w, x_i \rangle < 0]$ – не дифференцируема

$\Rightarrow L(M_i) \le \tilde{L}(M_i)$ – верхняя оценка $\left(\log(1+e^{-M_i}),\ e^{-M_i}, \dots\right)$

$\Rightarrow$ Обучение: $\min\limits_w Q(w) \Rightarrow \min\limits_w \frac{1}{\ell} \sum\limits_{i=1}^{\ell} \tilde{L}(w, x_i)$.

№12.

Лог. регрессия возвращает вероятность принадлежности объекта к классу $p(y_i = +1 | x_i)$.

Пусть $b(x)$ – наш алгоритм, тогда $b(x) \in [0,1]$

$\mathbb{P}\{\text{встретим } (x_i, y_i)\} = b(x)^{[y_i = +1]} \cdot (1 - b(x))^{[y_i = -1]}$

Правдоподобие выборки: $Q(b, X) = \prod\limits_{i=1}^{\ell} b(x_i)^{[y_i = +1]} (1 - b(x_i))^{[y_i = -1]} \to \max\limits_b$

$$-\log(+Q) = -\sum_{i=1}^{\ell} \left([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i)) \right) \to \min_b$$

$$\mathbb{E}[L | x] = -p(y = +1 | x) \log b - (1 - p(y = +1 | x)) \log(1 - b)$$

$$\frac{\partial \mathbb{E}[L | x]}{\partial b} = -\frac{p}{b} + \frac{1 - p}{1 - b} = 0 \Rightarrow b^* = p$$

$b(x) = \sigma(\langle w, x \rangle)$, где $\sigma(z) = \frac{1}{1 + \exp(-z)} \Rightarrow Q = \sum \log(1 + \exp(-y_i \langle w, x \rangle))$

$$b(x_i) = \sigma(\langle w, x_i \rangle), \quad \sigma(z) = \frac{1}{1+\exp(-z)}$$

$$\Rightarrow p(y_i = +1 \mid x_i) = \frac{1}{1+\exp(-\langle w, x_i \rangle)}$$; Выразим $\langle w, x_i \rangle$:

$$1+\exp(-\langle w, x_i \rangle) = \frac{1}{p} \qquad \exp = \frac{1}{p} - 1 \quad ; \; -\langle w, x_i \rangle = \log\left(\frac{1-p}{p}\right)$$

$$\langle w, x_i \rangle = \log\left(\frac{p}{1-p}\right)$$

$$-\sum_{i=1}^{l}\Big([y_i = +1] \log b(x) + [y_i = -1] \log(1 - b(x))\Big) =$$

$$= -\sum_{i=1}^{l}\left([y_i = +1] \log \frac{1}{1+\exp(-\langle w, x_i \rangle)} + [y_i = -1] \log \left(\underbrace{1 - \frac{1}{1+\exp(-\langle w, z \rangle)}}_{\frac{\exp(-\langle w, z \rangle)}{1+\exp(-\langle w, z \rangle)}}\right)\right) =$$

$$= -\sum_{i=1}^{l}\left([y_i = +1] \log \frac{1}{1+\exp(-\langle w, x_i \rangle)} + [y_i = -1] \log \frac{1}{1+\exp(+\langle w, z \rangle)}\right) =$$

$$= \sum_{i=1}^{l}\Big([y_i = +1] \log(1+\exp(-\langle w, x_i \rangle)) + [y_i = -1] \log(1+\exp(\langle w, x_i \rangle))\Big) =$$

$$= \sum_{i=1}^{l} \log(1+\exp(-y_i \langle w, x_i \rangle))$$ — лог. функц. потерь.

Продолжение на след. странице.

13.

	$y = 1$	$y = -1$
$a(x) = 1$	True Positive (TP)	False Positive (FP)
$a(x) = -1$	False negative (FN)	True Negative (TN)

Таблица 1. Матрица ошибок

precision (точность) = TP/(TP + FP) -- доля по горизонтали.
recall (полнота) = TPR = TP/(TP + FN) -- доля по вертикали.
F-мера = 2*precision*recall/(precision + recall)

14. Алгоритм построения ROC-curve: берём классификатор b(x), упорядочиваем объекты по убыванию. Обходим последовательность "сверху вниз". Каждый раз, когда охватываем один положительный объект шагаем вверх на 1/l+, минус -- на 1/l- вправо. Если охватили одновременно несколько объектов, то отмеряем новую координату и соединяем точки наклонной кривой. Пройдя все возможные l + 1 объект мы попадём в точку (1, 1)
ROC-curve строится в координатах (fpr, tpr), рассчитываемых как вертикальные доли матрицы ошибок.

AUC = площадь под ROC-curve.

15. Считаем, что объект относится к одному из K классов.
One-vs-all: обучаем K классификаторов $b_1(x), b_2(x), \dots, b_K(x)$.
Классификатор $b_i(x)$ обучаем на выборке $(x_j,\ 2[y_j = i] - 1)_{j=1}^{l}$.
Наш алгоритм примет вид: $a(x) = \arg\max\ \ b_i(x)$

All-vs-all: обучаем C_K^2 классификаторов.
Классификатор $a_{ij}(x)$ обучаем на подвыборке $X_{ij} = \{(x, y) \in X \,|\, [y = i]\ or\ [y = j]\}$
Такой классификатор умеет бинарно отделять объект между i, j.
Далее финальный алгоритм строим через голосование: $a(x) = \arg\max \sum_{i=1}^{K} \sum_{j \neq i} [a_{ij}(x) = k]$

16. **Пересекающиеся классы** = объект может относиться к нескольким классам сразу.
Тогда ответы выборки примут вид $Y = \{0,\ 1\}^{lxK}$.
Самый простой подход: предполагаем, что классы независимы. Обучаем K классификаторов и тогда объект i примет вид: $(a_1(x_i), a_2(x_i), \dots, a_K(x_i))$

17. Для каждого класса k построим матрицу ошибок: TP_k, TN_k, FP_k, FN_k.
Микро-усреднение: вычисляем $\overline{TP} = \dfrac{1}{K} \sum TP_k$ и прочие средние элементы матрицы ошибок, после чего считаем метрики качества: precision, recall, ... от этих средних значений.

Макро-усреднение: Для каждого класса считаем его метрики качества: precision_k, recall_k, ... и после усредняем именно метрики качества.

Какое именно усреднение выбирать зависит от того, хотим ли мы учитывать масштабы разных классов при оценивании или нет.

18. Решающее дерево -- это граф-дерево, в каждой терминальной вершине которого записан прогноз: один из классов. Во всех вершинах кроме терминальных записаны предикаты: функции-условия, которые определяют по какому ребру конкретный объект будет передвигаться. Объект стартует из корневой вершины и спускается по ребрам до терминальной, где и получает свой прогноз. $a(x) = \sum c_i [x \in J_i]$, где $\sqcup_i J_i = X$, а c_i -- предсказываемые значения. Процесс ограничивается критерием останова.

19. В корневой вершине разбиваем все объекты на
$R_1 = \{x \in X \,|\, x_j < t\},\ R_2 = \{x \in X \,|\, x_j \geqslant t\}$, подбирая оптимальные **j, t** с точки зрения критерия информативности $Q(R, j, t) = H(R) - \frac{|R_1|}{|R|}H(R_1) - \frac{|R_2|}{|R|}H(R_2)$, где на месте H(R) может стоять дисперсия, энтропия или что-то такое.

20. Критерий информативности должен возрастать по разнообразию данных в множестве: чем более однородным получился набор данных, тем ниже H(R). Критерий информативности используется в жадном алгоритме и принимает участие в функционале ошибки в каждой вершине.

21. Пусть p_i -- доля класса i. Тогда $G(R) = \sum_{i=1}^{K} p_i(1 - p_i)$ -- критерий Джини.
$E(R) = \sum_{i=1}^{K} p_i \log_2(\frac{1}{p_i})$ -- энтропийный критерий.

22. Для начала берем функцию расстояния, которая обязана быть только симметричной и неотрицательной. Затем берём либо функцию весов w(i, u, x^(i)), либо ядро $K\Big(\frac{\rho(u, x^{(i)})}{h}\Big)$, где u -- объект, для которого строим предсказание, x^(i), какой-то сосед.
Затем упорядочиваем соседей по неубыванию расстояния:
$\rho(u, x^{(1)}) \leqslant \ldots \leqslant \rho(u, x^{(k)}) \leqslant \ldots \leqslant \rho(u, x^{(l)})$
Индексы иксов выбраны как раз из упорядочивания.
Тогда д**ля классификации** предсказание примет вид:
$a(u) = \arg\max_{y \in Y} \sum_{i=1}^{k} w(i, u, x^{(i)})[y^{(i)} = y],\ w() = \frac{k + 1 - i}{k}$
Весовая функция может быть другой и вообще, там может стоять функция-ядро.

Для регрессии: $a(u) = \arg\min_{c \in \mathbb{R}} \sum_{i=1}^{k} K\Big(\frac{\rho(u, x^{(i)})}{h}\Big)(c - y^{(i)})^2$

23. Неустойчивость к шуму при миньковских метриках на p > 1. Проклятие размерности.

24. **Расстояние Минковского:** $\rho_p = \Big(\sum_{i=1}^{n} |x_i - y_i|^p\Big)^{\frac{1}{p}}$, p=1 для Манхэттенского, p=2, для Евклидова. Если устремить p к бесконечности, то легко показать, что:
$\lim_{p \to \infty} \rho_p(x, y) = \max_{i=1,\ldots,n} |x_i - y_i|$, что называется **расстоянием Чебышёва.** Теперь выведем **косинусную меру:** $< x, y > = |x| \cdot |y| \cos\theta \Rightarrow \theta = \arccos\Big(\frac{< x, y >}{|x| \cdot |y|}\Big)$
Все вышеперечисленные расстояния работают на вещественных векторах. Здесь нам важно то, что мы работаем с евклидовым пространством. Однако, если наши объекты представляют собой множества, то можно прибегнуть к **расстоянию Джаккарда:**
$\rho_J(A, B) = 1 - \frac{|A \cap B|}{A \cup B}$

25. **Композиция (ансамбль)** a_N(x) = f(b_1(x), ... , b_N(x)) -- Пусть мы хотим построить композицию из N алгоритмов. Тогда сгенерируем N выборок **бутстрапом.** То есть равномерно выберем по n элементов с возвращением. На всех выборках обучим N алгоритмов: b_1(x), ... , b_N(x), тогда финальный алгоритм будет, например:
a(x) = 1/n (b_1(x) + ... + b_N(x)).

Предположим теперь, что ошибки алгоритма некореллированы и имеют нулевое матожидание. Посчитаем теперь среднюю ошибку для b_1(x), ... и a(x):

$$Error_1 = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\epsilon_i^2,\ Error_N = \mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N}\epsilon_i^2\right) = Error_1/N$$

26. **Бэггинг - это метод обучения**: Строим n случайных подвыборок {X_n, Y_n} из X, Y бутстрапом (то есть сэмплированием с повторениями). Обучаем на них n алгоритмов b_i. Итоговый алгоритм берём как средневзвешанное.

В **случайном лесе** мы генерируем случайные выборки бутстрэпом и в каждом алгоритме смотрим на случайное подмножество признаков, после чего усредняем результат.

В случайном лесе каждое дерево обучается по своему подмножеству объектов X_i, Y_i. Остальные объекты представляют собой контрольную выборку и на них можно будет посчитать ошибку, она и будет представлять собой Out-of_bag error (OOB):

$$OOB_i = \sum_{j=1}^{l} L\left(y_j, \frac{1}{\sum_{j=1}^{l}[x_i \notin X_i]}\sum_{j=1}^{l}[x_j \notin X_i]b_n(x_j)\right)$$

Это простой с точки зрения вычислений способ оценить ошибку нашего алгоритма.

Продолжение на след. странице.

№27. Бустинг: (простой)

$b_1(x) := \arg\min\limits_{b \in A} \frac{1}{2}\sum\limits_{i=1}^{\ell}(b(x_i) - y_i)^2$, где A – семейство алгоритмов.

$S_i^{(1)} := y_i - b_1(x_i) \Rightarrow b_2(x) = \arg\min\limits_{b \in A} \frac{1}{\ell}\sum\limits_{i=1}^{\ell}(b(x_i) - S_i^{(1)})^2$

По индукции: $S_i^{(N)} = y_i - \sum\limits_{n=1}^{N-1} b_n(x_i) = y_i - a_{N-1}(x)$

Тогда: $a_N = \sum\limits_{n=1}^{N} b_n$. – итоговый алгоритм.

• Градиентный бустинг: $b_0(x)$ – какой-то очень простой алгоритм.

Допустим обучили $a_{N-1}(x) = \sum\limits_{n=0}^{N-1} \gamma_n b_n(x)$, тогда:

γ_N, b_N так: $\min\limits_{b_N, \gamma_N} \sum\limits_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + \gamma_N b_N(x_i))$ $(*)$

Теперь разберёмся, как решать $(*)$.

• Рассмотрим вспомогательную задачу: $L(y_i, a_{N-1}(x) + S_i) \xrightarrow[S_1, \dots, S_\ell]{} \min$

• Логично, что $-\left.\frac{\partial L}{\partial z}\right|_{z = a_{N-1}(x_i)} = S_i$, чтобы ходить на наискорейшее убывание.

Тогда $\boxed{\bar{S} = -\nabla_z \sum\limits_{i=1}^{L} L(y_i, z_i)\Big|_{z_i = a_{N-1}(x_i)}}$

Теперь подберём b_N, приближающий точки $\{S_i\}$: $b_N = \arg\min\limits_{b \in A} \sum\limits_{i=1}^{L}(b(x_i) - S_i)^2$

Finally: $\gamma_N = \arg\min\limits_{\gamma \in \mathbb{R}} \sum\limits_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) - \gamma_N b_N(x_i))$

• Сдвиг: S_i.

28. Сокращение шага в градиентном бустинге -- это способ регуляризации модели, решающий проблему того, что бустинг довольно рано выходит на асимптоту ошибки, а следовательно, рано начинает настраиваться на шум и переобучаться. Решение заключается в подборе параметра η:

$$a_N(x) = a_{N-1}(x) - \eta\gamma_N b_N(x)$$

№29. метод обучения $\mu : (\mathbb{X} \times \mathbb{Y})^\ell \to \mathcal{A}$

Пусть $L(\mu) = \mathbb{E}_X \left[\mathbb{E}_{x,y} \left[(y - \mu(X)(x))^2 \right] \right] =$

Усреднение по всем выборкам.

$$= \int_{(\mathbb{X}\times\mathbb{Y})^\ell} \int_{\mathbb{X}\times\mathbb{Y}} (y - \mu(X)(x))^2 p(x,y) \prod_{i=1}^{\ell} p(x_i, y_i)\, dx\, dy\, dx_1 dy_1 \dots dx_\ell dy_\ell$$

Лемма: $\mathbb{E}_{x,y}[y - \mu(X)(x)]^2 = \mathbb{E}_{x,y}[y - \mathbb{E}(y|x)]^2 + \mathbb{E}_{x,y}[\mathbb{E}(y|x) - \mu(X)(x)]^2$

$\triangleleft$ $(y - a(x))^2 = (y - \mathbb{E}(y|x) + \mathbb{E}(y|x) - a(x))^2 =$

$= (y - \mathbb{E}(y|x))^2 + 2(y - \mathbb{E}(y|x))(\mathbb{E}(y|x) - a(x)) + (\mathbb{E}(y|x) - a(x))^2$

Возьмём $\mathbb{E}_{x,y}$, чтобы получить функционал ср.кв. риска.

Посмотрим на ~~третье~~ второе слагаемое: $\cancel{\mathbb{E}_{x,y}(\mathbb{E}(y|x) - a(x))^2} =$

$= \mathbb{E}_x \mathbb{E}_y \Big[2(y - \mathbb{E}(y|x)) \underbrace{(\mathbb{E}(y|x) - a(x))}_{\text{не зависит от } y} \Big] \Big| x = 2\mathbb{E}_x \Big[(\mathbb{E}(y|x) - a(x)) \underbrace{\mathbb{E}_y(y - \mathbb{E}(y|x))}_{|x} \Big] =$

$= 0$ $\qquad [\mathbb{E}(y|x) - \mathbb{E}(y|x)] = 0$

$\Rightarrow \mathbb{E}_{x,y} = \mathbb{E}_{x,y}[y - \mathbb{E}(y|x)]^2 + \mathbb{E}_{x,y}[\mathbb{E}(y|x) - \underbrace{a(x)}_{\mu(X)(x)}]^2$ $\triangleright$

Продолжение на след. странице.

Тогда $L(\mu) = \mathbb{E}_X\left[\mathbb{E}_{x,y}\left[y - \mu(X)(x)\right]^2\right] =$

$$= \cancel{\mathbb{E}_X}\, \underbrace{\mathbb{E}_{x,y}\left[y - \mathbb{E}(y|x)\right]^2}_{\text{не зависит от } X} + \mathbb{E}_X \underbrace{\mathbb{E}_{x,y}}_{\text{можно поменять местами.}}\left[\mathbb{E}(y|x) - \mu(X)(x)\right]^2$$

.Посмотрим на второе слагаемое:

$$\mathbb{E}_{x,y}\mathbb{E}_X\left[\mathbb{E}(y|x) - \mathbb{E}_X(\mu(X)(x)) + \mathbb{E}_X(\mu(X)(x)) - \mu(X)(x)\right]^2 =$$

$$= \mathbb{E}_{x,y}\mathbb{E}_X\underbrace{\left[\mathbb{E}(y|x) - \mathbb{E}_X(\mu(X)(x))\right]^2}_{\text{не зависит от } X.} + \mathbb{E}_{x,y}\mathbb{E}_X\left[\mathbb{E}_X(\mu(X)(x)) - \mu(X)(x)\right]^2 +$$

$$+ 2\,\mathbb{E}_{x,y}\mathbb{E}_X\left[\left(\mathbb{E}(y|x) - \mathbb{E}_X(\mu(X)(x))\right)\underbrace{\left(\mathbb{E}_X(\mu(X)(x)) - \mu(X)(x)\right)}_{\mathbb{E}_X \text{ от этой части даёт } 0.}\right] =$$

$$= \mathbb{E}_{x,y}\left[\mathbb{E}(y|x) - \mathbb{E}_X\mu(X)(x)\right]^2 + \mathbb{E}_{x,y}\mathbb{E}_X\left[\mathbb{E}_X(\mu(X)(x)) - \mu(X)(x)\right]^2$$

Подставим в $L(\mu)$, получим:

$$L(\mu) = \underbrace{\mathbb{E}_{x,y}\left[y - \mathbb{E}(y|x)\right]^2}_{\text{шум}} + \underbrace{\mathbb{E}_{x,y}\left[\mathbb{E}(y|x) - \mathbb{E}_X\mu(X)\right]^2}_{\substack{\text{смещение} \\ \text{от } a^*(x) = \mathbb{E}(y|x)}} + \underbrace{\mathbb{E}_{x,y}\mathbb{E}_X\left[\mathbb{E}_X\mu(X) - \mu(X)\right]^2}_{\text{разброс, т.к. } \mathrm{Var}(\mu).}$$

Наконец, получили искомое разложение.

30. Переобучение "=" разброс. Смещение -- насколько сложную зависимость мы способны восстановить. *Время для охуительных рассуждений.*

	Смещение	**Разброс**
Линейные модели	Большое, так как они тупые	Маленький
Решающие деревья	Маленькое, так как они умные и сложные	Большой, горе от ума
Бэггинг	Матож выборочного среднего равен матож любого слагаемого, поэтому смещение не меняется	Падает по числу алгоритмов, включенных в бэггинг
Бустинг	Падает, так как мы пошагово "умнеем" .	Не меньше, чем для одного базового алгоритма

31. Бэггинг -- способ понижения разброса. В вопросе 25 вывели, что разброс будет усредняться. Смещение же никак не будет меняться.

При бустинге смещение будет понижаться, а разброс может вырасти из-за того, что мы слишком настраиваемся на шумы.