Оценка классификации

Классификация мер эффективности

- Пусть **D** - набор для тестирования, содержащий *n* точек в *d*-мерном пространстве, пусть {*c*1, *c*2, ..., c_k} обозначает набор из *k* меток классов, а *M* - классификатор. Для $\mathbf{x}_i \in \mathbf{D}$ пусть y_i обозначает его истинный класс, а $\hat{y}_i$ = M ($\mathbf{x}_i$) обозначает его предсказанный класс.

Частота ошибок

- Частота ошибок - это доля неверных прогнозов для классификатора на тестовом наборе, определяемая как

$$Error\ Rate\ = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i) \qquad (22.1)$$

- где *I* - индикаторная функция, которая равна 1, когда ее аргумент истинен, и 0 в противном случае. Частота ошибок - это оценка вероятности ошибочной классификации. Чем ниже частота ошибок, тем лучше классификатор.

Доля правильных ответов

- Доля правильных ответов классификатора - это доля правильных прогнозов на тестовом наборе:

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i) = 1 - \ Error\ Rate \quad (22.2)$$

- Данная оценка дает оценку вероятности правильного прогноза; таким образом, чем выше оценка, тем лучше классификатор.

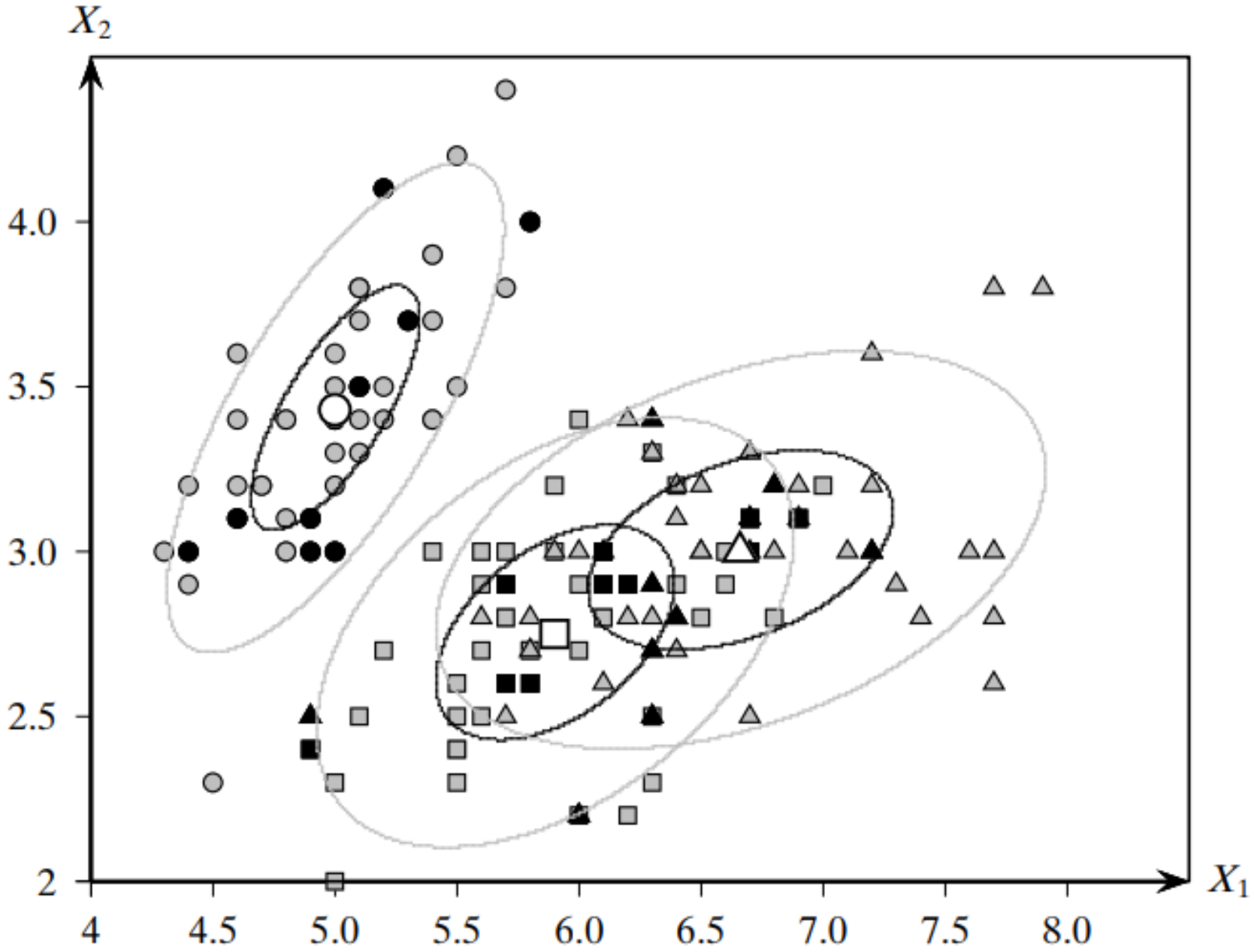


Рисунок 22.1. Набор данных Iris: три класса.

Меры основанные на таблице сопряженности

- Пусть $\mathcal{D} = \{\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_k\}$ обозначает разбиение точек тестирования на основе их истинных меток классов, где

$$\mathbf{D}_j = \{\mathbf{x}_i \in \mathbf{D} \mid y_i = c_j\}$$

- Пусть n_i = | $\mathbf{D}_i$ | обозначает размер истинного класса c_i. Пусть $\mathcal{R} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_k\}$ обозначает разбиение тестовых точек на основе предсказанных меток, то есть

$$\mathbf{R}_j = \{\mathbf{x}_i \in \mathbf{D} \mid \hat{y}_i = c_j\}$$

- Пусть m_j = | $\mathbf{R}_j$ | обозначают размер предсказанного класса c_j. $\mathcal{R}$ и $\mathcal{D}$ создают таблицу сопряженности **N** размером $k \times k$, также называемую *матрицей ошибок*, определяемую следующим образом:

$$\mathbf{N}(i,j) = n_{ij} = |\mathbf{R}_i \cap \mathbf{D}_j| = |\{\mathbf{x}_a \in \mathbf{D} \mid \hat{y}_a = c_i \text{ and } y_a = c_j\}|$$

- где $1 \le i, j \le k$. Счетчик n_{ij} обозначает количество точек с предсказанным классом c_i, истинная метка которого - c_j.

Доля правильных ответов / Точность

- Зависящая от класса *доля правильных ответов* или *точность* классификатора *M* для класса c_i задается как доля правильных прогнозов по всем точкам, которые, как предполагается, находятся в классе c_i.

$$acc_i = \text{prec}_i = \frac{n_{ii}}{m_i}$$

- где m_i - количество примеров, предсказанных как c_i классификатором *M*. Чем выше доля правильных ответов класса c_i, тем лучше классификатор.
- Общая точность или доля правильных ответов классификатора - это средневзвешенное значение доли правильных ответов для конкретного класса:

$$Accuracy = Precision = \sum_{i=1}^{k} \left(\frac{m_i}{n}\right) acc_i = \frac{1}{n}\sum_{i=1}^{k} n_{ii}$$

Покрытие/Полнота

- Зависящее от класса *покрытие* или *полнота M* для класса c_i - это доля правильных прогнозов по всем точкам в классе c_i:

$$coverage_i = recall_i = \frac{n_{ii}}{n_i}$$

- где n_i - количество точек в классе c_i. Чем выше покрытие, тем лучше классификатор.

F-мера

- *Специфическая для класса F-мера* пытается сбалансировать значения точности и полноты, вычисляя их среднее гармоническое значение для класса c_i:

$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 \cdot \text{prec}_i \cdot \text{recall}_i}{\text{prec}_i + \text{recall}_i} = \frac{2n_{ii}}{n_i + m_i}$$

- Чем выше значение F_i, тем лучше классификатор.
- Общая *F-мера* для классификатора *M* - это среднее значение конкретных классов:

$$F = \frac{1}{k} \sum_{i=1}^{r} F_i$$

- Для идеального классификатора максимальное значение F-меры равно 1.

Бинарная классификация: положительный и отрицательный класс

- *Истинно положительные результаты*: количество точек, которые классификатор правильно предсказывает как положительные:

$$TP = n_{11} = |\{\mathbf{x}_i \mid \hat{y}_i = y_i = c_1\}|$$

- *Ложно положительные*: количество точек, которые классификатор предсказывает как положительные, которые на самом деле относятся к отрицательному классу:

$$FP = n_{12} = |\{\mathbf{x}_i \mid \hat{y}_i = c_1 \text{ and } y_i = c_2\}|$$

- *Ложно отрицательные*: количество точек, которые классификатор предсказывает как отрицательные, который на самом деле принадлежит к положительному классу:

$$FN = n_{21} = |\{\mathbf{x}_i \mid \hat{y}_i = c_2 \text{ and } y_i = c_1\}|$$

- *Истинно отрицательные*: количество точек, которые классификатор правильно считает отрицательными:

$$TN = n_{22} = |\{\mathbf{x}_i \mid \hat{y}_i = y_i = c_2\}|$$

Частота ошибок

- Частота ошибок [уравнение (22.1)] для случая двоичной классификации дается как доля ошибок (или ложных предсказаний):

$$Error\ Rate\ = \frac{FP + FN}{n}$$

Доля правильных ответов

- Доля правильных ответов [уравнение (22.2)] - доля верных прогнозов:

$$Accuracy\ = \frac{TP + TN}{n}$$

- Выше приведены общие показатели эффективности классификатора. Мы можем получить меры, характерные для конкретных классов, следующим образом.

Точность, зависящая от класса

- Точность для положительного и отрицательного классов задается как

$$\text{prec}_P = \frac{TP}{TP + FP} = \frac{TP}{m_1}$$

$$\text{prec}_N = \frac{TN}{TN + FN} = \frac{TN}{m_2}$$

- где $m_i = |\mathbf{R}_i|$ - количество точек, прогнозируемых M как имеющих класс c_i.

Чувствительность: истинно положительный результат

- Истинно положительный показатель, также называемый *чувствительностью*, это доля правильных прогнозов по всем точкам в положительном классе, то есть это просто полнота для положительного класса

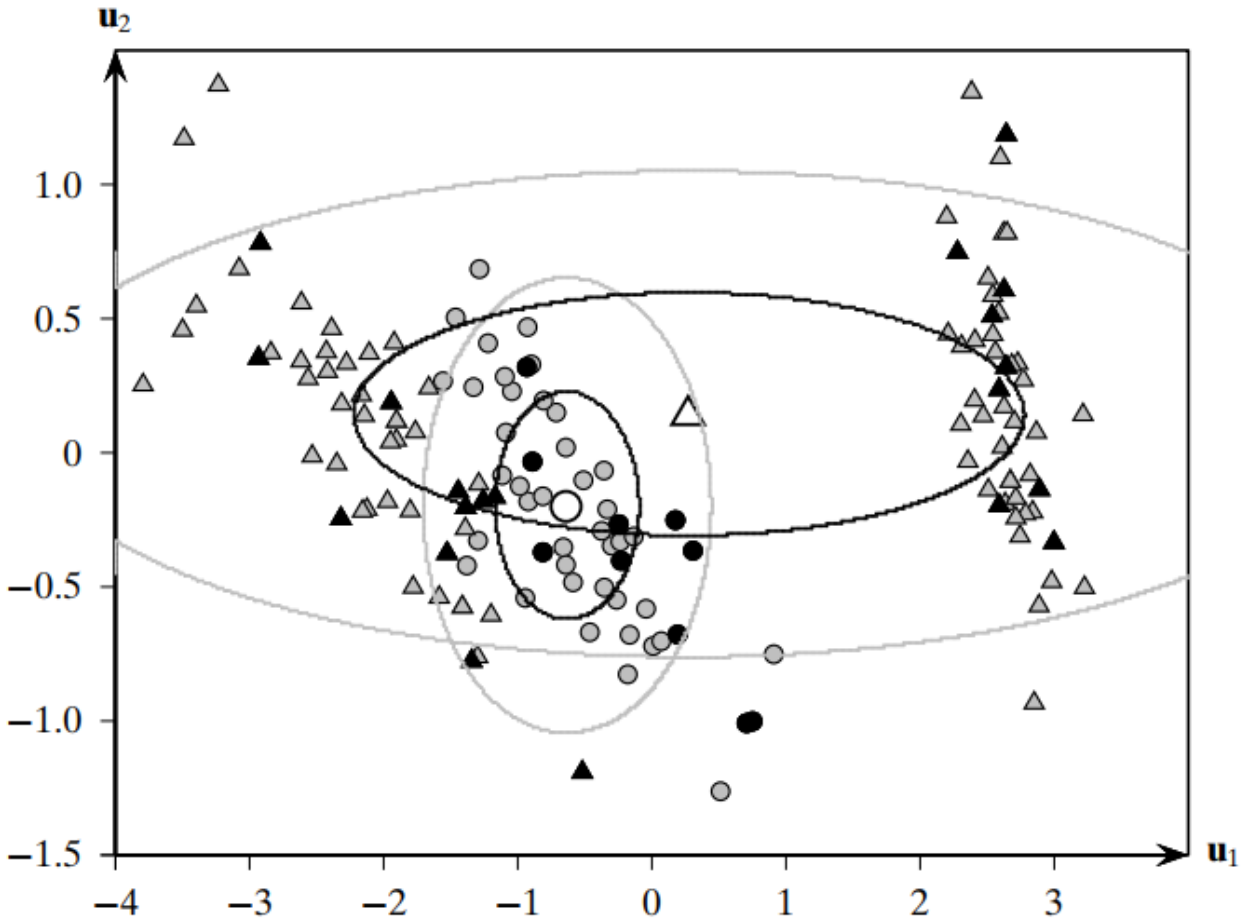
$$TPR = recall_P = \frac{TP}{TP + FN} = \frac{TP}{n_1}$$

- где n_1 - размер положительного класса.

Ложно отрицательный результат

- Ложно отрицательный результат определяется как

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{n_2} = 1 - \ specificity$$



Главные компоненты набора данных Iris: тренировочный и тестировочный наборы

ROC анализ

- Рабочая характеристика приемника (англ. receiver operating characteristic - ROC) - это популярная стратегия для оценки производительности классификаторов, в случаях с двумя классами. ROC-анализ требует, чтобы классификатор выводил значение оценки для положительного класса для каждой точки в тестовом наборе.
- ROC-анализ строит график производительности классификатора по всем возможным значениям порогового параметра ρ. В частности, для каждого значения ρ он отображает частоту ложно положительных результатов (специфичность 1) на оси абсцисс в сравнении с частотой истинных положительных результатов (чувствительность) на оси ординат. Полученный график называется *кривой ROC* или *графиком ROC* для классификатора.

- Пусть $S(\mathbf{x}_i)$ обозначает действительную оценку положительного результата класса, полученного классификатором M для точки $\mathbf{x}_i$. Пусть максимальный и минимальный пороги оценки, полученные в тестовом наборе данных **D**, будут следующими:

$$\rho^{min} = \min_i \{S(\mathbf{x}_i)\} \qquad \rho^{max} = \max_i \{S(\mathbf{x}_i)\}$$

	True	
Predicted	Pos	Neg
Pos	0	0
Neg	FN	TN

	True	
Predicted	Pos	Neg
Pos	TP	FP
Neg	0	0

	True	
Predicted	Pos	Neg
Pos	TP	0
Neg	0	TN

Площадь под кривой ROC

- Площадь под кривой ROC, сокращенно обозначается AUC (англ. area under ROC curve), может использоваться как мера производительности классификатора. Поскольку общая площадь графика равна 1, AUC лежит в интервале [0,1] - чем выше, тем лучше. Значение AUC - это, по сути, вероятность того, что классификатор оценит случайный положительный тестовый случай выше, чем случайный отрицательный тестовый случай.

ALGORITHM 22.1. ROC Curve and Area under the Curve

ROC-CURVE(D, M):

$n_1 \leftarrow \left|\{\mathbf{x}_i \in \mathbf{D} \,|\, y_i = c_1\}\right|$ // size of positive class

$n_2 \leftarrow \left|\{\mathbf{x}_i \in \mathbf{D} \,|\, y_i = c_2\}\right|$ // size of negative class

// classify, score, and sort all test points

$L \leftarrow$ sort the set $\{(S(\mathbf{x}_i), y_i) : \mathbf{x}_i \in \mathbf{D}\}$ by decreasing scores

$FP \leftarrow TP \leftarrow 0$

$FP_{prev} \leftarrow TP_{prev} \leftarrow 0$

$AUC \leftarrow 0$

$\rho \leftarrow \infty$

foreach $(S(\mathbf{x}_i), y_i) \in L$ **do**

  if $\rho > S(\mathbf{x}_i)$ **then**

    plot point $\left(\frac{FP}{n_2}, \frac{TP}{n_1}\right)$

    $AUC \leftarrow AUC + \text{TRAPEZOID-AREA}\left(\left(\frac{FP_{prev}}{n_2}, \frac{TP_{prev}}{n_1}\right), \left(\frac{FP}{n_2}, \frac{TP}{n_1}\right)\right)$

    $\rho \leftarrow S(\mathbf{x}_i)$

    $FP_{prev} \leftarrow FP$

    $TP_{prev} \leftarrow TP$

  if $y_i = c_1$ **then** $TP \leftarrow TP + 1$

  else $FP \leftarrow FP + 1$

plot point $\left(\frac{FP}{n_2}, \frac{TP}{n_1}\right)$

$AUC \leftarrow AUC + \text{TRAPEZOID-AREA}\left(\left(\frac{FP_{prev}}{n_2}, \frac{TP_{prev}}{n_1}\right), \left(\frac{FP}{n_2}, \frac{TP}{n_1}\right)\right)$

TRAPEZOID-AREA$((x_1, y_1), (x_2, y_2))$:

$b \leftarrow |x_2 - x_1|$ // base of trapezoid

$h \leftarrow \frac{1}{2}(y_2 + y_1)$ // average height of trapezoid

return $(b \cdot h)$

- Значение AUC вычисляется по мере добавления каждой новой точки на график ROC. Алгоритм сохраняет предыдущие значения ложно и истинно положительных результатов, FP_{prev} и TP_{prev}, для предыдущего порога оценки ρ. Учитывая текущие значения FP и TP, мы вычисляем площадь под кривой, определяемой четырьмя точками

$$(x_1, y_1) = \left(\frac{FP_{\text{prev}}}{n_2}, \frac{TP_{\text{prev}}}{n_1}\right) \qquad (x_2, y_2) = \left(\frac{FP}{n_2}, \frac{TP}{n_1}\right)$$

$$(x_1, 0) = \left(\frac{FP_{\text{prev}}}{n_2}, 0\right) (x_2, 0) = \left(\frac{FP}{n_2}, 0\right)$$

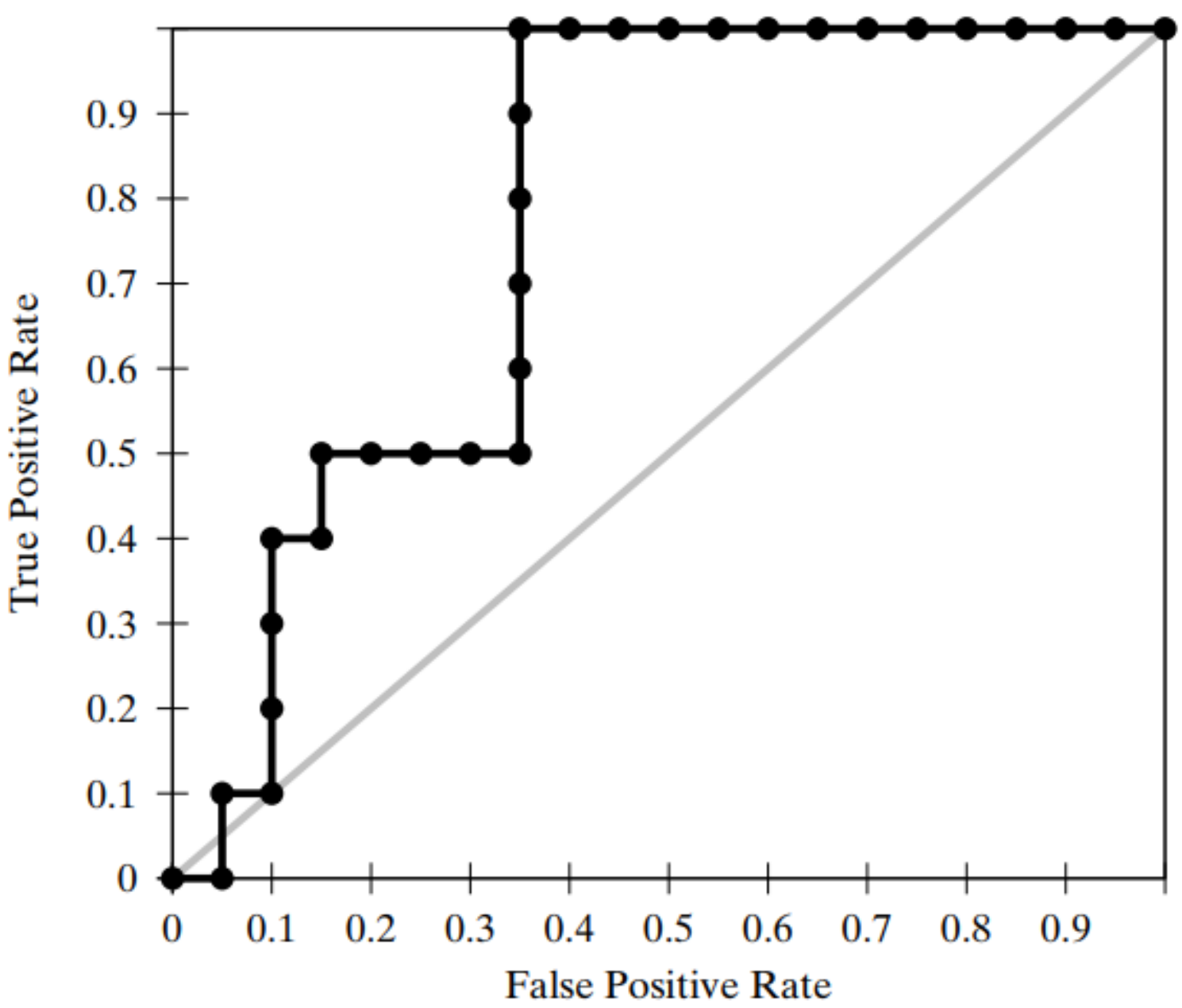


График ROC для главных компонент набора данных Iris. Показаны кривые ROC для наивного байесовского (черный) и случайного (серый) классификаторов

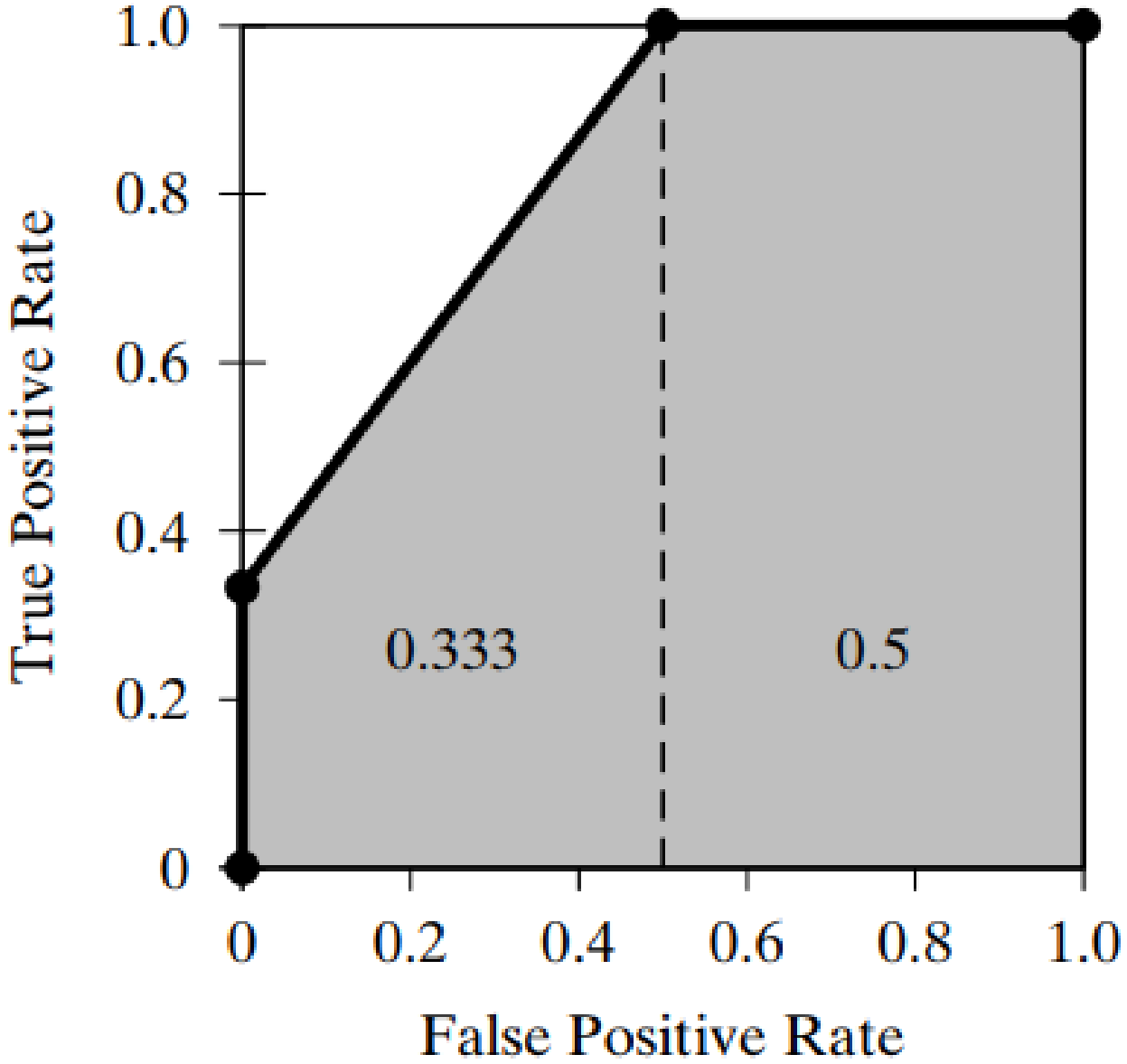


График ROC и AUC: область трапеции

Случайный классификатор

- Любая фиксированная вероятность предсказания, скажем *r*, для положительного класса, дает точку (*r*, *r*) в пространстве ROC. Диагональная линия, таким образом, представляет производительность случайного классификатора по всем возможным порогам прогнозирования положительного класса *r*. Из этого следует, что если кривая ROC для любого классификатора ниже диагонали, это указывает на худшую производительность, чем у случайного предположения. В таких случаях инвертирование присвоения классов дает лучший классификатор. Как следствие диагональной кривой ROC, значение AUC для случайного классификатора составляет 0.5. Таким образом, если какой-либо классификатор имеет значение AUC менее 0.5, это указывает на производительность хуже случайной.

Дисбаланс классов

- Стоит отметить, что кривые ROC нечувствительны к несбалансированным классам. Это связано с тем, что *TPR*, интерпретируемый как вероятность предсказания положительной точки как положительной, и *FPR*, интерпретируемый как вероятность предсказания отрицательной точки как положительной, не зависят от соотношения размеров положительного и отрицательного классов. Это желательное свойство, поскольку кривая ROC по существу останется той же самой, независимо от того, сбалансированы ли классы (имеют относительно одинаковое количество точек) или не сбалансированы (когда один класс имеет намного больше точек, чем другой).

Оценка классификаторов. К-кратная перекрестная проверка

- Перекрестная проверка делит набор данных **D** на *K* частей равного размера, называемых свертками, а именно $\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K$. Каждый сверток $\mathbf{D}_i$, в свою очередь, рассматривается как набор для тестирования, а остальные свертки составляют набор для обучения $\mathbf{D} \setminus \mathbf{D}_i = \bigcup_{j \neq i} \mathbf{D}_j$. После обучения модели M_i на $\mathbf{D} \setminus \mathbf{D}_i$ мы оцениваем ее эффективность на множестве тестирования $\mathbf{D}_i$, чтобы получить *i*-ю оценку θ_i. Ожидаемое значение показателя эффективности затем можно оценить как

$$\hat{\mu}_\theta = E[\theta] = \frac{1}{K} \sum_{i=1}^{K} \theta_i \; (22.3)$$

- и его дисперсия как

$$\hat{\sigma}_\theta^2 = \frac{1}{K} \sum_{i=1}^{K} (\theta_i - \hat{\mu}_\theta)^2, (22.4)$$

ALGORITHM 22.2. *K*-fold Cross-Validation

CROSS-VALIDATION(K, $\mathbf{D}$):

1 $\mathbf{D} \leftarrow$ randomly shuffle $\mathbf{D}$

2 $\{\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K\} \leftarrow$ partition $\mathbf{D}$ in K equal parts

3 **foreach** $i \in [1, K]$ **do**

4 $\quad M_i \leftarrow$ train classifier on $\mathbf{D} \setminus \mathbf{D}_i$

5 $\quad \theta_i \leftarrow$ assess M_i on $\mathbf{D}_i$

6 $\hat{\mu}_\theta = \frac{1}{K} \sum_{i=1}^{K} \theta_i$

7 $\hat{\sigma}_\theta^2 = \frac{1}{K} \sum_{i=1}^{K} (\theta_i - \hat{\mu}_\theta)^2$

8 **return** $\hat{\mu}_\theta, \hat{\sigma}_\theta^2$

Повторная выборка начальной загрузки

ALGORITHM 22.3. Bootstrap Resampling Method

BOOTSTRAP-RESAMPLING(K, D):

1 **for** $i \in [1, K]$ **do**
2 $\quad \mathbf{D}_i \leftarrow$ sample of size n with replacement from **D**
3 $\quad M_i \leftarrow$ train classifier on $\mathbf{D}_i$
4 $\quad \theta_i \leftarrow$ assess M_i on **D**
5 $\hat{\mu}_\theta = \frac{1}{K} \sum_{i=1}^{K} \theta_i$
6 $\hat{\sigma}_\theta^2 = \frac{1}{K} \sum_{i=1}^{K} (\theta_i - \hat{\mu}_\theta)^2$
7 **return** $\hat{\mu}_\theta, \hat{\sigma}_\theta^2$

Доверительные интервалы

- Предположим, что каждое θ_i имеет конечное среднее $E[\theta_i] = \mu$ и конечную дисперсию $var\ (\theta_i) = \sigma^2$.
- Пусть $\hat{\mu}$ обозначает среднее выборки:

$$\hat{\mu} = \frac{1}{K}(\theta_1 + \theta_2 + \cdots + \theta_K)$$

- По линейности ожидания имеем

$$E[\hat{\mu}] = E\left[\frac{1}{K}(\theta_1 + \theta_2 + \cdots + \theta_K)\right] = \frac{1}{K}\sum_{i=1}^{K} E[\theta_i] = \frac{1}{K}(K\mu) = \mu$$

- Используя линейность дисперсии для независимых случайных величин и отмечая, что $\mathrm{var}(aX) = a^2$ для $a \in \mathbb{R}$, дисперсия $\hat{\mu}$ задается как

$$\mathrm{var}(\hat{\mu}) = \mathrm{var}\left(\frac{1}{K}(\theta_1 + \theta_2 + \cdots + \theta_K)\right) = \frac{1}{K^2}\sum_{i=1}^{K} \mathrm{var}(\theta_i) = \frac{1}{K^2}\left(K\sigma^2\right) = \frac{\sigma^2}{K}$$

- Таким образом, стандартное отклонение $\hat{\mu}$ определяется как

$$\mathrm{std}(\hat{\mu}) = \sqrt{\mathrm{var}(\hat{\mu})} = \frac{\sigma}{\sqrt{K}}$$

- Нас интересует распределение *z*-показателя $\hat{\mu}$, которое само по себе является случайной величиной.

$$Z_K = \frac{\hat{\mu} - E[\hat{\mu}]}{\mathrm{std}(\hat{\mu})} = \frac{\hat{\mu} - \mu}{\frac{\sigma}{\sqrt{K}}} = \sqrt{K}\left(\frac{\hat{\mu} - \mu}{\sigma}\right)$$

- Z_K определяет отклонение оценочного среднего от истинного среднего в терминах его стандартного отклонения. Центральная предельная теорема утверждает, что по мере увеличения размера выборки случайная величина Z_K сходится по распределению к стандартному нормальному распределению (которое имеет среднее значение 0 и дисперсию 1). То есть при $K \to \infty$ для любого $x \in \mathbb{R}$ имеем

$$\lim_{K \to \infty} P(Z_K \leq x) = \Phi(x)$$

- где $\Phi(x)$ – кумулятивная функция распределения для стандартной нормальной функции плотности f (x | 0,1).

- Пусть $z_{\alpha/2}$ обозначает значение z-показателя, которое включает $\alpha/2$ вероятностной массы для стандартного нормального распределения, то есть

$$P\left(0 \leq Z_K \leq z_{\alpha/2}\right) = \Phi\left(z_{\alpha/2}\right) - \Phi(0) = \alpha/2$$

- тогда, поскольку нормальное распределение симметрично относительно среднего, мы имеем

$$\lim_{K\to\infty} P\left(-z_{\alpha/2} \leq Z_K \leq z_{\alpha/2}\right) = 2 \cdot P\left(0 \leq Z_K \leq z_{\frac{\alpha}{2}}\right) = \alpha \quad (22.5)$$

- Обратите внимание, что

$$-z_{\frac{\alpha}{2}} \leq Z_K \leq z_{\frac{\alpha}{2}} \Longrightarrow -z_{\frac{\alpha}{2}} \leq \sqrt{K}\left(\frac{\hat{\mu}-\mu}{\sigma}\right) \leq z_{\frac{\alpha}{2}}$$

$$\Longrightarrow -z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{K}} \leq \hat{\mu}-\mu \leq z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{K}}$$

$$\Longrightarrow \left(\hat{\mu}-z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{K}}\right) \leq \mu \leq \left(\hat{\mu}+z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{K}}\right)$$

- Подставляя вышеуказанное в формулу (22.5), получаем оценки истинного среднего μ через оценочное значение $\hat{\mu}$, т. е.

$$\lim_{K\to\infty} P\left(\hat{\mu} - z_{\alpha/2}\frac{\sigma}{\sqrt{K}} \le \mu \le \hat{\mu} + z_{\alpha/2}\frac{\sigma}{\sqrt{K}}\right) = \alpha \ (22.6)$$

- Таким образом, для любого заданного уровня достоверности α мы можем вычислить вероятность того, что истинное среднее значение μ лежит в $\alpha\%$ доверительном интервале $\left(\hat{\mu} - z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{K}}, \hat{\mu} + z_{\alpha/2}\frac{\sigma}{\sqrt{K}}\right)$.

Неизвестная дисперсия

- Приведенный выше анализ предполагает, что нам известна истинная дисперсия σ^2, что, как правило, не так. Однако мы можем заменить σ^2 выборочной дисперсией

$$\hat{\sigma}^2 = \frac{1}{K}\sum_{i=1}^{K} (\theta_i - \hat{\mu})^2 \quad (22.7)$$

- поскольку $\hat{\sigma}^2$ является последовательной оценкой для σ^2, то есть при $K \to \infty$ $\hat{\sigma}^2$ сходится с вероятностью 1, *сходится почти наверняка*, к σ^2. Центральная предельная теорема утверждает, что определенная ниже случайная величина Z_K^* сходится по распределению к стандартному нормальному распределению:

$$Z_K^* = \sqrt{K}\left(\frac{\hat{\mu} - \mu}{\hat{\sigma}}\right) \quad (22.8)$$

- Таким образом,

$$\lim_{K\to\infty} P\left(\hat{\mu} - z_{\alpha/2}\frac{\hat{\sigma}}{\sqrt{K}} \le \mu \le \hat{\mu} + z_{\alpha/2}\frac{\hat{\sigma}}{\sqrt{K}}\right) = \alpha (22.9)$$

- Другими словами,

$$\left(\hat{\mu} - z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{K}}, \hat{\mu} + z_{\alpha/2}\frac{\sigma}{\sqrt{K}}\right)$$

- доверительный интервал α для μ.

Небольшой размер выборки

- Доверительный интервал в формуле (22.9) применяется только при размере выборки K $\rightarrow \infty$. Рассмотрим случайные величины V_i для $i = 1, \ldots, K$, определенные как

$$V_i = \frac{\theta_i - \hat{\mu}}{\sigma}$$

- Далее рассмотрим сумму их квадратов:

$$S = \sum_{i=1}^{K} V_i^2 = \sum_{i=1}^{K} \left(\frac{\theta_i - \hat{\mu}}{\sigma} \right)^2 = \frac{1}{\sigma^2} \sum_{i=1}^{K} (\theta_i - \hat{\mu})^2 = \frac{K\hat{\sigma}}{\sigma^2} \quad (22.10)$$

- Последний шаг следует из определения выборочной дисперсии в формуле (22.7).

- Рассмотрим случайную величину Z_K^* в уравнении (22.8).

$$Z_K^* = \sqrt{K}\left(\frac{\hat{\mu} - \mu}{\hat{\sigma}}\right) = \left(\frac{\hat{\mu} - \mu}{\hat{\sigma}/\sqrt{K}}\right)$$

- Разделив числитель и знаменатель в приведенном выше выражении на $\hat{\sigma}/\sqrt{K}$, получим

$$Z_K^* = \left(\frac{\hat{\mu} - \mu}{\sigma/\sqrt{K}} \Big/ \frac{\hat{\sigma}/\sqrt{K}}{\sigma/\sqrt{K}}\right) = \left(\frac{\frac{\hat{\mu} - \mu}{\sigma/\sqrt{K}}}{\hat{\sigma}/\sigma}\right) = \frac{Z_K}{\sqrt{S/K}} \quad (22.11)$$

- Последний шаг следует из уравнения (22.10), потому что

$$S = \frac{K\hat{\sigma}^2}{\sigma^2} \text{ implies that } \frac{\hat{\sigma}}{\sigma} = \sqrt{S/K}$$

- В частности, мы выбираем значение $t_{\alpha/2,K-1}$ так, чтобы кумулятивная функция распределения t с $K-1$ степенями свободы охватывала $\alpha/2$ вероятностной массы, т. е.

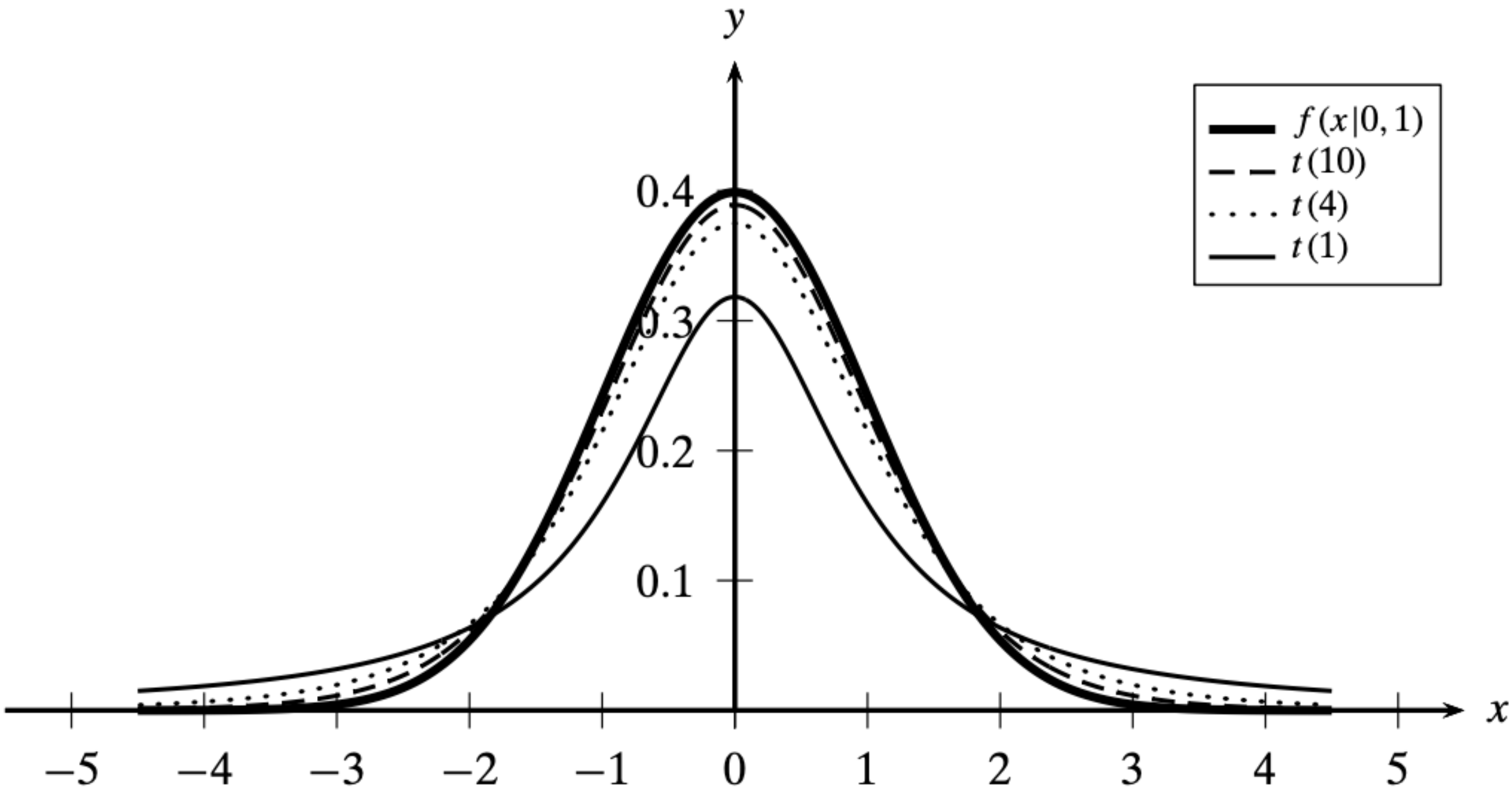
$$P\left(0 \le Z_K^* \le t_{\alpha/2,K-1}\right) = T_{K-1}\left(t_{\alpha/2}\right) - T_{K-1}(0) = \alpha/2$$

- где T_{K-1} – кумулятивная функция распределения для t распределения Стьюдента с $K-1$ степенями свободы. Поскольку t распределение симметрично относительно среднего, мы имеем

$$P\left(\hat{\mu} - t_{\alpha/2,K-1}\frac{\hat{\sigma}}{\sqrt{K}} \le \mu \le \hat{\mu} + t_{\alpha/2,K-1}\frac{\hat{\sigma}}{\sqrt{K}}\right) = \alpha \ (22.12)$$

- Таким образом, доверительный интервал $\alpha\%$ для истинного среднего μ равен

$$\left(\hat{\mu} - t_{\alpha/2,K-1}\frac{\hat{\sigma}}{\sqrt{K}} \le \mu \le \hat{\mu} + t_{\alpha/2,K-1}\frac{\hat{\sigma}}{\sqrt{K}}\right)$$


y
0.4
0.3
0.2
0.1
x
−5
−4
−3
−2
−1
0
1
2
3
4
5
f (x|0, 1)
t (10)
t (4)
t (1)

Сравнение классификаторов: парный t-тест

- Чтобы определить, имеют ли два классификатора разную или одинаковую производительность, определим случайную величину δ_i как разницу в их производительности на *i*-м наборе данных:

$$\delta_i = \theta_i^A - \theta_i^B$$

- Теперь рассмотрим оценки ожидаемой разницы и дисперсии разностей:

$$\hat{\mu}_\delta = \frac{1}{K}\sum_{i=1}^{K} \delta_i \qquad \hat{\sigma}_\delta^2 = \frac{1}{K}\sum_{i=1}^{K} (\delta_i - \hat{\mu}_\delta)$$

- Нулевая гипотеза H_0 состоит в том, что их производительность одинакова, то есть истинная ожидаемая разница равна нулю, тогда как альтернативная гипотеза H_a заключается в том, что они не одинаковы, то есть истинная ожидаемая разница μ_δ не равна нулю:

$$H_0: \quad \mu_\delta = 0 \qquad H_a: \quad \mu_\delta \neq 0$$

- Определим случайную величину z-оценки для ожидаемой разницы как

$$Z_{\delta}^{*} = \sqrt{K}\left(\frac{\hat{\mu}_{\delta} - \mu_{\delta}}{\hat{\sigma}_{\delta}}\right)$$

- Следуя аналогичным аргументам, как в формуле (22.11) Z_{δ}^{*} соответствует t распределению с $K-1$ степенями свободы. Однако при нулевой гипотезе $\mu_{\delta} = 0$, и, следовательно,

$$Z_{\delta}^{*} = \sqrt{K}\left(\frac{\hat{\mu}_{\delta}}{\hat{\sigma}_{\delta}}\right) \sim t_{K-1}$$

- где обозначение $Z_{\delta}^{*} \sim t_{K-1}$ означает, что Z_{δ}^{*} следует распределению t с $K-1$ степенями свободы. Учитывая желаемый уровень достоверности α, мы заключаем, что

$$P\left(-t_{\frac{\alpha}{2},K-1} \leq Z_{\delta}^{*} \leq t_{\frac{\alpha}{2},K-1}\right) = \alpha$$

ALGORITHM 22.4. Paired t-Test via Cross-Validation

PAIRED t-TEST(α, K, $\mathbf{D}$):

1 $\mathbf{D} \leftarrow$ randomly shuffle $\mathbf{D}$
2 $\{\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K\} \leftarrow$ partition $\mathbf{D}$ in K equal parts
3 **foreach** $i \in [1, K]$ **do**
4 $\quad M_i^A, M_i^B \leftarrow$ train the two different classifiers on $\mathbf{D} \setminus \mathbf{D}_i$
5 $\quad \theta_i^A, \theta_i^B \leftarrow$ assess M_i^A and M_i^B on $\mathbf{D}_i$
6 $\quad \delta_i = \theta_i^A - \theta_i^B$
7 $\hat{\mu}_\delta = \frac{1}{K} \sum_{i=1}^{K} \delta_i$
8 $\hat{\sigma}_\delta^2 = \frac{1}{K} \sum_{i=1}^{K} (\delta_i - \hat{\mu}_\delta)^2$
9 $Z_\delta^* = \frac{\sqrt{K}\hat{\mu}_\delta}{\hat{\sigma}_\delta}$
10 **if** $Z_\delta^* \in \left(-t_{\alpha/2,K-1}, t_{\alpha/2,K-1}\right)$ **then**
11 $\quad$ Accept H_0; both classifiers have similar performance
12 **else**
13 $\quad$ Reject H_0; classifiers have significantly different performance

Разложение на смещение и разброс (Bias-variance decomposition)

- Для обучающего набора $\mathbf{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, содержащего *n* точек $\mathbf{x}_i \in \mathbb{R}^d$, с соответствующими им классами y_i, изученная модель классификации *M* предсказывает класс для данной контрольной точки **x**.
- *Функция потерь* определяет стоимость или штраф за предсказание того, что класс будет $\hat{y} = M(\boldsymbol{x})$, когда истинным классом является *y*. Обычно используемой функцией потерь для классификации является коэффициент неправильной классификации (zero-one loss), определяемый как

$$L(y, M(\mathbf{x})) = I(M(\mathbf{x}) \neq y) = \begin{cases} 0 & \text{if } M(\mathbf{x}) = y \\ 1 & \text{if } M(\mathbf{x}) \neq y \end{cases}$$

- Другим часто используемой функцией потерь является *квадрат потерь*, определяемый как

$$L(y, M(\mathbf{x})) = (y - M(\mathbf{x}))^2$$

Ожидаемые потери

- Идеальный или оптимальный классификатор - это тот, который минимизирует функцию потерь. Минимизация ожидаемых потерь:

$$E_y[L(y, M(\mathbf{x})) \mid \mathbf{x}] = \sum_y L(y, M(\mathbf{x})) \cdot P(y \mid \mathbf{x}) \quad (22.13)$$

- где $P(y \mid \mathbf{x})$ - это условная вероятность класса y для данной контрольной точки $\mathbf{x}$, а E_y означает, что математическое ожидание берется для различных значений класса y.

- Минимизация ожидаемого коэффициента неправильной классификации соответствует минимизации коэффициента ошибок. Пусть $M(\mathbf{x}) = c_i$, тогда имеем

$$E_y[L(y, M(\mathbf{x})) \mid \mathbf{x}] = E_y[I(y \neq M(\mathbf{x})) \mid \mathbf{x}]$$

$$= \sum_y I(y \neq c_i) \cdot P(y \mid \mathbf{x})$$

$$= \sum_{y \neq c_i} P(y \mid \mathbf{x})$$

$$= 1 - P(c_i \mid \mathbf{x})$$

- Таким образом, чтобы минимизировать ожидаемые потери, мы должны выбрать c_i как класс, который максимизирует апостериорную вероятность, то есть $c_i = \text{argmax}_y P(y|\mathbf{x})$.

Смещение и дисперсия. Ожидаемый квадрат потерь (22.14)

$$E_y[L(y, M(\mathbf{x}, \mathbf{D})) \mid \mathbf{x}, \mathbf{D}]$$

$$= E_y\left[(y - M(\mathbf{x}, \mathbf{D}))^2 \mid \mathbf{x}, \mathbf{D}\right]$$

$$= E_y\left[(y \underbrace{-E_y[y \mid \mathbf{x}] + E_y[y \mid \mathbf{x}]}_{\text{add and subtract same term}} - M(\mathbf{x}, \mathbf{D}))^2 \mid \mathbf{x}, \mathbf{D}\right]$$

$$= E_y\left[\left(y - E_y[y \mid \mathbf{x}]\right)^2 \mid \mathbf{x}, \mathbf{D}\right] + E_y\left[\left(M(\mathbf{x}, \mathbf{D}) - E_y[y \mid \mathbf{x}]\right)^2 \mid \mathbf{x}, \mathbf{D}\right]$$

$$+E_y\left[2\left(y - E_y[y \mid \mathbf{x}]\right) \cdot \left(E_y[y \mid \mathbf{x}] - M(\mathbf{x}, \mathbf{D})\right) \mid \mathbf{x}, \mathbf{D}\right]$$

$$= E_y\left[\left(y - E_y[y \mid \mathbf{x}]\right)^2 \mid \mathbf{x}, \mathbf{D}\right] + \left(M(\mathbf{x}, \mathbf{D}) - E_y[y \mid \mathbf{x}]\right)^2$$

$$+2\left(E_y[y \mid \mathbf{x}] - M(\mathbf{x}, \mathbf{D})\right) \cdot \underbrace{\left(E_y[y \mid \mathbf{x}] - E_y[y \mid \mathbf{x}]\right)}_{0}$$

$$= \underbrace{E_y\left[\left(y - E_y[y \mid \mathbf{x}]\right)^2 \mid \mathbf{x}, \mathbf{D}\right]}_{\text{var}\,(y|\mathbf{x})} + \underbrace{\left(M(\mathbf{x}, \mathbf{D}) - E_y[y \mid \mathbf{x}]\right)^2}_{\text{squared-error}}$$

- Средняя или ожидаемая квадратичная ошибка для данной контрольной точки **x** по всем обучающим выборкам (22.15) тогда задается как

$$E_{\mathbf{D}}\left[\left(M(\mathbf{x},\mathbf{D}) - E_y[y \mid \mathbf{x}]\right)^2\right]$$

$$= E_{\mathbf{D}}\left[(M(\mathbf{x},\mathbf{D}) \underbrace{-E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})] + E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})]}_{\text{add and subtract same term}} - E_y[y \mid \mathbf{x}])^2\right]$$

$$= E_{\mathbf{D}}\left[(M(\mathbf{x},\mathbf{D}) - E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})])^2\right] + E_{\mathbf{D}}\left[\left(E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})] - E_y[y \mid \mathbf{x}]\right)^2\right]$$

$$+2\left(E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})] - E_y[y \mid \mathbf{x}]\right) \cdot \underbrace{\left(E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})] - E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})]\right)}_{0}$$

$$= \underbrace{E_{\mathbf{D}}\left[(M(\mathbf{x},\mathbf{D}) - E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})])^2\right]}_{\text{variance}} + \underbrace{\left(E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})] - E_y[y \mid \mathbf{x}]\right)^2}_{\text{bias}}$$

- Комбинируя уравнения (22.14) и (22.15) ожидаемый квадрат потерь по всем тестовым точкам **x** и по всем обучающим наборам **D** размера **n** дает следующее разложение на шум, дисперсию и смещение (22.16):

$$E_{\mathbf{x},\mathbf{D},y}\left[(y - M(\mathbf{x},\mathbf{D}))^2\right]$$

$$= E_{\mathbf{x},\mathbf{D},y}\left[\left(y - E_y[y \mid \mathbf{x}]\right)^2 \mid \mathbf{x},\mathbf{D}\right] + E_{\mathbf{x},\mathbf{D}}\left[\left(M(\mathbf{x},\mathbf{D}) - E_y[y \mid \mathbf{x}]\right)^2\right]$$

$$= \underbrace{E_{\mathbf{x},y}\left[\left(y - E_y[y \mid \mathbf{x}]\right)^2\right]}_{\text{noise}} + \underbrace{E_{\mathbf{x},\mathbf{D}}\left[(M(\mathbf{x},\mathbf{D}) - E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})])^2\right]}_{\text{average variance}}$$

$$+ \underbrace{E_{\mathbf{x}}\left[\left(E_{\mathbf{D}}[M(\mathbf{x},\mathbf{D})] - E_y[y \mid \mathbf{x}]\right)^2\right]}_{\text{average bias}}$$

Классификаторы ансамбля

- Классификатор называется *неустойчивым*, если небольшие возмущения в обучающей выборке приводят к большим изменениям границы прогноза или решения. Цель обучения - уменьшить ошибку классификации за счет уменьшения дисперсии или смещения, в идеале и того, и другого. Методы ансамбля создают *комбинированный классификатор*, используя выходные данные нескольких *базовых классификаторов*, которые обучаются на разных подмножествах данных.

Бэггинг

- Бэггинг, сокращенно от бутстреп-агрегирования, представляет собой метод классификации ансамбля, который использует несколько бутстреп выборок (с заменой) из входных обучающих данных **D** для создания немного разных обучающих наборов $\mathbf{D}_i$, $i = 1,2, \ldots, K$.
- Пусть количество классификаторов, которые предсказывают класс **x** как c_j, задано как

$$v_j(\mathbf{x}) = \left|\left\{M_i(\mathbf{x}) = c_j \mid i = 1, \ldots, K\right\}\right|$$

- Комбинированный классификатор, обозначенный $\boldsymbol{M}^K$, предсказывает класс контрольной точки **x** *большинством голосов* среди *k* классов:

$$\mathbf{M}^K(\mathbf{x}) = \arg\max_{c_j} \left\{v_j(\mathbf{x}) \mid j = 1, \ldots, k\right\}$$

- Для двоичной классификации, предполагая, что классы заданы как {+1, −1}, объединенный классификатор $\mathbf{M}^K$ может быть выражен проще

$$\mathbf{M}^K(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{K} M_i(\mathbf{x})\right)$$

- Бэггинг может помочь уменьшить дисперсию, особенно если базовые классификаторы неустойчивы, из-за эффекта усреднения при голосовании большинством. В целом это не оказывает большого влияния на смещение.

Бустинг

- *Бустинг* - это еще один метод ансамбля, который тренирует базовые классификаторы на разных выборках. Однако основная идея состоит в том, чтобы тщательно отбирать выборки, чтобы *повысить* производительность трудно классифицируемых экземпляров.
- Бустинг наиболее выгоден, когда базовые классификаторы *слабые*, то есть имеют коэффициент ошибок, который немного меньше, чем у случайного классификатора.

ALGORITHM 22.5. Adaptive Boosting Algorithm: AdaBoost

ADABOOST(K, D):

1 $\mathbf{w}^0 \leftarrow \left(\frac{1}{n}\right) \cdot \mathbf{1} \in \mathbb{R}^n$

2 $t \leftarrow 1$

3 **while** $t \leq K$ **do**

5 $\quad \mathbf{D}_t \leftarrow$ weighted resampling with replacement from $\mathbf{D}$ using $\mathbf{w}^{t-1}$

6 $\quad M_t \leftarrow$ train classifier on $\mathbf{D}_t$

7 $\quad \epsilon_t \leftarrow \sum_{i=1}^{n} w_i^{t-1} \cdot I\big(M_t(\mathbf{x}_i) \neq y_i\big)$ `// weighted error rate on` $\mathbf{D}$

8 $\quad$ **if** $\epsilon_t = 0$ **then break**

9 $\quad$ **else if** $\epsilon_t < 0.5$ **then**

10 $\quad\quad \alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ `// classifier weight`

11 $\quad\quad$ **foreach** $i \in [1, n]$ **do**

$\quad\quad\quad$ `// update point weights`

12 $\quad\quad\quad w_i^t = \begin{cases} w_i^{t-1} & \text{if } M_t(\mathbf{x}_i) = y_i \\ w_i^{t-1}\left(\frac{1-\epsilon_t}{\epsilon_t}\right) & \text{if } M_t(\mathbf{x}_i) \neq y_i \end{cases}$

14 $\quad\quad \mathbf{w}^t = \frac{\mathbf{w}^t}{\mathbf{1}^T\mathbf{w}^t}$ `// normalize weights`

15 $\quad\quad t \leftarrow t + 1$

16 **return** $\{M_1, M_2, \ldots, M_K\}$

- **Adaptive Boosting: AdaBoost** Пусть **D** - входной обучающий набор, состоящий из *n* точек $\mathbf{x}_i \in \mathbb{R}^d$.
- Изначально все точки имеют одинаковый вес, то есть

$$\mathbf{w}^0 = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)^T = \frac{1}{n}\mathbf{1}$$

- где $1 \in \mathbb{R}^n$ n-мерный вектор всех единиц. Во время итерации *t* обучающая выборка $\mathbf{D}_t$ получается посредством взвешенной передискретизации с использованием распределения $\mathbf{w}^{t-1}$, то есть мы рисуем выборку размера *n* с заменой, так что *i*-я точка выбирается в соответствии с ее вероятностью ω_i^{t-1}. Затем мы обучаем классификатор M_t, используя $\mathbf{D}_t$, и вычисляем его взвешенную частоту ошибок ϵ_t для всего входного набора данных **D**:

$$\epsilon_t = \sum_{i=1}^{n} w_i^{t-1} \cdot I(M_t(\mathbf{x}_i) \neq y_i)$$

- где *I* - индикаторная функция, которая равна 1, когда ее аргумент истинен, то есть когда M_t неверно классифицирует $\mathbf{x}_i$, и равен 0 в противном случае.

- Затем вес для t-го классификатора устанавливается как

$$\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

- и вес для каждой точки $\mathbf{x}_i \in \mathbf{D}$ обновляется в зависимости от того, была ли точка классифицирована неправильно или нет

$$w_i^t = w_i^{t-1} \cdot \exp\{\alpha_t \cdot I(M_t(\mathbf{x}_i) \neq y_i)\}$$

- Таким образом, если предсказанный класс соответствует истинному классу, то есть, если $M_t(\mathbf{x}_i) = y_i$, то $I(M_t(\mathbf{x}_i) \neq y_i) = 0$, и вес для точки $\mathbf{x}_i$ остается неизменным. С другой стороны, если точка классифицирована неправильно, то есть $M_t(\mathbf{x}_i) \neq y_i$, то мы имеем $I(M_t(\mathbf{x}_i) \neq y_i) = 1$, и

$$w_i^t = w_i^{t-1} \cdot \exp\{\alpha_t\} = w_i^{t-1} \exp\left\{\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\right\} = w_i^{t-1}\left(\frac{1}{\epsilon_t} - 1\right)$$

- После обновления весов точек мы повторно нормализуем веса так, чтобы $\mathbf{w}_t$ стало вектором вероятности (строка 14):

$$\mathbf{w}^t = \frac{\mathbf{w}^t}{\mathbf{1}^T\mathbf{w}^t} = \frac{1}{\sum_{j=1}^n w_j^t}\left(w_1^t, w_2^t, \ldots, w_n^t\right)^T$$

Комбинированный классификатор

- Учитывая набор классификаторов после бустинга M_1, M_2, ..., M_K, вместе с их весами α_1, α_2, ..., α_K, класс для контрольного примера **x** получается посредством голосования взвешенного большинства. Пусть v_j (**x**) обозначает взвешенный голос за класс c_j над K классификаторами, заданный как

$$v_j(\mathbf{x}) = \sum_{t=1}^{K} \alpha_t \cdot I\left(M_t(\mathbf{x}) = c_j\right)$$

- Поскольку $I(M_t(\mathbf{x}) = c_j)$ равно 1, только когда M_t (**x**) = c_j, переменная v_j (**x**) просто получает результат для класса c_j среди K базовых классификаторов с учетом весов классификатора. Комбинированный классификатор, обозначенный $\mathbf{M}^K$, затем прогнозирует класс для **x** следующим образом:

$$\mathbf{M}^K(\mathbf{x}) = \arg\max_{c_j} \left\{ v_j(\mathbf{x}) \mid j = 1, \ldots, k \right\}$$

- В случае двоичной классификации с классами {+1, −1} комбинированный классификатор $\mathbf{M}^K$ можно выразить проще как

$$\mathbf{M}^K(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{K} \alpha_t M_t(\mathbf{x})\right)$$

- **Бэггинг как частный случай AdaBoost:** Бэггинг можно рассматривать как частный случай AdaBoost, где $\omega^t = \frac{1}{n}\mathbf{1}$, а α_t = 1 для всех *K* итераций. В этом случае для взвешенной повторной выборки по умолчанию используется обычная повторная выборка с заменой, а для прогнозируемого класса для тестового примера также по умолчанию используется простое правило большинства.