

Metody Sztucznej Inteligencji

Wykład 4: **Projektowanie eksperymentów oceniających jakość klasyfikacji**

dr inż. Paweł Ksieniewicz

Katedra Systemów i Sieci Komputerowych

10 kwietnia 2019

If you torture the data long enough,
it will confess.

Ronald H. Coase

~~~~~  
Dlaczego?  
~~~~~

Motywacje

Projektując klasyfikator staramy się odpowiedzieć na dwa kluczowe pytania:

Motywacje

Projektując klasyfikator staramy się odpowiedzieć na dwa kluczowe pytania:

- 1 Jak możemy oszacować oczekiwany błąd klasyfikatora dla zadanego problemu?

Motywacje

Projektując klasyfikator staramy się odpowiedzieć na dwa kluczowe pytania:

- 1 Jak możemy oszacować oczekiwany błąd klasyfikatora dla zadanego problemu?
- 2 Jak możemy stwierdzić, że jeden klasyfikator generuje dla zadanego problemu mniejszy oczekiwany błąd niż inny?

Uwagi początkowe

\mathcal{DS} , \mathcal{TS} i \mathcal{VS}

Uwagi początkowe

$$\mathcal{DS}, \mathcal{TS} \text{ i } \mathcal{VS}$$

↪ Przystępując do badań, zadany problem jest klasyfikacja dostępnego nam **zbioru danych** \mathcal{DS} .

Uwagi początkowe

\mathcal{DS} , \mathcal{TS} i \mathcal{VS}

- ↪ Przystępując do badań, zadany problem jest klasyfikacja dostępnego nam **zbioru danych** \mathcal{DS} .
- ↪ Zbiór danych wykorzystany do uczenia klasyfikatora nazywamy **zbiorem uczącym** \mathcal{TS} .

Uwagi początkowe

\mathcal{DS} , \mathcal{TS} i \mathcal{VS}

- ↪ Przystępując do badań, zadaniem problemem jest klasyfikacja dostępnego nam **zbioru danych** \mathcal{DS} .
- ↪ Zbiór danych wykorzystany do uczenia klasyfikatora nazywamy **zbiorem uczącym** \mathcal{TS} .
- ↪ Teoretycznie moglibyśmy użyć go w pełni jako zbioru uczącego, jednak **błąd** uzyskany na zbiorze uczącym **nie może** zostać użyty do porównania dwóch algorytmów.

Uwagi początkowe

\mathcal{DS} , \mathcal{TS} i \mathcal{VS}

- ↪ Przystępując do badań, zadany problem jest klasyfikacja dostępnego nam **zbioru danych** \mathcal{DS} .
- ↪ Zbiór danych wykorzystany do uczenia klasyfikatora nazywamy **zbiorem uczącym** \mathcal{TS} .
- ↪ Teoretycznie moglibyśmy użyć go w pełni jako zbioru uczącego, jednak **błąd** uzyskany na zbiorze uczącym **nie może** zostać użyty do porównania dwóch algorytmów.
- ↪ Potrzebujemy **zbioru walidacyjnego** \mathcal{VS} , różniącego się od **zbioru uczącego**.



Uwagi początkowe

Podział zbioru danych pół na pół.

Uwagi początkowe

Podział zbioru danych pól na pól.

- ↪ Przy podziale zbioru danych \mathcal{DS} na \mathcal{TS} i \mathcal{VS} , pojedyncza runda walidacji nie będzie wystarczająca, ponieważ:

Uwagi początkowe

Podział zbioru danych pół na pół.

- ↗ Przy podziale zbioru danych \mathcal{DS} na \mathcal{TS} i \mathcal{VS} , pojedyncza runda walidacji nie będzie wystarczająca, ponieważ:
 - × \mathcal{TS} i \mathcal{VS} mogą okazać się niewielkie i zawierać szum oraz wyjątki w postaci obserwacji odstających (*outliers*).

Uwagi początkowe

Podział zbioru danych pół na pół.

- ↗ Przy podziale zbioru danych \mathcal{DS} na \mathcal{TS} i \mathcal{VS} , pojedyncza runda walidacji nie będzie wystarczająca, ponieważ:
- × \mathcal{TS} i \mathcal{VS} mogą okazać się niewielkie i zawierać szum oraz wyjątki w postaci obserwacji odstających (*outliers*).
 - × Metody uczące mogą być zależne od innych losowych czynników mających wpływ na jakość generalizacji (jak choćby punkt startowy dla *metody gradientu prostego*).

Uwagi początkowe

Podział zbioru danych pół na pół.

- ↗ Przy podziale zbioru danych \mathcal{DS} na \mathcal{TS} i \mathcal{VS} , pojedyncza runda walidacji nie będzie wystarczająca, ponieważ:
 - × \mathcal{TS} i \mathcal{VS} mogą okazać się niewielkie i zawierać szum oraz wyjątki w postaci obserwacji odstających (*outliers*).
 - × Metody uczące mogą być zależne od innych losowych czynników mających wpływ na jakość generalizacji (jak choćby punkt startowy dla *metody gradientu prostego*).
 - * *metody niedeterministyczne są generalnie przygnębiające i pokazują światu niemoc, więc jeśli nie chcemy odczuwać depresji podczas badań, starajmy się ich unikać*

Uwagi początkowe

Klasyfikatorem nazywamy tak *algorytm* jak zbudowany przez niego *model*.

Uwagi początkowe

Klasyfikatorem nazywamy tak *algorytm* jak zbudowany przez niego *model*.

- ↪ Jeśli wytrenujemy klasyfikator jednokrotnie, uzyskamy tylko jeden klasyfikator i jeden błąd walidacji.

Klasyfikatorem nazywamy tak *algorytm* jak zbudowany przez niego *model*.

- ↪ Jeśli wytrenujemy klasyfikator jednokrotnie, uzyskamy tylko jeden klasyfikator i jeden błąd walidacji.
- ↪ Aby uniezależnić się od losowych czynników mających wpływ na jakość generalizacji, na podstawie jednego algorytmu trenujemy kilka klasyfikatorów i testujemy je na wielu zbiorach walidacyjnych. Otrzymamy tak wektor błędów.

Klasyfikatorem nazywamy tak *algorytm* jak zbudowany przez niego *model*.

- ↪ Jeśli wytrenujemy klasyfikator jednokrotnie, uzyskamy tylko jeden klasyfikator i jeden błąd walidacji.
- ↪ Aby uniezależnić się od losowych czynników mających wpływ na jakość generalizacji, na podstawie jednego algorytmu trenujemy kilka klasyfikatorów i testujemy je na wielu zbiorach walidacyjnych. Otrzymamy tak wektor błędów.
- ↪ Możemy dzięki temu uzyskać oczekiwany błąd algorytmu klasyfikacji lub porównać go z innymi algorytmami na podstawie dystrybucji zawartych w nim błędów.

Uwagi początkowe

Rzeczywistość.

Uwagi początkowe

Rzeczywistość.

- ↗ Wszystkie wnioski wyciągnięte z eksperymentów są zależne od danego zbioru danych.

Uwagi początkowe

Rzeczywistość.

- ↪ Wszystkie wnioski wyciągnięte z eksperymentów są zależne od danego zbioru danych.
- ↪ W zgodzie z teorią o braku darmowych obiadów Wolperta, **najlepszy algorytm klasyfikacji nie istnieje.**

Uwagi początkowe

Rzeczywistość.

- ↪ Wszystkie wnioski wyciągnięte z eksperymentów są zależne od danego zbioru danych.
- ↪ W zgodzie z teorią o braku darmowych obiadów Wolperta, **najlepszy algorytm klasyfikacji nie istnieje.**
- ↪ Podział zbioru danych na \mathcal{TS} i \mathcal{VS} dokonywany jest tylko na potrzeby testowania, a finalny klasyfikator uczy się już na całym zbiorze \mathcal{DS} .

Uwagi początkowe

Tak naprawdę to

Uwagi początkowe

Tak naprawdę to jest jeszcze czwarty zbiór.

Uwagi początkowe

Tak naprawdę to jest jeszcze czwarty zbiór.

↪ \mathcal{TS} wykorzystujemy do optymalizacji parametrów.

Uwagi początkowe

Tak naprawdę to jest jeszcze czwarty zbiór.

- ↪ \mathcal{TS} wykorzystujemy do optymalizacji parametrów.
- ↪ \mathcal{VS} wykorzystujemy do optymalizacji *hiperparametrów* algorytmu.

Uwagi początkowe

Tak naprawdę to jest jeszcze czwarty zbiór.

- ↪ \mathcal{TS} wykorzystujemy do optymalizacji parametrów.
- ↪ \mathcal{VS} wykorzystujemy do optymalizacji *hiperparametrów* algorytmu.
- ↪ Zbiór testowy wykorzystujemy na końcu.

Kryteria oceny

Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:

Kryteria oceny

Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:

↪ Oczekiwany błąd lub funkcja straty.

Kryteria oceny

Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:

- ↪ Oczekiwany błąd lub funkcja straty.
- ↪ Czas i złożoność obliczeniowa* uczenia.

Kryteria oceny

Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:

- ↪ Oczekiwany błąd lub funkcja straty.
- ↪ Czas i złożoność obliczeniowa* uczenia.
- ↪ Czas i złożoność obliczeniowa* testowania.

Kryteria oceny

Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:

- ↪ Oczekiwany błąd lub funkcja straty.
- ↪ Czas i złożoność obliczeniowa* uczenia.
- ↪ Czas i złożoność obliczeniowa* testowania.
- ↪ Zapotrzebowanie pamięciowe.

Kryteria oceny

Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:

- ↪ Oczekiwany błąd lub funkcja straty.
- ↪ Czas i złożoność obliczeniowa* uczenia.
- ↪ Czas i złożoność obliczeniowa* testowania.
- ↪ Zapotrzebowanie pamięciowe.
- ↪ Interpretowalność – jeśli metoda zezwala na ekstrakcję wiedzy.

Kryteria oceny

Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:

- ↪ Oczekiwany błąd lub funkcja straty.
- ↪ Czas i złożoność obliczeniowa* uczenia.
- ↪ Czas i złożoność obliczeniowa* testowania.
- ↪ Zapotrzebowanie pamięciowe.
- ↪ Interpretowalność – jeśli metoda zezwala na ekstrakcję wiedzy.
- ↪ Łatwość implementacji.

Kryteria oceny

Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:


- ↪ Oczekiwany błąd lub funkcja straty.
- ↪ Czas i złożoność obliczeniowa* uczenia.
- ↪ Czas i złożoność obliczeniowa* testowania.
- ↪ Zapotrzebowanie pamięciowe.
- ↪ Interpretowalność – jeśli metoda zezwala na ekstrakcję wiedzy.
- ↪ Łatwość implementacji.
- ↪ *Cost-sensitive learning* (koszt podjęcia decyzji, lub rzadziej, koszt uczenia).

Kryteria oceny


Możemy porównywać ze sobą algorytmy, wykorzystując poniższe kryteria:

- ↪ Oczekiwany błąd lub funkcja straty.
- ↪ Czas i złożoność obliczeniowa* uczenia.
- ↪ Czas i złożoność obliczeniowa* testowania.
- ↪ Zapotrzebowanie pamięciowe.
- ↪ Interpretowalność – jeśli metoda zezwala na ekstrakcję wiedzy.
- ↪ Łatwość implementacji.
- ↪ *Cost-sensitive learning* (koszt podjęcia decyzji, lub rzadziej, koszt uczenia).

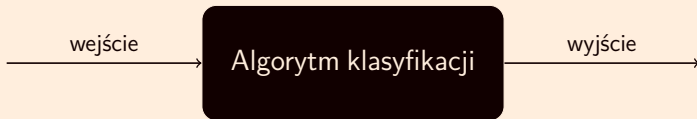
**To zdecydowanie powinno pojawić się gdzieś wcześniej na studiach. Jeśli budujemy jakiegokolwiek modele lub cokolwiek optymalizujemy, powinniśmy znać podstawy struktur danych i złożoności obliczeniowej.*



Wytyczne projektowania eksperymentów uczenia maszyn



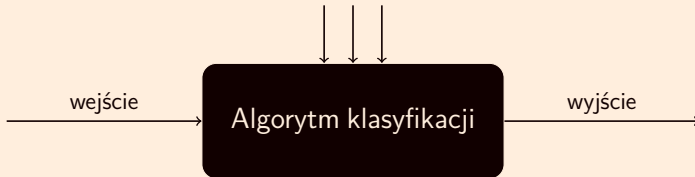
Projektowanie eksperymentu



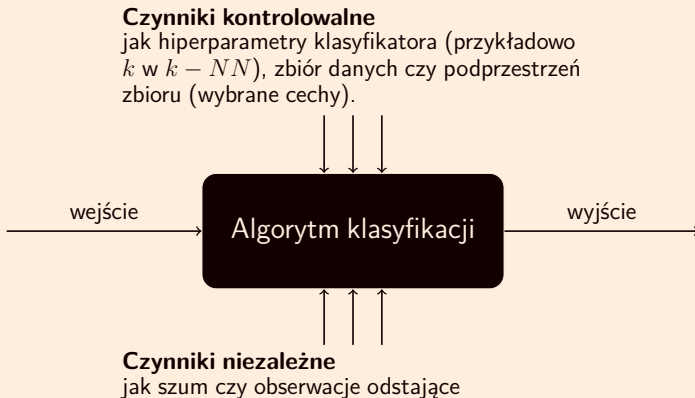
Projektowanie eksperymentu

Czynniki kontrolowalne

jak hiperparametry klasyfikatora (przykładowo k w $k - NN$), zbiór danych czy podprzestrzeń zbioru (wybrane cechy).



Projektowanie eksperymentu



Projektowanie eksperymentu

- ↪ Staramy się sprawdzić w jaki sposób **kontrolowalne czynniki** procesu uczenia wpływają na **wybrane kryterium** oceny.

Projektowanie eksperymentu

- ↪ Staramy się sprawdzić w jaki sposób **kontrolowalne czynniki** procesu uczenia wpływają na **wybrane kryterium** oceny.
- ↪ Możemy zastosować jedno z trzech podejść.

1. Ślepa kura

↪ Zaczynamy ze standardowymi parametrami i losowo modyfikujemy ich wartości, licząc na szczęście.

1. Ślepa kura

- ↪ Zaczynamy ze standardowymi parametrami i losowo modyfikujemy ich wartości, licząc na szczęście.
- ↪ Nie jest to przesadnie systematyczne podejście, nie gwarantuje odnalezienia najlepszego zestawu parametrów.

1. Ślepa kura

- ↪ Zaczynamy ze standardowymi parametrami i losowo modyfikujemy ich wartości, licząc na szczęście.
- ↪ Nie jest to przesadnie systematyczne podejście, nie gwarantuje odnalezienia najlepszego zestawu parametrów.
- ↪ Jeśli badacz dysponuje dobrą intuicją (lub dostateczną wiedzą o problemie) może się sprawdzić.

2. Zmieniaj jeden parametr na raz

- ↪ Wybieramy zakresy, w którym będziemy analizować wpływ poszczególnych parametrów.

2. Zmieniaj jeden parametr na raz

- ↪ Wybieramy zakresy, w którym będziemy analizować wpływ poszczególnych parametrów.
- ↪ Główną wadą takiego podejścia jest założenie, że parametry wzajemnie na siebie nie wpływają.

3. Podejście wieloczynnikowe

- ↪ Parametry są kombinowane wspólnie w założonej przestrzeni kombinacji (najpopularniejszy przykład to *grid search*).

3. Podejście wieloczynnikowe

- ↪ Parametry są kombinowane wspólnie w założonej przestrzeni kombinacji (najpopularniejszy przykład to *grid search*).
- ↪ Głównym problemem jest to, że działa ono głównie w przestrzeniach dyskretnych.

3. Podejście wieloczynnikowe

- ↪ Parametry są kombinowane wspólnie w założonej przestrzeni kombinacji (najpopularniejszy przykład to *grid search*).
- ↪ Głównym problemem jest to, że działa ono głównie w przestrzeniach dyskretnych.
- ↪ Pojawia się kwestia dyskretyzacji ciągłych wartości parametrów.



Główne zasady

↪ **Losowość** — kolejność przebiegów powinna być losowa, aby zapewnić **niezależność wyników eksperymentów**.

Główne zasady

- ↪ **Losowość** — kolejność przebiegów powinna być losowa, aby zapewnić **niezależność wyników eksperymentów**.
- ↪ **Replikacja** — dla każdej kombinacji parametrów, eksperyment powinien być powtórzony wielokrotnie, aby **uśrednić wpływ czynników niekontrolowalnych**.

Główne zasady

- ↗ **Losowość** — kolejność przebiegów powinna być losowa, aby zapewnić **niezależność wyników eksperymentów**.
- ↗ **Replikacja** — dla każdej kombinacji parametrów, eksperyment powinien być powtórzony wielokrotnie, aby **uśrednić wpływ czynników niekontrolowalnych**.
- ↗ **Blokowanie** — jeśli porównujemy algorytmy uczenia, powinny zostać one przebadane **na tych samych podzbiorach** danych, ponieważ w innym wypadku błąd nie będzie zależeć tylko od różnych algorytmów, ale także od różnych podzbiorów.

Eksperymenty SI — wytyczne

1. Jasno zdefiniuj cele.

Eksperymenty SI — wytyczne

1. Jasno zdefiniuj cele.
2. Wybierz adekwatną miarę jakości.

Eksperymenty SI — wytyczne

1. Jasno zdefiniuj cele.
2. Wybierz adekwatną miarę jakości.
3. Wybierz należyte czynniki (szukasz najlepszej wartości któregoś z hiperparametrów, najlepszego algorytmu z wybranej puli czy może z jakiejś dziwnej przyczyny najlepszego zbioru danych?).

Eksperymenty SI — wytyczne

4. Zaprojektuj eksperyment:

Eksperymenty SI — wytyczne

4. Zaprojektuj eksperyment:

- × Wybierz liczbę replikacji (najczęściej zależy od wielkości zbioru).

Eksperymenty SI — wytyczne

4. Zaprojektuj eksperyment:

- × Wybierz liczbę replikacji (najczęściej zależy od wielkości zbioru).
- × Pamiętaj o tym, że używanie niewielkich zbiorów prowadzi do uzyskania niestabilnych wyników, a więc nie będą one znaczące statystycznie lub nie doprowadzą do żadnych wniosków.

4. Zaprojektuj eksperyment:

- × Wybierz liczbę replikacji (najczęściej zależy od wielkości zbioru).
- × Pamiętaj o tym, że używanie niewielkich zbiorów prowadzi do uzyskania niestabilnych wyników, a więc nie będą one znaczące statystycznie lub nie doprowadzą do żadnych wniosków.
- × Unikaj zbiorów **zabawkowych** lub **syntetycznych**. Mogą być przydatne do zdobycia pewnej intuicji wobec kalibracji parametrów, ale zachowanie algorytmu może okazać się diametralnie różne przy danych wielowymiarowych.

Eksperymenty SI — wytyczne

4. Zaprojektuj eksperyment:

- × Wybierz liczbę replikacji (najczęściej zależy od wielkości zbioru).
- × Pamiętaj o tym, że używanie niewielkich zbiorów prowadzi do uzyskania niestabilnych wyników, a więc nie będą one znaczące statystycznie lub nie doprowadzą do żadnych wniosków.
- × Unikaj zbiorów **zabawkowych** lub **syntetycznych**. Mogą być przydatne do zdobycia pewnej intuicji wobec kalibracji parametrów, ale zachowanie algorytmu może okazać się diametralnie różne przy danych wielowymiarowych.
- × Staraj się wykorzystywać **rzeczywiste zbiory** zebrane w naturalnych warunkach. Laboratorium to Twój wróg.

Najlepsze zbiory rosną w dolinach otoczonych lasem.

Najlepsze leżą na ulicy, przykryte brudem miasta i rozjechane przez ruch samochodowy.

Myślałam, że takie dziewicze najlepsze.

Nietknięte złym dotykiem.

Zły dotyk jest bardziej realny niż dziewictwo.

Eksperymenty SI — wytyczne

5. Dokonaj kilku testowych przebiegów eksperymentu z losowymi parametrami przed uruchomieniem przeglądu z użyciem *grid search*.

Eksperymenty SI — wytyczne

5. Dokonaj kilku testowych przebiegów eksperymentu z losowymi parametrami przed uruchomieniem przeglądu z użyciem *grid search*.
6. Eksperymentator nie może być stronniczy podczas eksperymentu. Najlepiej, żeby algorytm testował ktoś inny niż jego twórca.

Eksperymenty SI — wytyczne

5. Dokonaj kilku testowych przebiegów eksperymentu z losowymi parametrami przed uruchomieniem przeglądu z użyciem *grid search*.
6. Eksperymentator nie może być stronniczy podczas eksperymentu. Najlepiej, żeby algorytm testował ktoś inny niż jego twórca.
7. Zamiast programować wszystko samodzielnie, używaj oprogramowania ze źródeł godnych zaufania (nie dość, że masz mniej pracy, to jeszcze będzie lepiej zoptymalizowane i przetestowane).

Eksperymenty SI — wytyczne

8. Jeśli decydujesz się na analizę statystyczną (a wypadałoby), sformułuj swoje cele w statystyczny sposób

Eksperymenty SI — wytyczne

8. Jeśli decydujesz się na analizę statystyczną (a wypadałoby), sformułuj swoje cele w statystyczny sposób, przykładowo:

Poprawnie

”Czy możemy stwierdzić, że średni błąd klasyfikatora wytrenowanego algorytmem A jest znacząco niższy niż średni błąd klasyfikatora wytrenowanego przez algorytm B?”

zamiast

Nędznie

„Czy A jest dokładniejsze od B?”

9. Pamiętaj, że testy statystyczne **NIGDY, PRZENIGDY** nie powiedzą nam, czy hipoteza jest poprawna czy nie, ale jedynie czy **wyduje się prawdopodobna**.

Eksperymenty SI — wytyczne

9. Pamiętaj, że testy statystyczne **NIGDY, PRZENIGDY** nie powiedzą nam, czy hipoteza jest poprawna czy nie, ale jedynie czy **wyduje się prawdopodobna**.
10. Zawsze istnieje ryzyko, że nie uzyskamy sensownych wniosków, lub, iż nasze wnioski są błędne (szczególnie przy niewielkim i zaszumionym zbiorze danych).

Eksperymenty SI — wytyczne

9. Pamiętaj, że testy statystyczne **NIGDY, PRZENIGDY** nie powiedzą nam, czy hipoteza jest poprawna czy nie, ale jedynie czy **wyduje się prawdopodobna**.
10. Zawsze istnieje ryzyko, że nie uzyskamy sensownych wniosków, lub, iż nasze wnioski są błędne (szczególnie przy niewielkim i zaszumionym zbiorze danych).
11. Wszystkie wnioski wyciągnięte z eksperymentów są uwarunkowane przez testowane przez nas zbiory danych.



Walidacja krzyżowa



Walidacja krzyżowa

↪ Dla spełnienia zasady **replikacji** musimy uzyskać wiele \mathcal{TS} i \mathcal{VS} ze zbioru \mathcal{DS} .

Walidacja krzyżowa

- ↪ Dla spełnienia zasady **replikacji** musimy uzyskać wiele \mathcal{TS} i \mathcal{VS} ze zbioru \mathcal{DS} .
- ↪ Jeśli \mathcal{DS} jest wystarczająco duży, możemy podzielić go na k części, a każdą z nich na \mathcal{TS} i \mathcal{VS} .

Walidacja krzyżowa

- ↪ Dla spełnienia zasady **replikacji** musimy uzyskać wiele \mathcal{TS} i \mathcal{VS} ze zbioru \mathcal{DS} .
- ↪ Jeśli \mathcal{DS} jest wystarczająco duży, możemy podzielić go na k części, a każdą z nich na \mathcal{TS} i \mathcal{VS} .
- ↪ Niestety \mathcal{DS} nigdy nie jest wystarczająco duży. Wykorzystujemy więc te same dane, losowo podzielone k razy. To właśnie jest **walidacja krzyżowa** (*cross-validation*).

Stratyfikacja

↗ Chcielibyśmy aby \mathcal{TS} i \mathcal{VS} były możliwie duże, ponieważ tylko tak możemy zagwarantować sobie solidność estymacji błędu. Jednocześnie, chcemy aby zbiory były *możliwie rozłączne*.

Stratyfikacja

- ↗ Chcielibyśmy aby \mathcal{TS} i \mathcal{VS} były możliwie duże, ponieważ tylko tak możemy zagwarantować sobie solidność estymacji błędu. Jednocześnie, chcemy aby zbiory były *możliwie rozłączne*.
- ↗ Klasy powinny być **reprezentowane w tych samych proporcjach w każdym z podzbiorów**. A więc staramy się nie zaburzać prawdopodobieństwa *a priori*.

k-fold cross-validation

Podziel \mathcal{DS} losowo na k **równych** podzbiorów:

$$\mathcal{VS}_1 = \mathcal{DS}_1$$

$$\mathcal{TS}_1 = \mathcal{DS}_2 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k$$

$$\mathcal{VS}_2 = \mathcal{DS}_2$$

$$\mathcal{TS}_2 = \mathcal{DS}_1 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k$$

$$\vdots$$

$$\vdots$$

$$\mathcal{VS}_k = \mathcal{DS}_k$$

$$\mathcal{TS}_k = \mathcal{DS}_1 \cup \mathcal{DS}_2 \cup \dots \cup \mathcal{DS}_{k-1}$$

k-fold cross-validation

Podziel \mathcal{DS} losowo na k **równych** podzbiorów:

$$\mathcal{VS}_1 = \mathcal{DS}_1$$

$$\mathcal{TS}_1 = \mathcal{DS}_2 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k$$

$$\mathcal{VS}_2 = \mathcal{DS}_2$$

$$\mathcal{TS}_2 = \mathcal{DS}_1 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k$$

$$\vdots$$

$$\vdots$$

$$\mathcal{VS}_k = \mathcal{DS}_k$$

$$\mathcal{TS}_k = \mathcal{DS}_1 \cup \mathcal{DS}_2 \cup \dots \cup \mathcal{DS}_{k-1}$$

$\leadsto k$ standardowo wynosi 10 lub 30.

k-fold cross-validation

Podziel \mathcal{DS} losowo na k **równych** podzbiorów:

$$\begin{array}{ll} \mathcal{VS}_1 = \mathcal{DS}_1 & \mathcal{TS}_1 = \mathcal{DS}_2 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k \\ \mathcal{VS}_2 = \mathcal{DS}_2 & \mathcal{TS}_2 = \mathcal{DS}_1 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k \\ \vdots & \vdots \\ \mathcal{VS}_k = \mathcal{DS}_k & \mathcal{TS}_k = \mathcal{DS}_1 \cup \mathcal{DS}_2 \cup \dots \cup \mathcal{DS}_{k-1} \end{array}$$

↪ k standardowo wynosi 10 lub 30.

↪ Jeśli \mathcal{DS} jest duże, k może być niewielkie.

k-fold cross-validation

Podziel \mathcal{DS} losowo na k **równych** podzbiorów:

$$\begin{array}{ll} \mathcal{VS}_1 = \mathcal{DS}_1 & \mathcal{TS}_1 = \mathcal{DS}_2 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k \\ \mathcal{VS}_2 = \mathcal{DS}_2 & \mathcal{TS}_2 = \mathcal{DS}_1 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k \\ \vdots & \vdots \\ \mathcal{VS}_k = \mathcal{DS}_k & \mathcal{TS}_k = \mathcal{DS}_1 \cup \mathcal{DS}_2 \cup \dots \cup \mathcal{DS}_{k-1} \end{array}$$

- ↪ k standardowo wynosi 10 lub 30.
- ↪ Jeśli \mathcal{DS} jest duże, k może być niewielkie.
- ↪ Jeśli \mathcal{DS} jest niewielkie, k powinno być wysokie.

k-fold cross-validation

Podziel \mathcal{DS} losowo na k **równych** podzbiorów:

$$\begin{array}{ll} \mathcal{VS}_1 = \mathcal{DS}_1 & \mathcal{TS}_1 = \mathcal{DS}_2 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k \\ \mathcal{VS}_2 = \mathcal{DS}_2 & \mathcal{TS}_2 = \mathcal{DS}_1 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k \\ \vdots & \vdots \\ \mathcal{VS}_k = \mathcal{DS}_k & \mathcal{TS}_k = \mathcal{DS}_1 \cup \mathcal{DS}_2 \cup \dots \cup \mathcal{DS}_{k-1} \end{array}$$

- ↪ k standardowo wynosi 10 lub 30.
- ↪ Jeśli \mathcal{DS} jest duże, k może być niewielkie.
- ↪ Jeśli \mathcal{DS} jest niewielkie, k powinno być wysokie.
- ↪ Skrajny przypadek tej walidacji to *one-leave-out*.

k-fold cross-validation

Podziel \mathcal{DS} losowo na k **równych** podzbiorów:

$$\begin{array}{ll} \mathcal{VS}_1 = \mathcal{DS}_1 & \mathcal{TS}_1 = \mathcal{DS}_2 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k \\ \mathcal{VS}_2 = \mathcal{DS}_2 & \mathcal{TS}_2 = \mathcal{DS}_1 \cup \mathcal{DS}_3 \cup \dots \cup \mathcal{DS}_k \\ \vdots & \vdots \\ \mathcal{VS}_k = \mathcal{DS}_k & \mathcal{TS}_k = \mathcal{DS}_1 \cup \mathcal{DS}_2 \cup \dots \cup \mathcal{DS}_{k-1} \end{array}$$

- ↪ k standardowo wynosi 10 lub 30.
- ↪ Jeśli \mathcal{DS} jest duże, k może być niewielkie.
- ↪ Jeśli \mathcal{DS} jest niewielkie, k powinno być wysokie.
- ↪ Skrajny przypadek tej walidacji to *one-leave-out*.
- ↪ Możemy powtórzyć tę procedurę wielokrotnie (na przykład *10x10-fold cross-validation*) aby uzyskać bardziej solidną estymację błędu.

5x2 cross-validation (Dietterich, 1998)

Dzielimy \mathcal{DS} losowo na dwie równe części. Pierwszej używamy do uczenia, drugiej do walidacji. Później zamieniamy je rolami. Kolejny fold otrzymujemy przez kolejne losowanie z \mathcal{DS} .

$$\mathcal{VS}_1 = \mathcal{DS}_1^{(1)}$$

$$\mathcal{TS}_1 = \mathcal{DS}_1^{(2)}$$

$$\mathcal{VS}_2 = \mathcal{DS}_1^{(2)}$$

$$\mathcal{TS}_2 = \mathcal{DS}_1^{(1)}$$

$$\vdots$$

$$\vdots$$

$$\mathcal{VS}_9 = \mathcal{DS}_5^{(1)}$$

$$\mathcal{TS}_9 = \mathcal{DS}_5^{(2)}$$

$$\mathcal{VS}_{10} = \mathcal{DS}_5^{(2)}$$

$$\mathcal{TS}_{10} = \mathcal{DS}_5^{(1)}$$

5x2 cross-validation (Dietterich, 1998)

Dzielimy \mathcal{DS} losowo na dwie równe części. Pierwszej używamy do uczenia, drugiej do walidacji. Później zamieniamy je rolami. Kolejny fold otrzymujemy przez kolejne losowanie z \mathcal{DS} .

$$\begin{array}{ll} \mathcal{VS}_1 = \mathcal{DS}_1^{(1)} & \mathcal{TS}_1 = \mathcal{DS}_1^{(2)} \\ \mathcal{VS}_2 = \mathcal{DS}_1^{(2)} & \mathcal{TS}_2 = \mathcal{DS}_1^{(1)} \\ \vdots & \vdots \\ \mathcal{VS}_9 = \mathcal{DS}_5^{(1)} & \mathcal{TS}_9 = \mathcal{DS}_5^{(2)} \\ \mathcal{VS}_{10} = \mathcal{DS}_5^{(2)} & \mathcal{TS}_{10} = \mathcal{DS}_5^{(1)} \end{array}$$

Moglibyśmy dokonać podziału więcej niż pięciokrotnie, ale Dietterich wykazał, że po pięciu podziałach zbiory współdzielą zbyt wiele wzorców. Tak statystyki wyliczone z takich podzbiorów stają się zależne. Ufajmy matematykom w przypadkach, w których przedstawiają nam dowody na to, że nie musimy pracować więcej niż to konieczne.


~~~~~  
A wkrótce testy.  
~~~~~