

Semi-supervised learning — Teoria

1. Problem semi-supervised learning

W klasycznej klasyfikacji nadzorowanej (supervised learning) uczymy model na podstawie danych:

- Macierzy cech: $X = (X_1, X_2, \dots, X_p)$
- oraz etykiet: $y \in \{0, 1\}$.

W semi-supervised learning zakładamy, że **część danych w zbiorze treningowym ma etykiety**, a część jest nieetykietowana. Typowo mamy:


- Dla niektórych próbek y_i jest znane (np. 0 lub 1).
- Dla reszty $y_i = -1$ (w notacji scikit-learn).

Cel:

- Wykorzystać zarówno etykietowane, jak i nieetykietowane dane do poprawy klasyfikacji.
-

2. Generowanie danych

W zadaniu generujemy dwa sztuczne zbiory danych:

 **make_circles:**

- Zbiór dwuklasowy w kształcie dwóch współśrodkowych okręgów (trudny problem nieliniowy).
- Parametry: `n_samples=1000`, `noise=0.1`.

✅ **make_classification:**

- Dwuwymiarowy problem liniowy.
- Parametry: `n_samples=1000`, `n_features=2`, `n_informative=2`.

Podział danych na zbiór treningowy i testowy realizujemy funkcją:

$$(X_{train}, X_{test}, y_{train}, y_{test}) = \text{train_test_split}(X, y, \text{test_size} = 0.3)$$

✦ 3. Etykietowanie

W semi-supervised learning przyjmujemy, że:

- Losowo wybieramy g przykładów z klasy pozytywnej i g przykładów z klasy negatywnej (w naszym zadaniu: klasy 0 i 1).
 - Oznaczamy je jako etykietowane.
 - Resztę przykładów traktujemy jako nieetykietowane (oznaczamy etykietą -1).
-

✦ 4. Metody klasyfikacji

Porównujemy 4 metody:

♦ (a) Naive Method

- Trenujemy model tylko na etykietowanych danych.
- Ignorujemy dane nieetykietowane.

♦ (b) Self-training

- Uczymy model na etykietowanych danych.
- Model przypisuje etykiety (pseudo-etykiety) najbardziej pewnym nieetykietowanym próbom.
- Te pseudo-etykiety są dodawane do zbioru treningowego w kolejnych iteracjach.

- ♦ (c) **Label Propagation**

- Traktuje dane jako graf podobieństw.
- Rozpoczyna od etykietowanych danych i propaguje etykiety w grafie na podstawie sąsiedztwa.

- ♦ (d) **Label Spreading**

- Podobny do Label Propagation, ale dodatkowo „wygładza” predykcje poprzez regularizację, co zwiększa stabilność.
-

✦ 5. Model bazowy

Jako model bazowy stosujemy:

- SVM z jądrem RBF (`SVC(probability=True, kernel='rbf')`).

Dla metod semi-supervised (`SelfTrainingClassifier`) bazowy klasyfikator jest owinięty wewnątrz w model.

✦ 6. Metryka oceny

Oceniamy dokładność (accuracy):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

gdzie:

- TP — True Positives
 - TN — True Negatives
 - FP — False Positives
 - FN — False Negatives
-

📌 7. Eksperyment

✅ Dla każdej wartości $g \in \{1, 2, 3, 4, 5\}$:

- Losujemy g przykładów z każdej klasy jako etykietowane.
- Uruchamiamy każdy model (4 metody).
- Obliczamy dokładność na zbiorze testowym.

✅ Powtarzamy eksperyment $n_{repeats}$ razy, aby uwzględnić losowość etykietowania i treningu.

📌 8. Analiza wyników

Wyniki przedstawiamy na wykresach:

- Oś x: g — liczba etykietowanych próbek na klasę.
- Oś y: dokładność.
- Boxplot pokazuje rozkład dokładności (wariancję).

Interpretacja:

- Dla małego g metody propagacyjne (Label Propagation, Label Spreading) powinny przewyższać metodę Naive.
- Self-training może wypadać gorzej, jeśli pseudo-etykiety są błędne.
- Zwiększanie g powinno prowadzić do wzrostu dokładności wszystkich metod.