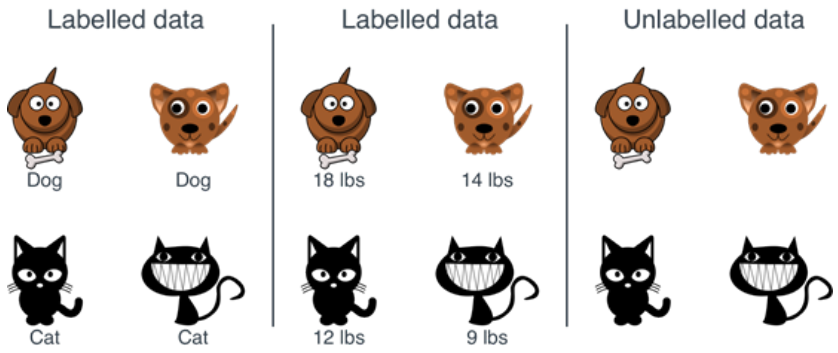
Wprowadzenie do Uczenia Częściowo Nadzorowanego

Jędrzej Sarna

31 października 2023

1 Wprowadzenie

Uczenie częściowo nadzorowane to dziedzina uczenia maszynowego, która próbuje rozwiązać problemy, w których występują zarówno dane z etykietami, jak i dane bez etykiet, wykorzystując koncepcje charakterystyczne zarówno dla metod klasteryzacji, jak i klasyfikacji.



Rysunek 1: Przykład danych z etykietami oraz bez etykiet

Wysoka dostępność próbek bez etykiet oraz trudność w poprawnym oznaczaniu ogromnych zbiorów danych skłoniły wielu badaczy do zbadania najlepszych podejść, które umożliwiają rozszerzenie wiedzy dostarczonej przez próbki z etykietami na większą populację "nieoznaczoną", bez utraty dokładności. Podczas tego referatu przedstawię bliżej ten dział uczenia maszynowego oraz poruszę takie kwestie jak:

- Częściowo nadzorowany scenariusz
- Różne podejścia do uczenia częściowo nadzorowanego
- Założenia potrzebne do poprawnego operowania w danych scenariuszach

Następnie przejdziemy do przedstawienia kilku algorytmów uczenia częściowo nadzorowanego i pokażemy przykłady ich praktycznego zastosowania w języku Python. Algorytmy jakie omówimy to:

- The Generative Gaussian Mixture algorithm
- Self-Training
- Co-Training

2 Częściowo nadzorowany scenariusz

2.1 Proces Generowania Danych (Data Generating Process)

Załóżmy, że mamy pewien proces generowania danych p, dla którego mamy zależność:

$$p(\overline{x}, \overline{y}) = p(\overline{y}|\overline{x})p(\overline{x}) \text{ lub } p(\overline{x}|\overline{y})p(\overline{y})$$

Jednak w przeciwieństwie do podejścia nadzorowanego, w którym możemy polegać na całkowicie oznakowanym zbiorze danych, mamy tylko ograniczoną liczbę N punktów danych pobranych z p z etykietą, jak poniżej:

$$\mathbf{X}_L = \{\overline{x}_0^L, \overline{x}_1^L, \ldots, \overline{x}_N^L\} \text{ gdzie } \overline{x}_i^L \in \mathbb{R}^p$$
$$\mathbf{Y}_L = \{\overline{y}_0^L, \overline{y}_1^L, \ldots, \overline{y}_N^L\} \text{ gdzie } \overline{y}_i^L \in \mathbb{R}^q$$

Podobnie jak w przypadku innych metod, zakłada się, że zbiór treningowy jest losowany jednostajnie, aby nie wykluczyć żadnego regionu rozkładu p. Gdy ten warunek jest spełniony, możliwe jest rozważenie większej liczby (M) nieoznakowanych próbek wylosowanych z rozkładu brzegowego $p(\overline{x})$:

$$\mathbf{X}_U = \{\overline{x}_0^U, \overline{x}_1^U, \ldots, \overline{x}_M^U\} \text{ gdzie } \overline{x}_i^U \in \mathbb{R}^p$$

Kontekst uczenia częściowo nadzorowanego jest następnie definiowany przez sumę dwóch zbiorów $\{\mathbf{X}_L, \mathbf{Y}_L\}$ i $\mathbf{X}_U$. Ważnym założeniem dotyczącym nieoznakowanych próbek jest to, że ich rozkład, który nie różni się dramatycznie od oznakowanego pod względem równowagi klas.

W ogólnym przypadku nie ma ograniczeń co do wartości N i M, jednakże, problem częściowo nadzorowany zwykle pojawia się, gdy liczba nieoznakowanych punktów jest (znacznie) większa niż moc oznaczonego zbioru. Jeśli możemy wylosować $N >> M$ oznakowanych punktów z p, to prawdopodobnie nie ma sensu kontynuować pracy z metodami pół-nadzorowanymi, a klasyczne metody nadzorowane będą prawdopodobnie najlepszym wyborem. Dodatkowa złożoność, której potrzebujemy, jest uzasadniona przez $M >> N$, co jest powszechnym warunkiem we wszystkich sytuacjach, w których ilość dostępnych nieoznakowanych danych jest duża, a liczba poprawnie liczba poprawnie oznaczonych próbek jest znacznie niższa.

Jako podstawową zasadę, możemy uznać zależność, że jeśli wiedza o $\mathbf{X}_U$ zwiększa naszą wiedzę na temat wcześniejszego rozkładu $p(\overline{x})$, to algorytm pół-nadzorowany będzie prawdopodobnie działał lepiej niż jego nadzorowany - a zatem ograniczony do $\mathbf{X}_L$ - odpowiednik. Z drugiej strony, jeśli punkty bez etykiet pochodzą z różnych rozkładów lub z regionów danych wykluczonych z procesu uczenia, wynik końcowy może być znacznie gorszy.

W rzeczywistych przypadkach nie ma sposobu, aby od razu zrozumieć, czy algorytm półnadzorowany jest najlepszym wyborem. Dlatego też walidacja krzyżowa i wszelakie porównania są najlepszymi praktykami do zastosowania podczas oceny scenariusza. Powinno być również jasne, że podczas gdy w scenariuszu nadzorowanym jesteśmy bezpośrednio zainteresowani warunkowym rozkładem prawdopodobieństwa $p(\overline{y}|\overline{x})$ i możemy pozbyć się $p(\overline{x})$, w scenariuszu pół-nadzorowanym często jesteśmy zmuszeni do modelowania $p(\overline{x})$ w celu wykorzystania próbek bez etykiet. Taki problem można również przeanalizować w inny sposób, który ujawnia kolejne ograniczenie uczenia pół-nadzorowanego.

2.2 Scenariusz przyczynowy oraz antyprzyczynowy

Ponieważ uczenie częściowo nadzorowane (SSL) próbuje wykorzystać informacje z $p(\overline{x})$, aby pomóc w przewidywaniu Y na podstawie X, należy rozważyć przypadki kiedy wiedza o $p(\overline{x})$ wpływa na wiedzę o $p(\overline{y}|\overline{x})$. Załóżmy z góry, że zbiór X jest zbiorem przyczyn, a Y jest zbiorem skutków.

2.2.1 Przewidywanie skutku na podstawie przyczyny (scenariusz przyczynowy)

Rozważmy przypadek, w którym próbujemy oszacować rozkład warunkowy $p(\overline{y}|\overline{x})$. Czy możemy założyć, że dodatkowa wiedza o $p(\overline{x})$ zwiększa wiedzę o $p(\overline{y}|\overline{x})$ w kontekście nauczania częściowo nadzorowanego? Jeśli modelujemy warunkowy rozkład skutków biorąc pod uwagę zestaw przyczyn, wszystkie informacje potrzebne do podjęcia decyzji, która przyczyna jest najbardziej prawdopodobna, są już zakodowane w modelu. Dzieje się tak, ponieważ taki proces jest regulowany tylko przez znajomość danych $p(x)$, które wywołały wszystkie efekty $p(y)$ (zatem danych oznakowanych). Rozważmy prosty przykład:

Wyobraź sobie, że próbujesz przewidzieć wynik egzaminu ucznia (y) na podstawie liczby godzin, które spędził na naukę (x). Zebrałeś dane od kilku uczniów i stworzyłeś model reprezentujący zależność między liczbą godzin nauki a wynikami egzaminów. Twój model jest reprezentowany jako p(y | x).

Dane:

- Uczeń A uczył się przez 2 godziny i zdobył 80 punktów.
- Uczeń B uczył się przez 4 godziny i zdobył 90 punktów.
- Uczeń C uczył się przez 6 godzin i zdobył 95 punktów.

Na podstawie tych danych stworzyłeś model, który oddaje zależność między liczbą godzin nauki (x) a wynikami egzaminów (y). Twój model może wyglądać mniej więcej tak:

p(y | x) = y = 10x + 60

Teraz pytanie brzmi, czy wiedza na temat rozkładu liczby godzin nauki (p(x)) znacząco poprawi zdolność do przewidywania wyników egzaminów (y). W tym przypadku odpowiedź brzmi "nie". Model p(y | x) już zawiera zależność między liczbą godzin nauki a wynikami egzaminów. Rozkład godzin nauki (p(x)) nie dostarcza dodatkowych informacji, które zmieniłyby przewidywania.

2.2.2 Przewidywanie przyczyny na podstawie skutku (scenariusz antyprzyczynowy)

Zwrócimy się teraz w przeciwnym kierunku, gdzie szukamy $p(\overline{x}|\overline{y})$. Ta sytuacja, którą nazywamy scenariuszem antyprzyczynowym, może wydawać się nienaturalna, ale jest właściwie wszechobecna w uczeniu maszynowym.

Rozważmy jak poprzednio zadanie przewidywania wyniku egzaminu (y) na podstawie długości nauki. Scenariusz jest zatem następujący: osoba zamierza napisać na dany wynik, a ta intencja powoduje naukę przez określoną ilość czasu - w tym sensie y powoduje x. Czyli u nas $p(x|y)$ reprezentuje mechanizm przyczynowy, który generuje (x) z (y) , i jest niezależny od rozkładu przyczyny $p(y)$ (zgodnie z poprzednim przykładem). Z drugiej strony, $p(y|x)$ jest wrażliwe na zmianę rozkładu $p(y)$. Dlaczego?

Załóżmy, że w poprzednich latach uczniowie mieli zazwyczaj wyniki egzaminów (y) skupione wokół średniej wartości. Jednak w obecnym roku niespodziewanie uczniowie zdobywają wyższe wyniki, a rozkład p(y) przesuwa się w kierunku wyższych wyników. W takim przypadku, nawet jeśli wcześniejszy model p(y|x) jest poprawny (wynik rośnie wraz z liczbą godzin nauki), nowy rozkład p(y) wpłynie na predykcje modelu. Model będzie przewidywał wyższe wyniki, ponieważ rozkład wyników p(y) się zmienił.

Dlatego, ogólnie rzecz biorąc, podczas szacowania p(y|x), lepiej byłoby najpierw modelować p(x|y), a następnie skonstruować p(y|x) przy użyciu reguły Bayesa $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$

Większość przykładów omówionych w tym rozdziale opiera się na tym założeniu. Często uważamy klasę za przyczynę, a atrybuty za efekty ((tzn. fakt, że kwiat należy do danej klasy w zbiorze danych Iris, determinuje określony zestaw cech, takich jak długość płatka, szerokość i tym podobne).

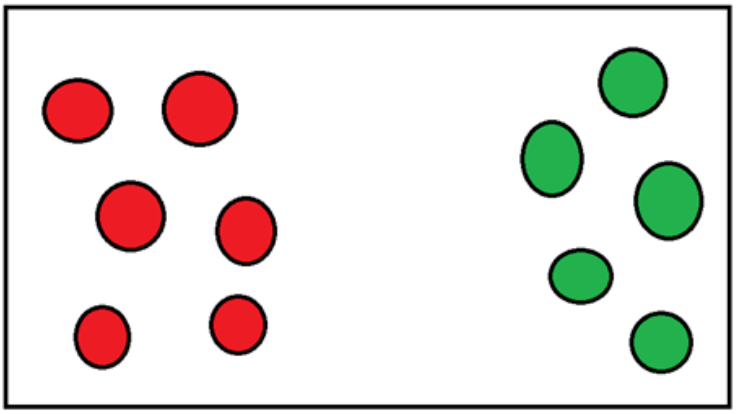
3 Indukcja oraz Transdukcja - różne podejścia uczenia się

3.1 Inductive learning (Indukcja)

Uczenie indukcyjne to nic innego jak zasada stojąca za nadzorowanymi algorytmami uczenia maszynowego, w których model próbuje zbudować związek między zmiennymi niezależnymi a zmienną zależną, badając ukryte wzorce w danych treningowych. Model w tym przypadku uczy się na podstawie ograniczonego zakresu danych treningowych (jedynie dane oznakowane), aby mógł przewidzieć wartość dowolnego punktu danych z nieoznakowanego zbioru danych (testowego zbioru danych). Należy tutaj zauważyć, że model nie jest narażony na dane testowe (dane bez etykiet) podczas fazy uczenia się i otrzymuje jedynie dane treningowe do celów uczenia się.

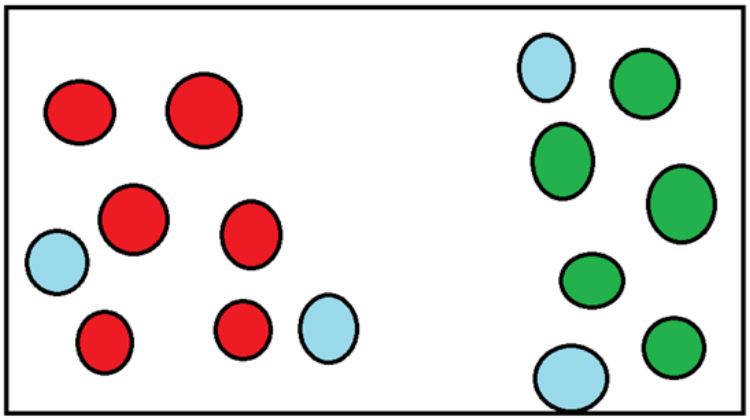
3.1.1 Inductive learning (Indukcja) - przykład

Wyobraźmy sobie, że mamy problem klasyfikacji, w którym musimy przewidzieć, czy dana osoba jest mężczyzną czy kobietą. Mamy następujący treningowy zbiór danych -

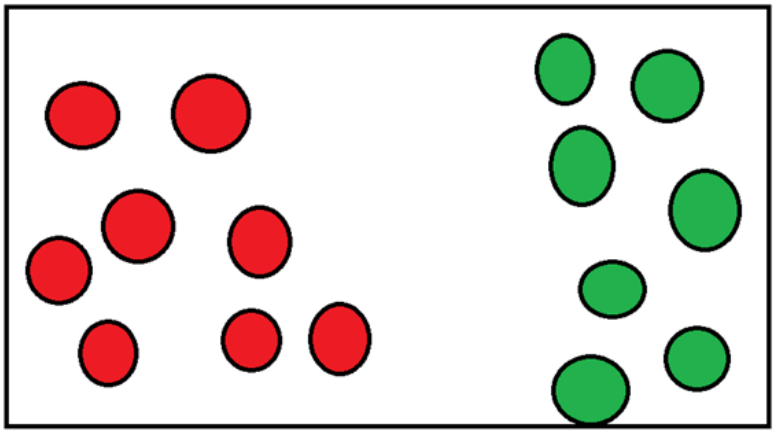


Zielone kółka oznaczają mężczyzn, a czerwone - kobiety. Na tych danych wytrenowaliśmy model uczenia maszynowego, a następnie wprowadziliśmy punkty danych testowych.

Niebieskie okręgi to punkty danych testowych -



Ponieważ model został już 'nauczony' za pomocą algorytmu uczenia maszynowego, możemy użyć tego modelu do przewidywania tych nowych punktów danych. Otrzymaliśmy następujące wyniki



To nic innego jak klasyczny przykład nauczania indukcyjnego.

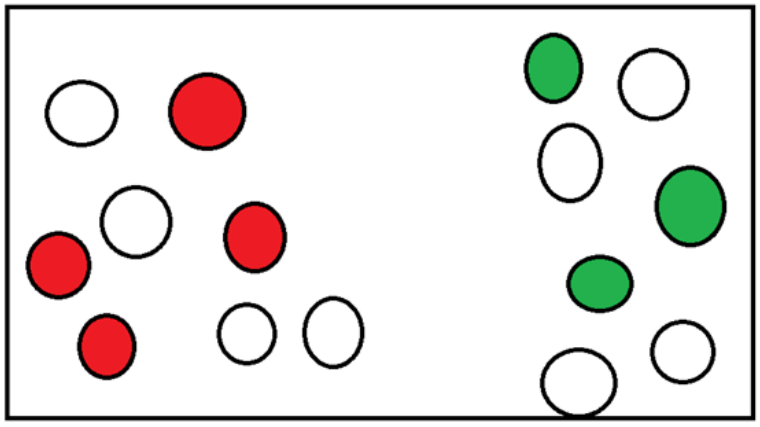
3.2 Transductive learning (Transdukcja)

W uczeniu transdukcyjnym zarówno treningowy, jak i testowy zestaw danych jest wystawiony na działanie modelu w samej fazie uczenia. Model próbuje znaleźć jakiekolwiek informacje o wzorcu w połączonym zbiorze danych (trening + testowanie), a następnie wykorzystuje te informacje do przewidywania wartości nieoznakowanych punktów danych testowych.

3.2.1 Transductive learning (Transdukcja) - przykład

Rozważmy, że mamy pół-nadzorowany problem przewidywania, czy dana osoba jest mężczyzną czy kobietą. Problem z pół-nadzorowanym uczeniem się polega na tym, że wszystkie punkty danych w zbiorze danych nie będą oznaczone.

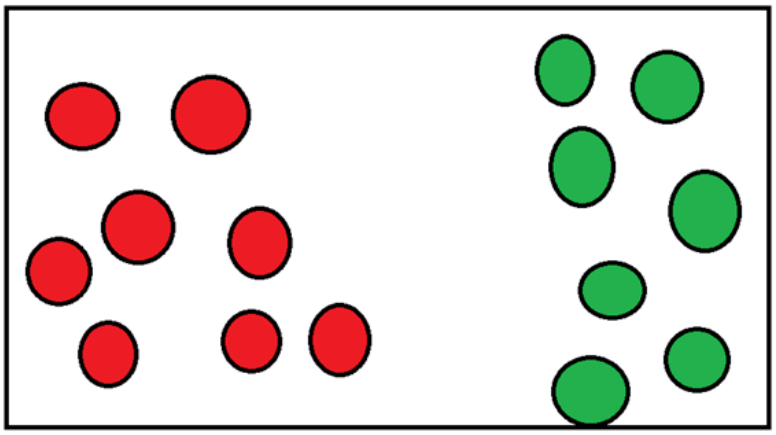
Wyobraźmy sobie, że nasze punkty danych wyglądają następująco-



Tutaj widzimy, że niektóre punkty danych są oznaczone jako mężczyźni (kolor zielony), a niektóre punkty danych są oznaczone jako kobiety (kolor czerwony), ale są też punkty danych, które pozostają nieoznaczone (kolor biały). Zamiast dzielić dane na zbiór uczący i testowy, jak w podejściu uczenia indukcyjnego, możemy zbudować model w oparciu o cały zbiór danych, wykorzystując wszystkie dostępne punkty danych (oznaczone i nieoznaczone).

Istnieje kilka algorytmów do tego celu, jednak w tym referacie skupimy się jedynie na samej idei uczenia transdukcyjnego.

Na koniec otrzymamy następujące dane wyjściowe-



3.3 Indukcja oraz Transdukcja - podsumowanie

W tym momencie możemy krótko podsumować dwa poznane rodzaje nauczania:

- Uczenie indukcyjne trenuje model z oznaczonymi punktami danych i próbuje przewidzieć etykietę nieoznaczonych punktów danych. Uczenie transdukcyjne trenuje cały zestaw danych i próbuje przewidzieć etykietę nieoznakowanych punktów danych.
- W uczeniu indukcyjnym, jeśli wprowadzany jest nowy nieoznakowany punkt danych, możemy użyć już wytrenowanego modelu do przewidywania. W przypadku uczenia transdukcyjnego może być konieczne ponowne przeszkolenie całego modelu.
- Transdukcyjne uczenie jest więc bardziej kosztowne obliczeniowo niż uczenie indukcyjne

4 Założenia dla uczenia częściowo nadzorowanego

Jak wyjaśniono wcześniej, pół-nadzorowane uczenie nie gwarantuje poprawy modelu nadzorowanego. Zły wybór może doprowadzić do dramatycznego pogorszenia wydajności. Możliwe jest jednak określenie kilku podstawowych założeń, które są wymagane, aby uczenie częściowo nadzorowane działało poprawnie. Nie zawsze są to matematycznie udowodnione twierdzenia, ale raczej obserwacje empiryczne, które uzasadniają całkowicie arbitralny wybór podejścia.

4.1 Założenie gładkości (jednostajna ciągłość)

Rozważmy funkcję o wartościach rzeczywistych $f(x)$ i odpowiadające jej przestrzenie metryczne X i Y. Mówi się, że taka funkcja spełnia warunek Lipschitza, jeśli:

$$\exists K \; \forall x_1, x_2 \in X \Rightarrow d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$$

Innymi słowy, jeśli dwa punkty x_1 i x_2 znajdują się blisko siebie, odpowiadające im wartości wyjściowe y_1 i y_2 nie mogą być arbitralnie oddalone od siebie. Warunek ten ma fundamentalne znaczenie w problemach regresji, gdzie uogólnienie jest często wymagane dla punktów, które znajdują się pomiędzy próbkami treningowymi. Na przykład, jeśli musimy przewidzieć wynik dla punktu $x_t : x_1 < x_t < x_2$, a regresor spełnia warunek Lipschitza, możemy być pewni, że y_t będzie poprawnie ograniczony przez y_1 i y_2. W uczeniu częściowo nadzorowanym pomocne jest dodanie wyraźnego ograniczenia (związanego

z założeniem klastra, o którym dokładniej później): jeśli dwa punkty znajdują się w regionie o dużej gęstości (klastrze) i są blisko siebie, to odpowiadające im dane wyjściowe również muszą być blisko siebie. W bardziej formalny sposób, założenie można wyrazić jako:

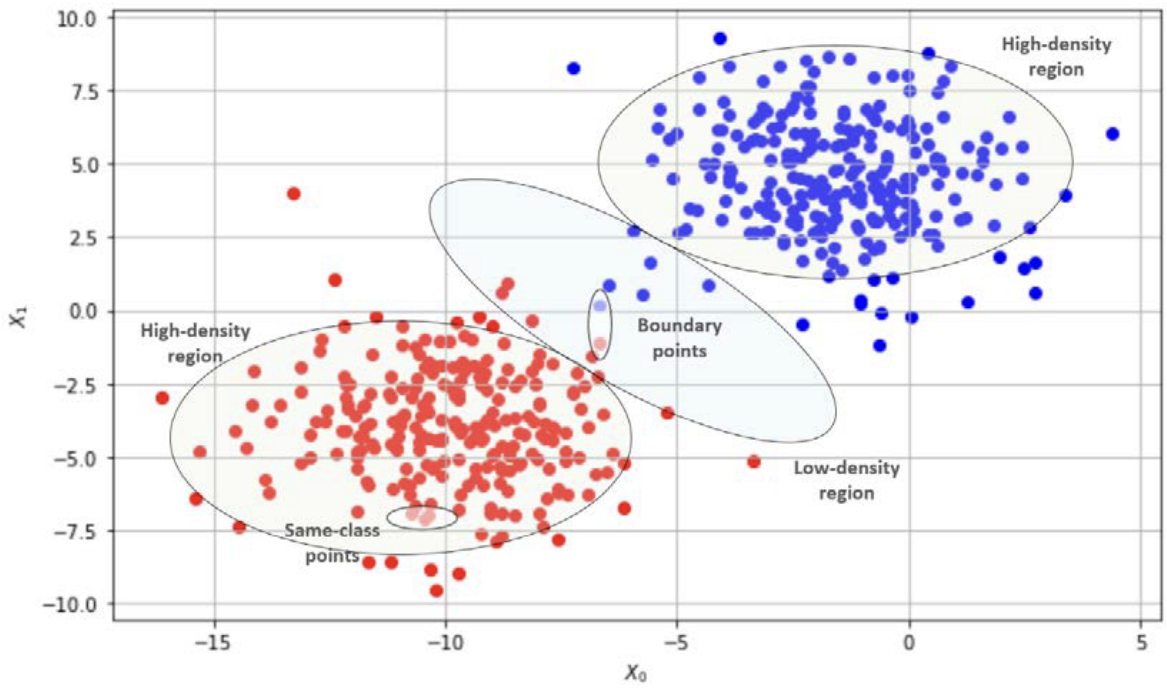
$$\text{Jeżeli } f(\bar{x}_c;\bar{\theta}) = y_c, \;\; \exists\delta > 0 \; \forall \bar{x} \in X : \; d(\bar{x},\bar{x}_c) < \delta \Rightarrow f(\bar{x};\bar{\theta}) = y_c$$

W tym wzorze $f(\bar{x};\bar{\theta})$ jest ogólnym klasyfikatorem parametrycznym. Stąd, biorąc pod uwagę punkt $\bar{x}_c$, który jest klasyfikowany jako y_c, istnieje kula, w której wszystkie punkty zostaną sklasyfikowane w ten sam sposób. Definicja ta nie nakłada żadnych ograniczeń na δ , ale dla naszych celów musimy założyć, że istnieją dwa marginesy większe od zera (δ_m i δ_M), aby wprowadzić zarówno dolną, jak i górną granicę dla δ ($\delta_m < \delta < \delta_M$). W ten sposób ograniczamy rodzinę funkcji do zbioru stosunkowo "wolno zmieniających się".

W kontekście uczenia półnadzorowanego założenie gładkości odgrywa fundamentalną rolę, ponieważ jeśli dwie próbki znajdują się w regionie o niskiej gęstości, mogą należeć do różnych klastrów, a ich etykiety mogą być bardzo różne. Nie zawsze tak jest, ale warto uwzględnić to ograniczenie, aby umożliwić dalsze założenia w wielu definicjach modeli pół-nadzorowanych.

4.2 Założenie klastra

To założenie jest ściśle powiązane z poprzednim. Można je wyrazić za pomocą sekwencji współzależnych warunków. Klastry są regionami o dużej gęstości (High-density regions), dlatego jeśli dwa punkty są blisko siebie, prawdopodobnie należą do tego samego klastra, a ich etykiety muszą być takie same. Regiony o niskiej gęstości (Low-density regions) są przestrzeniami rozdzielającymi, dlatego próbki należące do regionu o niskiej gęstości są prawdopodobnie punktami granicznymi, a ich klasy mogą być różne. Aby lepiej zrozumieć tę koncepcję,rozważmy następujący dwuwymiarowy przykład:



Rysunek 2: Reprezentacja dwóch oddzielnych dwuwymiarowych klastrów

W scenariuszu pół-nadzorowanym nie moglibyśmy znać etykiety punktu należącego do regionu o wysokiej gęstości. Jeśli jednak jest on na tyle blisko oznaczonego punktu, że możliwe jest zbudowanie kuli, w której wszystkie punkty mają taką samą średnią gęstość, wówczas możemy przewidzieć etykietę naszej próbki testowej. Jeśli zamiast tego przejdziemy do regionu o niskiej gęstości, proces staje się trudniejszy, ponieważ dwa punkty mogą znajdować się bardzo blisko, ale z różnymi etykietami.

4.3 Założenie o rozmaitości n-wymiarowej

Jest to najmniej intuicyjne założenie, ale może być niezwykle przydatne do zmniejszenia złożoności wielu problemów. Upoważnia ono nas do zastosowania metod redukcji wymiarowości w celu uniknięcia klątwy wymiarowości (curse of dimensionality). W zakresie uczenia maszynowego główną konsekwencją takiego efektu jest to, że gdy wymiarowość próbek wzrasta, w celu osiągnięcia wysokiej dokładności, konieczne jest użycie coraz większej liczby próbek. Co więcej, zostało zaobserwowane, że dokładność klasyfikatorów statystycznych jest odwrotnie proporcjonalna do wymiarowości próbek (liczby cech).

To założenie jest zatem słuszne nie tylko dla uczenia częściowo nadzorowanego ale również dla innych rodzaji uczenia maszynowego. Aby dokładniej zrozumieć działanie założenia rozważmy przykład.

W przypadku wielu realistycznych zadań (np. określania jaką cyfrę piksele na obrazie przedstawiają), na świecie dostępnych jest znacznie więcej danych bez etykiet (np. obrazów, które mogą zawierać cyfry) niż z etykietami (np. obrazów, których tytułem jest cyfra) (przypadek uczenia częściowo nadzorowanego). Możemy jednak powiedzieć, że obrazy nie są w rzeczywistości próbkowane z jednostajnego rozkładu konfiguracji pikseli (nie są kompletnie losowe), więc wydaje się prawdopodobne, że istnieją pewna rozmaitości, która oddaje strukturę obrazów (przykładowo kształty, łuki). Jeśli założymy, że obrazy zawierające liczbę 4 leżą na swojej własnej rozmaitości, podczas gdy obrazy zawierające liczbę 5 leżą na innej rozmaitości, to możemy spróbować opracować reprezentacje dla każdej z tych rozmaitości przy użyciu tylko danych pikseli. Następnie, gdy mamy kilka próbek danych z etykietami, możemy użyć tych próbek, do etykietowania już zidentyfikowanych rozmaitości.

Założenie o rozmaitości jest również związane z technikami redukcji wymiarowości, takimi jak PCA (Analiza Składowych Głównych). Te techniki pozwalają na zredukowanie wysokowymiarowych danych do niższego wymiaru, zachowując jednocześnie istotne cechy struktury danych. W przypadku problemu rozpoznawania cyfr, PCA identyfikuje strukturę niskowymiarową (rozmaitość), która wychwytuje podstawowe cechy odróżniające jedną cyfrę od drugiej.

Po zdefiniowaniu scenariuszy odpowiednich dla nauczania częściowo nadzorowanego oraz odpowiednich założen, możemy rozpocząć badanie niektórych praktycznych algorytmów, które opierają się zarówno na oznaczonych, jak i nieoznaczonych zbiorach danych w celu przeprowadzania dokładniejszych klasyfikacji.

5 Generative Gaussian Mixture

Pierwszy model, który omówimy, nosi nazwę Generative Gaussian Mixture i ma na celu modelowanie procesu generowania danych p przy użyciu sumy ważonych rozkładów normalnych. Jego struktura pozwala nam nie tylko grupować istniejący zbiór danych w dobrze zdefiniowane klasy (reprezentowane jako gaussiany), ale także wyprowadzać prawdopodobieństwo przynależności każdego nowego punktu danych do każdej z klas. Model ten jest bardzo elastyczny i może być stosowany do rozwiązywania wszystkich problemów, w których konieczne jest jednoczesne przeprowadzenie klasteryzacji i klasyfikacji, uzyskując wektor prawdopodobieństwa przypisania, który określa prawdopodobieństwo generowania danego punktu przez konkretny (jeden ze zdefiniowanych) rozkładów normalnych. (scenariusz antyprzyczynowy)

5.1 Generative Gaussian Mixture - teoria

Generative Gaussian Mixture to algorytm dla częściowo nadzorowanej klasyfikacji i klasteryzacji, którego celem jest modelowanie rozkładów w poszczególnych klasach, które można wykorzystać do wyliczenia łącznego rozkładu $p(\overline{x}, \overline{y})$ (procesu generowania danych) mając zarówno etykietowany, jak i nieetykietowany zbiór danych (mamy tutaj do czynienia z nauczaniem transdukcyjnym).

Modele Generative Gaussian Mixture są bardzo pomocne, gdy konieczne jest znalezienie modelu, który wyjaśnia strukturę istniejących punktów danych, a ponadto ma możliwość wyprowadzania prawdopodobieństwa nowych punktów danych. Na przykład, system wykrywania anomalii może być mo-

delowany począwszy od zbioru danych normalnych i złośliwych działań. Generative Gaussian Mixture będzie w stanie je rozróżnić i odpowiedzieć na pytanie "Czy nowy punkt danych reprezentuje aktywność normalną czy złośliwą?", podając prawdopodobieństwo obu przypadków. Załóżmy, że mamy oznakowany zbiór danych X_l, Y_l zawierający N punktów danych i nieoznakowany zbiór danych X_u zawierający $M >> N$ punktów. Nie jest konieczne, aby $M >> N$, ale chcemy stworzyć prawdziwy scenariusz półnadzorowany, z tylko kilkoma oznakowanymi próbkami. Naszym celem jest określenie kompletnego rozkładu $p(\overline{x}, \overline{y})$ przy użyciu modelu generatywnego, a następnie uzyskanie rozkładu warunkowego $p(\overline{y}|\overline{x})$. Wykorzystajmy wielowymiarowy rozkład normalny do modelowania naszych danych:

$$f(\overline{\mathbf{x}}; \overline{\boldsymbol{\mu}}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\overline{\mathbf{x}} - \overline{\boldsymbol{\mu}})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \overline{\boldsymbol{\mu}})\right)$$

W związku z tym nasze parametry modelu są średnimi i macierzami kowariancji dla wszystkich gaussianów. Teraz konieczne jest zdefiniowanie liczby oczekiwanych gaussianów (która jest znana z oznaczonych próbek) oraz wektora wag, który reprezentuje prawdopodobieństwo przynależności do określonego gaussianu:

$$\bar{w} = (p(y = 1), p(y = 2), \ldots, p(y = q))$$

Możemy uzyskać wyrażenie na rozkład punktów X biorąc pod uwagę wektor parametrów $\bar{\theta}$ i wektor wag $\bar{w}$ (jest to nasz rozkład łączny $p(\overline{x}, \overline{y})$)(z twierdzenia o prawdopodobieństwie całkowitym):

$$p(\bar{x}_j; \bar{\theta}, \bar{w}) = \sum_{i=1}^{q} p(y_i)p(\bar{x}_j|y_i; \bar{\theta}) = \sum_{i=1}^{q} w_i p(\bar{x}_j|y_i; \bar{\theta})$$

W ten sposób łatwo jest zrozumieć rolę każdego gaussiana w określaniu prawdopodobieństwa nowego punktu. Dzięki temu rozróżnieniu możemy rozważyć funkcję logarytmu funkcji wiarygodności:

$$L(\bar{\theta}, \bar{w}) = \log \prod_{j=1}^{N} p(\bar{x}_j; \bar{\theta}, \bar{w}) = \sum_{j=1}^{N} \log \sum_{i=1}^{q} w_i p(\bar{x}_j|y_i; \bar{\theta})$$

Jak wyestymować teraz parametry naszego rozkładu? Wprowadzamy algorytm EM (Expectation - Maximization) znajdujący parametry maksymalizujące funkcję wiarygodności. Schemat algorytmu wygląda następująco:

1. Zainicjuj parametry (losowo lub na podstawie wcześniejszej wiedzy)
2. Krok E: oszacuj $p(y_i|\bar{x}_j; \bar{\theta}, \bar{w})$.
   Jak to zrobić? Z twierdzenia Bayesa mamy:

   $$p(y_i|\bar{x}_j; \bar{\theta}, \bar{w}) = \frac{p(y_i)p(\bar{x}_j|y_i; \bar{\theta})}{p(\bar{x}_j; \bar{\theta}, \bar{w})} = \frac{w_i p(\bar{x}_j|y_i; \bar{\theta})}{\sum_{i=1}^{q} w_i p(\bar{x}_j|y_i; \bar{\theta})} = \varphi_i(x_j)$$

   Dlatego też inaczej nazywa się ten krok oszacowaniem ukrytej zmiennej (latent variable), gdyż trzeba się jej "doszukać". Należy również rozpatrzyć dwa przypadki:

   - W przypadku próbek nieoznaczonych jest ona obliczana przez pomnożenie i-tej wagi gaussiana przez prawdopodobieństwo powiązane z rozkładem i-tego gaussiana.
   - W przypadku oznaczonych próbek może być reprezentowany przez wektor $\bar{p} = (0, 0, \ldots, 1, \ldots, 0, 0)$, gdzie 1 jest i-tym elementem. W ten sposób zmuszamy nasz model do "zaufania"oznaczonym próbkom, aby znaleźć najlepsze wartości parametrów, które maksymalizują funkcję wiarygodności całego zbioru danych.

   Widać, że mamy tutaj do czynienia z transdukcją (używamy zarówno próbek oznaczonych i nieoznaczonych w procesie uczenia się).

3. Krok M: aktualizacja parametrów zgodnie ze zmiennymi "ukrytymi"oszacowanymi w kroku E według następujących reguł

$$\begin{cases} w_i &= \frac{\sum_j \varphi_i(x_j)}{N} \\ \bar{\mu}_i &= \frac{\sum_j \varphi_i(x_j)\bar{x}_j}{\sum_j \varphi_i(x_j)} \\ \Sigma_i &= \frac{\sum_j \varphi_i(x_j)(\bar{x}_j - \bar{\mu}_i)(\bar{x}_j - \bar{\mu}_i)^T}{\sum_j \varphi_i(x_j)} \end{cases}$$

4. Powtarzaj Krok E oraz M aż do momentu, kiedy parametry przestaną się zmieniać (różnica między poprzednimi, a zaktualizowanymi jest poniżej pewnej ustalonej granicy).

5.2 Generative Gaussian Mixture - przykład

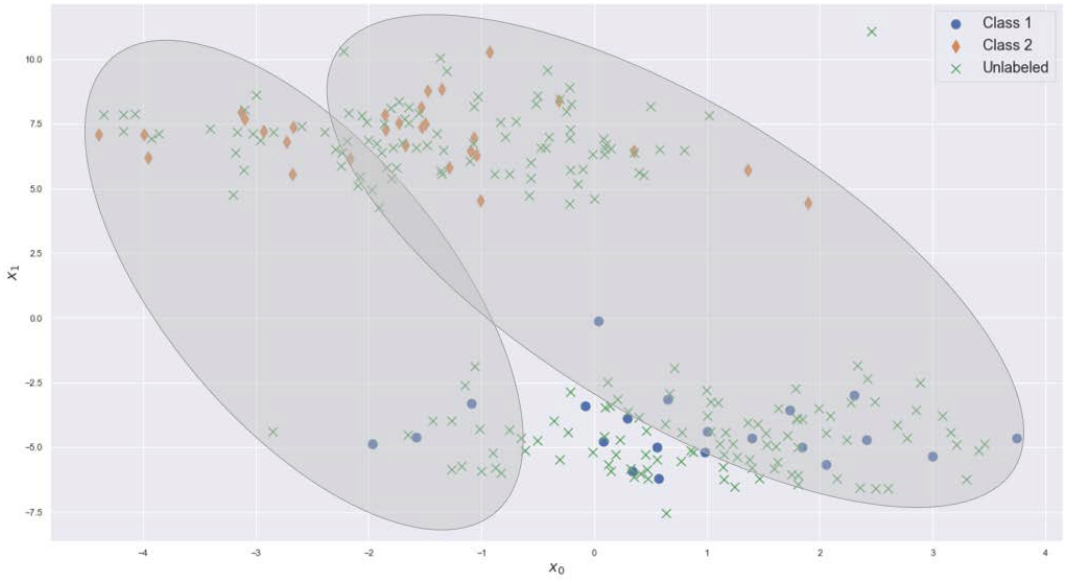
Przykład modelu Generative Gaussian Mixture jest zamieszczony w pliku .ipynb, gdzie zostaje dokładnie omówiony cały algorytm oraz kod.

6 Ważona logarytmiczna funkcja wiarygodności

W poprzednim przykładzie rozważaliśmy pojedynczą logarytmiczną funkcję wiarygodności zarówno dla oznaczonych, jak i nieoznaczonych próbek:

$$L(\bar{\theta}, \bar{w}) = \log \prod_{j=1}^{N} p(\bar{x}_j; \bar{\theta}, \bar{w}) = \sum_{j=1}^{N} \log \sum_{i=1}^{q} w_i p(\bar{x}_j | y_i; \bar{\theta})$$

Jest to równoznaczne ze stwierdzeniem, że ufamy nieoznakowanym punktom tak samo jak oznaczonym. Jednak w niektórych kontekstach założenie to może prowadzić do całkowicie błędnych oszacowań, jak pokazano na poniższym wykresie:



Rysunek 3: "Stronnicza"końcowa konfiguracja Generative Gaussian Mixture

W tym przypadku średnie i macierze kowariancji obu rozkładów gaussowskich zostały zniekształcone przez nieoznakowane punkty, a wynikowe oszacowanie gęstości jest wyraźnie nieprawidłowe.

Gdy takie zjawisko ma miejsce, najlepszą rzeczą do zrobienia jest rozważenie podwójnie ważonej funkcji wiarygodności. Jeśli pierwsze N próbek jest oznaczonych, a kolejne M jest nieoznaczonych, funkcję wiarygodności można wyrazić w następujący sposób:

$$L(\bar{\theta}, \bar{w}) = \sum_{j=1}^{N} \log \sum_{i} w_i p(\bar{x}_j | y_i; \bar{\theta}) + \lambda \sum_{j=N+1}^{N+M} \log \sum_{i} w_i p(\bar{x}_j | y_i; \bar{\theta})$$

We wzorze termin λ , jeśli jest mniejszy niż 1 i może zaniżać wagę nieoznakowanych próbek, przypisując większe znaczenie oznaczonemu zbiorowi danych. Modyfikacja algorytmu polega na dopilnowaniu, aby każda nieoznakowana waga była skalowana zgodnie z λ, zmniejszając szacowane prawdopodobieństwo.

Istnieje wiele potencjalnych zasad, aby określić wpływ nieoznakowanej próbki na oznakowaną. Możliwa strategia znalezienia optymalnego λ może być oparta na walidacji krzyżowej przeprowadzonej na oznaczonym zbiorze danych. Innym, bardziej złożonym podejściem jest rozważenie różnych rosnących wartości λ i wybranie tej, dla której logarytm funkcji wiarygodności jest maksymalny. W obu przypadkach celem jest znalezienie wartości, która pozwoli uniknąć dominacji nieoznakowanych próbek.

7 Self-Training

Self-Training jest bardzo intuicyjnym podejściem do klasyfikacji półnadzorowanej, opartym na szerokim zastosowaniu założenia gładkości oraz założenia klastra. Self-Training jest ogólnie dobrym wyborem, gdy oznaczony zbiór danych zawiera wystarczającą ilość informacji o podstawowym procesie generowania danych (to znaczy, że walidacja krzyżowa wykazuje stosunkowo wysoką dokładność) oraz gdy zakłada się, że nieoznakowana próbka jest odpowiedzialna tylko za 'dostrojenie' algorytmu. Gdy warunek ten nie jest spełniony, Self-Training nie może zostać wybrany, ponieważ w dużym stopniu opiera się na etykietowanych próbkach.

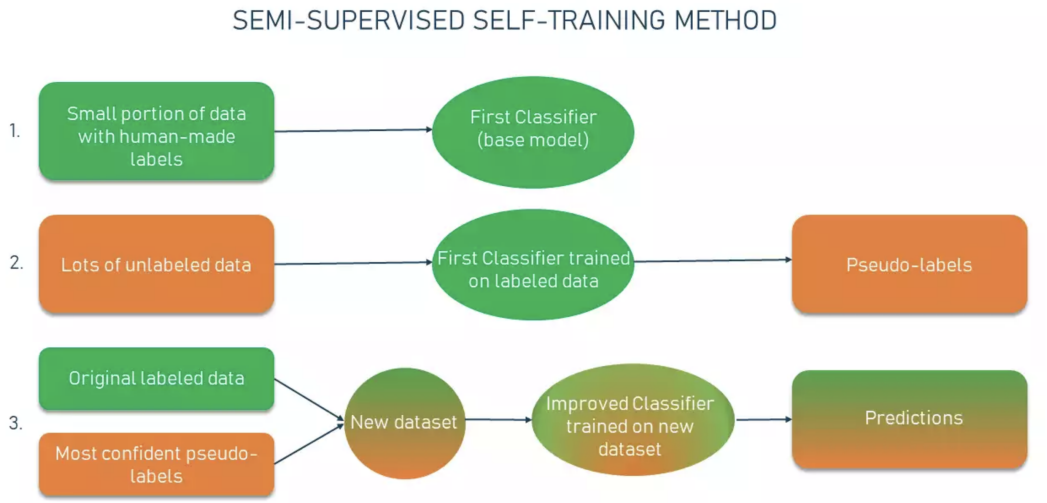
7.1 Self-Training - teoria

Załóżmy, że mamy zbiór danych oznaczonych próbek $\{X_L, Y_L\}$ i zakładamy, że został on pobrany jednostajnie z procesu generowania danych p. Ponadto istnieje inny zbiór nieoznakowanych punktów danych X_U, który zakłada się, że ma taki sam rozkład jak X_L. Załóżmy, że klasyfikator jest trenowany przy użyciu pierwszego oznakowanego zbioru danych (tj. początkowego zawierającego tylko wstępnie oznakowane punkty), a jego dokładność jest wystarczająco duża, aby uznać jego przewidywania za wiarygodne. W tym momencie algorytm Self-Training może spróbować uwzględnić nieoznakowany zbiór danych za pomocą prostej procedury iteracyjnej:

1. Wszystkie punkty $\bar{x}_u \in X_u$ są oceniane przez wstępny model, a każda predykcja jest reprezentowana przez wektor ufności $\bar{p}(\bar{x}_U) = (p_1, p_2, \ldots, p_m)$ (zakładając, że istnieje m klas).
2. Pierwszych k wartości z największą 'ufnością' (przynależności do dowolnej klasy) jest wybieranych, usuwanych z X_U i dodawanych do oznaczonego zbioru danych
3. Klasyfikator jest ponownie trenowany przy użyciu nowego zestawu treningowego.

Możemy zatem zauważyć, że mamy tutaj do czynienia z uczeniem indukcyjnym. Model uczy się początkowo tylko na oznakowanych danych i później "rozszerza się"na dane nieoznakowane.

Przedstawmy graficznie ten proces w celu lepszego zrozumienia:



Rysunek 4: Self-Training - algorytm

Proces ten należy powtarzać, aż wszystkie nieznakowane wartości zostaną oznakowane i nie będzie już potrzeby ponownego trenowania klasyfikatora. Metoda ta jest dość prosta i intuicyjna, ale jak można jej zaufać?

Pierwszym podstawowym założeniem jest to, że wszystkie wartości X są pobierane z tego samego rozkładu. Dlatego też, gdy pierwsza próbka treningowa nie jest zbyt mała, klasyfikator może zacząć uczyć się struktury procesu bazowego i rozwinąć zdolność uogólniania.

Kolejne założenia dotyczą gładkości i klastrów. W szczególności musimy założyć, że podobne próbki z dużym prawdopodobieństwem będą powiązane z podobnymi wynikami, wykluczając de facto możliwość nagłych skoków.

W związku z tym, jeśli początkowy klasyfikator jest już dobrze wytrenowany, będzie kilka predykcji, których 'ufność' jest wystarczająco duża, aby uzasadnić ich włączenie do nowego zestawu szkoleniowego. Liczba k wartości może być z góry określona lub oparta na podejściu 'adaptacyjnym'.
W pierwszym przypadku wybieramy zawsze k najlepszych próbek, dlatego k musi być wystarczająco małe, aby zagwarantować maksymalną dokładność, ale jednocześnie wystarczająco duże, aby nie spowalniać nadmiernie procedury uczenia.

Podejście 'adaptacyjne' natomiast może uwzględniać pewną minimalną granicę 'ufności' i wybierać tylko te próbki, które spełniają ten wymóg. Gdy stosowana jest taka alternatywa, ważne jest również rozważenie ograniczenia maksymalnej liczby wartości na iterację.

Algorytm Self-Training musi być uważnie monitorowany podczas całej fazy uczenia, w szczególności gdy zbiór danych z etykietami jest mały. Jeśli na przykład $\{X_L, Y_L\}$ jest losowany z ograniczonego obszaru p, tylko kilka bliskich nieoznakowanych punktów zostanie poprawnie rozpoznanych. Niestety bez żadnych dalszych wskazówek algorytm będzie kontynuował etykietowanie pozostałych wartości, zyskując coraz większą 'pewność' pomimo błędnych wyników. Problem ten jest konsekwencją założenia klastra podczas tworzenia nowych, rozszerzonych zbiorów treningowych.

Możliwym sposobem ograniczenia ryzyka jest upewnienie się o poprawności podstawowych założeń (to znaczy, że $\{X_L, Y_L\}$ równomiernie pokrywa cały proces generowania danych).Jeśli założenie nie jest przestrzegane, konieczne jest ponowne przeanalizowanie problemu lub znalezienie lepszych oznakowanych próbek. Teraz, na podstawie znanego nam już modelu Naiwnego Klasyfikatora Bayesowskiego pokażmy zastosowanie metody Self-Training.

7.2 Self-Training - przykład ze zbiorem danych Iris

Przykład zastosowania Self-Training jest zamieszczony w pliku .ipynb, gdzie zostaje dokładnie omówiony cały algorytm oraz kod.

8 Co-Training

Co-Training to kolejne stosunkowo proste, ale skuteczne podejście pół-nadzorowane. Jak w podejściu Self-Training polega ono na 'doszlifowaniu' już dobrego modelu opartego na danych oznakowanych. Ta metoda jest stosowana gdy zbiór danych jest wysokowymiarowy, a różne grupy cech są w stanie oddzielnie dobrze przewidywać przynależność do klas. Zatem jeśli każdy punkt danych zawiera cechy których nie można podzielić na oddzielne grupy, metoda ta jest nieskuteczna. Podział zbioru cech na podzbiory jest często dużym wyzwaniem i często wymaga ingerencji eksperta z danej dziedziny.

W oryginalnym artykule badawczym dotyczącym Co-Trainingu (*Combining Labeled and Unlabeled data with Co-Training, Avrim Blum, Tom Mitchell)* stwierdzono, że podejście to może być z powodzeniem stosowane na przykład w zadaniach klasyfikacji stron internetowych. Opis każdej strony internetowej można podzielić na dwa 'widoki': jeden ze słowami występującymi na tej stronie, a drugi ze słowami strony, która posiadała hyperlink prowadzący do niej.

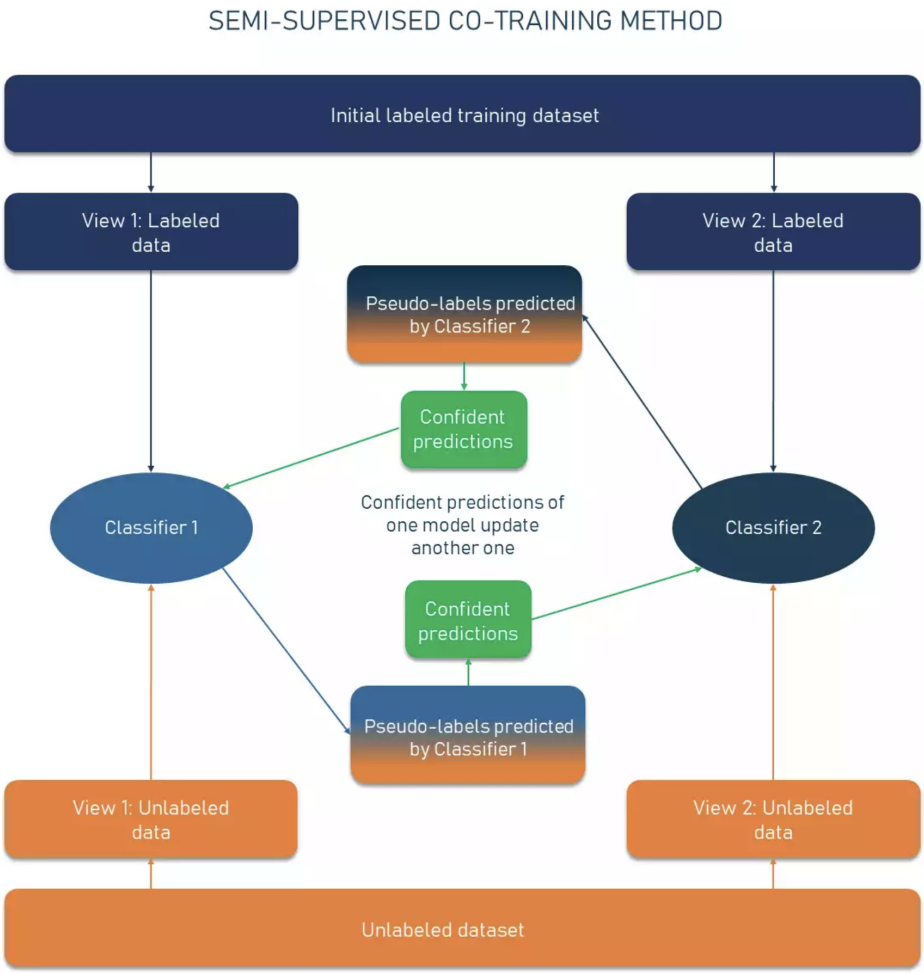
8.1 Co-Training - teoria

Załóżmy, że mamy oznakowany zbiór danych $\{X_L, Y_L\}$ z $\bar{x}_i \in \mathbb{R}^n$. Główną ideą podejścia Co-Training jest fakt, że w wielu przypadkach podzbiór cech $\bar{n}_0 = \{n_i, n_{i+1}, \ldots, n_k\}$ (n_i oznacza i-tą cechę) zwany "widokiem"(view) może być wykorzystywany przez "wyspecjalizowany"klasyfikator, w celu modelowania określonego zachowania, które przypisuje określoną klasę. Analogicznie, pozostałe cechy (w scenariuszu opartym na dwóch klasyfikatorach) są wykorzystywane przez inny klasyfikator, który powinien dojść do analogicznych wniosków, co pierwszy. Oznacza to, że oba klasyfikatory muszą przypisać próbkę do tej samej klasy (w zbiorze oznakowanym, czyli naszym wstępnym treningowym). W przypadku rozbieżności oznacza to, że zbiór treningowy nie zawierał wystarczającej ilości informacji do podjęcia prawidłowej decyzji. Jest to główne założenie podejścia Co-Training, nie zawsze jest ono możliwe do spełnienia.

Podejście Co-Training polega na przejściu przez następujące kroki:

1. X_L jest podzielone na (w naszym przypadku) dwa 'widoki' X_{L1} oraz X_{L2}
2. Pierwszy klasyfikator jest trenowany na zbiorze $\{X_{L1}, Y_L\}$
3. Drugi klasyfikator jest trenowany na zbiorze $\{X_{L2}, Y_L\}$
4. Biorąc zbiór nieoznakowany X_U, dzielimy go na dwa zbiory X_{U1} oraz X_{U2}. Oba klasyfikatory etykietują k punktów z największą ufnością (jak w Self-Training). W tym momencie możemy dokładniej zrozumieć sens przedsionka "Co-". Klasyfikatory kooperują ze sobą, mianowicie, jeżeli klasyfikator pierwszy jest 'pewny' danego wyniku, dołącza go do danych oznakowanych i tak samo w przypadku drugiego klasyfikatora. Oznacza to, że jeden klasyfikator może wyręczyć drugi z klasyfikacji 'mniej pewnego' dla niego punktu.
5. Po zklasyfikowaniu próbki zostają usunięte ze zbioru nieoznakowanego i dodane do zbioru treningowego (oznakowanego) i proces jest powtarzany od punktu 2. aż wszystkie próbki danych zostaną oznakowane.

Przedstawmy graficznie ten proces w celu lepszego zrozumienia:



Rysunek 5: Co-Training - algorytm

Zatem podobnie jak w przypadku Self-Training, Co-Training przechodzi wiele iteracji w celu oznaczenia wszystkich danych nieoznaczonych. Rozważmy teraz przykład zastosowania omawianego podejścia.

8.2 Co-Training - przykład ze zbiorem danych Wine

Przykład zastosowania Co-Training jest zamieszczony w pliku .ipynb, gdzie zostaje dokładnie omówiony cały algorytm oraz kod.

9 Podsumowanie

W tym referacie przedstawiliśmy uczenie częściowo nadzorowane, zaczynając od zdefiniowania pewnego rodzaju "środowiska", gdzie dane metody się sprawdzają. Poruszyliśmy temat różnych scenariuszy oraz ich wpływu na uczenie częściowo nadzorowane, przedstawiliśmy założenia, które muszą być spełnione aby odpowiednie metody działały poprawnie. Omówiliśmy znaczenie założenia gładkości gwarantujące pewnego rodzaju zdolność uogólniania. Następnie wprowadziliśmy założenie klastra, mówiące o pewnej strukturze danych, a jako ostatnie założenie omówiliśmy założenie rozmaitości i jego znaczenie przy danych wysokowymiarowych.

Kontynuacją referatu było wprowadzenie algorytmów z dziedziny nauczania częściowo nadzorowanego. Zaczeliśmy od modelu Generative Gaussian Mixture, który umożliwia grupowanie próbek oznakowanych i nieoznakowanych, wychodząc z założenia, że rozkłady w klasach są modelowane za pomocą wielowymiarowych rozkładów Gaussa. Wprowadziliśmy także metody Self-Training i Co-Training. Pierwsze z nich to algorytm wykorzystujący zarówno założenie klastra jak i gładkości do polepszenia jakości modelu nadzorowanego poprzez zastosowanie procedury iteracyjnej dla nieoznakowanego zbioru danych. Co-Training jest stosowany przy danych wysokowymiarowych i opiera się na symultanicznym wykorzystaniu dwóch różnych „widoków” tego samego zbioru danych przez dwa wyspecjalizowane klasyfikatory, aby zwiększyć 'pewność' przypisania nieoznakowanych próbek