Autoreferat

1 Imię i nazwisko: Wojciech Kotłowski

2 Posiadane dyplomy oraz stopnie naukowe

1. **Stopień doktora nauk technicznych** w dyscyplinie: **informatyka**, Politechnika Poznańska, Wydział Informatyki i Zarządzania, **3 marca 2009 r**.

   Tytuł pracy: *Podejście statystyczne do klasyfikacji porządkowej z ograniczeniami monotonicznymi*.
   Promotor: prof. dr hab. inż. Roman Słowiński.

2. **Tytuł zawodowy magistra fizyki**, Uniwersytet im. Adama Mickiewicza w Poznaniu, Wydział Fizyki, specjalność: fizyka teoretyczna (optyka kwantowa), **2006**.

3. **Tytuł zawodowy magistra informatyki**, Politechnika Poznańska, Wydział Informatyki i Zarządzania, specjalność: inteligentne systemy wspomagania decyzji, **2004**.

4. **Tytuł inżyniera informatyki**, Politechnika Poznańska, Wydział Elektryczny, **2002**.

3 Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

Od 10/2011	**Adiunkt**, Instytut Informatyki Politechniki Poznańskiej
06/2009–03/2012	**Staż podoktorski**, Centrum Wiskunde & Informatica (CWI) w Amsterdamie
11/2010–03/2011	**Staż naukowy**, Uniwersytet Kalifornijski w Santa Cruz
10/2008–09/2011	**Asystent**, Instytut Informatyki Politechniki Poznańskiej

4 Wskazanie osiągnięcia naukowego

4.1 Tytuł osiągnięcia naukowego

Algorytmy optymalne w sensie kryterium żalu w teorii przyrostowego uczenia się i statystycznej teorii uczenia się.

4.2 Lista publikacji

[A_1] Grünwald, P. i Kotłowski, W. (2010). Prequential plug-in codes that achieve optimal redundancy rates even if the model is wrong. *The IEEE International Symposium on Information Theory (ISIT 2010)*, strony 1383-1387. IEEE.

[A_2] Kotłowski, W., Grünwald, P. i de Rooij, S. (2010). Following the flattened leader. *Proceedings of the 23rd Annual Conference on Learning Theory (COLT 2010)*, strony 106-118. Omnipress.

[A_3] Kotłowski, W. i Grünwald, P. (2011). Maximum likelihood vs. sequential normalized maximum likelihood in on-line density estimation. *Proceedings of the 24th Annual Conference on Learning Theory (COLT 2011)*, wolumen 19 *Journal Machine Learning Research Workshop and Conference Proceedings*, strony 457–476. JMLR.

[A_4] Koolen, W., Kotłowski, W. i Warmuth, M. (2011). Learning eigenvectors for free. *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, strony 945–953. Curran Associates, Inc.

[A_5] Kotłowski, W. i Słowinski, R. (2013). On nonparametric ordinal classification with monotonicity constraints. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2576-2589.

[A_6] Bartlett, P., Grünwald, P., Harremoës, P., Hedayati, F. i Kotłowski, W. (2013). Horizon-independent optimal prediction with log-loss in exponential families. *Proceedings of the 26th Annual Conference on Learning Theory (COLT 2013)*, wolumen 30 *Journal of Machine Learning Research Workshop and Conference Proceedings*, strony 639–661. JMLR.

[A_7] Warmuth, M. K., Kotłowski, W. i Zhou, S. (2014). Kernelization of matrix updates, when and how? *Theoretical Computer Science*, 558:159–178.

[A_8] Van Erven, T., Kotłowski, W. i Warmuth, M. K. (2014). Follow the leader with dropout perturbations. *Proceedings of the 27th Annual Conference on Learning Theory (COLT 2014)*, wolumen 35 *Journal of Machine Learning Research Workshop and Conference Proceedings*, strony 949–974. JMLR.

[A_9] Nie, J., Kotłowski, W. i Warmuth, M. K. (2016). Online PCA with optimal regret. *Journal of Machine Learning Research*, 17(173):1–49.

[A_{10}] Kotłowski, W., Koolen, W. M. i Malek, A. (2016). Online isotonic regression. *Proceedings of the 29th Annual Conference on Learning Theory (COLT 2016)*, wolumen 49 *Journal of Machine Learning Research Workshop and Conference Proceedings*, strony 1165–1189. JMLR.

[A_{11}] Kotłowski, W. (2016). On minimaxity of follow the leader strategy in the stochastic setting. *Proceeding of the 27th International Conference on Algorithmic Learning Theory (ALT 2016)*, wolumen 9925 *Lecture Notes in Artificial Intelligence*, strony 261–275. Springer-Verlag.

[A_{12}] Kotłowski, W. i Dembczyński, K. (2017). Surrogate regret bounds for generalized classification performance metrics. *Machine Learning Journal*, 106:549–572.

[A_{13}] Kotłowski, W. (2017). Scale-invariant unconstrained online learning. *Proceeding of the 28th International Conference on Algorithmic Learning Theory (ALT 2017)*, wolumen 76 *Proceedings of Machine Learning Research*, strony 412–433. PMLR.

4.3 Omówienie osiągnięcia naukowego

Uczenie maszynowe, dyscyplina sztucznej inteligencji zajmująca się projektowaniem systemów uczących się wykonywać zadania na podstawie danych, osiągnęła w ostatnich latach znaczące sukcesy w wielu dziedzinach nauki, technologii i przemysłu. Kluczem do zrozumienia ostatnich osiągnięć w tej dziedzinie jest matematyczna teoria dotycząca uczenia się z danych, nazywana *teorią uczenia się*. Teoria ta dostarcza odpowiedniego języka i narzędzi, które pozwalają nam analizować algorytmy uczące się, lepiej zrozumieć ich działanie i ograniczenia, a także projektować algorytmy gwarantujące ich optymalną wydajność.

Teoria uczenia się dotyczy *problemów predykcyjnych*, które mogą być opisane jako gra pomiędzy algorytmem uczącym się a środowiskiem [Cesa-Bianchi i Lugosi, 2006]. W danej chwili czasowej $t = 1, 2, \ldots$, algorytm uczący się (zwany również *strategią predykcyjną*) otrzymuje pewną informację wejściową $x_t \in \mathcal{X}$ i dokonuje predykcji (akcji) $\widehat{p}_t \in \mathcal{A}$ dotyczącej nieznanego wyjścia $y_t \in \mathcal{Y}$, bazując na x_t i uprzednio zaobserwowanych danych $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$, przy czym przez $\mathcal{X}, \mathcal{Y}$ i $\mathcal{A}$ oznaczyliśmy tu, odpowiednio, przestrzeń wejść, przestrzeń wyjść oraz przestrzeń predykcji/akcji. Następnie środowisko ujawnia wartość na wyjściu y_t i algorytm karany jest *stratą* (*loss*) $\ell(y_t, \widehat{p}_t)$ wyznaczaną za pomocą *funkcji straty* $\ell \colon \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$, mierzącej jakość predykcji. Celem algorytmu jest działanie obarczone niewielką wartością straty. Jakość algorytmu mierzona jest zwykle względem pewnego punktu odniesienia, którym jest *referencyjna rodzina strategii predykcyjnych*[1] $\mathcal{F}$, przy czym różnica pomiędzy stratą otrzymaną przez algorytm a najmniejszą stratą otrzymaną przez jedną ze strategii z rodziny $\mathcal{F}$ nazywana jest, używając języka teorii gier, *żalem* (*regret*). Przykładowo, w problemie regresji liniowej $\mathcal{A} = \mathcal{Y} = \mathbb{R}$, $\mathcal{X} = \mathbb{R}^n$, dane wyjściowe x_t są wektorami cech $\boldsymbol{x}_t \in \mathbb{R}^n$, strata mierzona jest za pomocą błędu kwadratowego $\ell(y_t, \widehat{p}_t) = (y_t - \widehat{p}_t)^2$, a $\mathcal{F}$ jest rodziną wszystkich funkcji liniowych postaci $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ dla pewnego wektora wag (współczynników) $\boldsymbol{w} \in \mathbb{R}^n$. Predykcja algorytmu $\widehat{p}_t = f_t(\boldsymbol{x}_t)$ również otrzymywana jest przez jedną z funkcją liniowych $f_t \in \mathcal{F}$, a celem jest uzyskanie wartości straty niewiele większej od wartości straty najlepszej funkcji liniowej.

[1] Referencyjna rodzina strategii predykcyjnych jest w statystyce często nazywana *modelem* lub *klasą modeli*.

Istnieją dwa warianty teorii uczenia się z danych. *Statystyczna teoria uczenia się* [Devroye i inni, 1996; Vapnik, 1998] zakłada, że dane (obserwacje) $(x_1, y_1), (x_2, y_2), \ldots$ są generowane w sposób niezależny i pochodzą z tego samego (nieznanego) rozkładu prawdopodobieństwa $P(x, y)$. Algorytm uczący się oceniany jest wtedy za pomocą *oczekiwanej wartości straty* $\mathbb{E}_{(x_t,y_t)\sim P}\left[\ell(y_t, \widehat{p}_t(x_t))\right]$, a żal[2] algorytmu jest różnicą między jego oczekiwaną wartością straty a oczekiwaną wartością straty najlepszej strategii predykcyjnej z rodziny $\mathcal{F}$. Efektywny algorytm uczący się powinien uzyskiwać wartość żalu bliską zeru, gdy liczba obserwacji t jest odpowiednio duża, niezależnie od charakteru rozkładu generującego dane.

Drugi z wariantów, *teoria przyrostowego uczenia się* [Cesa-Bianchi i Lugosi, 2006; Shalev-Shwartz, 2012], nie nakłada żadnych stochastycznych założeń na środowisko generujące dane, używając do analizy algorytmów języka teorii gier iterowanych. Uczenie przyrostowe zajmuje się *sekwencyjnymi problemami predykcyjnymi*, dotyczącymi algorytmów iteracyjnie podejmujących akcje (predykcje) na podstawie napływających sekwencyjnie danych. Całkowitą jakość predykcji algorytmu mierzy się za pomocą sumarycznej wartości straty na pełnej sekwencji danych i porównuje się ją do sumarycznej wartości straty najlepszej (z perspektywy czasu) strategii predykcyjnej z rodziny $\mathcal{F}$. Różnicę między tymi wartościami nazywa się żalem. Celem algorytmu jest minimalizacja żalu na każdej sekwencji danych. Ponieważ definicja żalu oparta jest tu na skumulowanych wartościach straty, efektywny algorytm uczący się powinien gwarantować żal rosnący *subliniowo* z liczbą obserwacji t dla każdej sekwencji danych (co oznacza, że żal *na iterację* zbiega do zera). Co ciekawe, teoria przyrostowego uczenia się nie nakłada *żadnych* założeń na napływające dane: mogą one mieć naturę stochastyczną, być wynikiem deterministycznego procesu, lub wręcz być preparowanymi przez przeciwnika.

W obu wariantach teorii uczenia się żal określa więc *suboptymalność*: w jakim stopniu optymalna akcja, która mogłaby być podjęta (gdyby wszystkie dane lub ich rozkład byłyby znane zawczasu), i akcja, która faktycznie została podjęta, różnią się w sensie wartości straty? Jeśli problem predykcyjny nie jest trywialny to żadna strategia predykcyjna nie zagwarantuje zerowej wartości żalu w najgorszym przypadku. Nasuwa się więc pytanie: jaką najmniejszą wartość żalu można zagwarantować i jakie algorytmu taką gwarancję posiadają?

Celem niniejszego cyklu publikacji jest analiza teoretyczna i konstrukcja algorytmów uczących się z *optymalnymi* gwarancjami żalu. Większość wyników zaprezentowana jest w ramach teorii przyrostowego uczenia się, bez nakładania stochastycznych założeń na obserwowane dane. W niektórych przypadkach odwołujemy się do statystycznej teorii uczenia się aby otrzymać ograniczenia na żal.

Omówienie celów naukowych i wyników prac wchodzących w skład prezentowanego cyklu zostanie podzielone na sześć następujących części:

1. Przyrostowe uczenie się w modelu *predykcji z pomocą ekspertów* ([A_8], [A_{11}]).
2. Przyrostowe uczenie się z logarytmiczną funkcją straty ([A_1], [A_2], [A_3], [A_6]).
3. Przyrostowe uczenie się macierzy ([A_4], [A_7], [A_9]).
4. Przyrostowe uczenie się niezmiennicze względem skali ([A_{13}]).
5. Uczenie się funkcji monotonicznych ([A_5], [A_{10}]).
6. Redukcje żalu w klasyfikacji binarnej ([A_{12}]).

Pokażemy że różnice między tym częściami sprowadzają się tylko do konkretnego wyboru elementów definiujących problem predykcyjny: przestrzeni akcji, wejść i wyjść, referencyjnej rodziny strategii predykcyjnych oraz funkcji straty.

4.3.1 Predykcja z pomocą ekspertów

Zaczniemy od sekwencyjnego procesu predykcyjnego nazywanego *predykcją z pomocą ekspertów* (*prediction with expert advice*) [Cesa-Bianchi i inni, 1997; Cesa-Bianchi i Lugosi, 2006]). Problem ten, pomimo swojej prostoty, stanowi podstawę teorii przyrostowego uczenia się i prowadzi do licznych zastosowań praktycznych, takich jak zadanie wyboru najlepszego klasyfikatora [Zamani i inni, 2016], przyrostowy problem wyboru najkrótszej ścieżki [Kalai i Vempala, 2005], analiza szeregów czasowych [Dashevskiy i Luo, 2011], strategie w grach [V'yugin, 2015] i wiele innych.

[2] W literaturze statystycznej, żal w tym wypadku często nazywany jest *nadwyżką ryzyka*.

Celem algorytmu jest predykcja sekwencji na wyjściu $y_{1:T} = y_1, \ldots, y_T \in \mathcal{Y}$. W każdej iteracji $t = 1, 2, \ldots, T$, algorytm ma dostęp do K *ekspertów*, proponujących swoje własne predykcje $x_{t,1}, \ldots, x_{t,K}$ nieznanej wartości wyjściowej y_t. Algorytm wybiera, zwykle niedeterministycznie, jednego z ekspertów, oznaczonego przez k_t. Akcję algorytmu można reprezentować więc w postaci *wektora prawdopodobieństw* na zbiorze $\{1, \ldots, K\}$, $\widehat{\boldsymbol{p}}_t = (\widehat{p}_{t,1}, \ldots, \widehat{p}_{t,k})$ (przy czym $\widehat{p}_{t,k} \geqslant 0$ dla wszystkich k i $\sum_{k=1}^{K} \widehat{p}_{t,k} = 1$), zgodnie z którym określany jest wybór eksperta: $k_t \sim \widehat{\boldsymbol{p}}_t$; przestrzeń akcji $\mathcal{A}$ jest więc zbiorem rozkładów prawdopodobieństw na K elementach. Następnie środowisko ujawnia y_t i każdy ekspert karany jest stratą $\ell_{t,k} = \ell(y_t, x_{t,k})$, natomiast algorytm karany jest stratą wybranego eksperta: ℓ_{t,k_t}. W ramach analizy problemy będziemy zainteresowani *oczekiwaną wartością straty* algorytmu (ze względu na wewnętrzną randomizację algorytmu), wyrażoną jako $\widehat{\ell}_t = \sum_{k=1}^{K} \widehat{p}_{t,k} \ell_{t,k} = \widehat{\boldsymbol{p}}_t^{\top} \boldsymbol{\ell}_t$, gdzie $\boldsymbol{\ell}_t = (\ell_{t,1}, \ldots, \ell_{t,K})$ jest wektorem straty ekspertów w chwili t. Założymy, że maksymalna wielkość straty jest ograniczona, dla prostoty przyjmując $\ell_{t,k} \in [0, 1]$ dla dowolnych t, k. Ponieważ konkretna forma predykcji ekspertów i wartości na wyjściu nie są istotne dla dalszej analizy, będziemy rozważali wyłącznie wektory strat $\boldsymbol{\ell}_t$ i akcje algorytmu $\widehat{\boldsymbol{p}}_t$. Niech $\widehat{L}_T = \sum_{t=1}^{T} \widehat{\ell}_t$ oznacza sumaryczną stratę algorytmu, a $L_{T,k} = \sum_{t=1}^{T} \ell_{t,k}$, $k = 1, \ldots, K$, oznacza sumaryczne straty ekspertów. Jakość algorytmu będzie mierzona za pomocą żalu:

$$R(\widehat{\boldsymbol{p}}; \boldsymbol{\ell}_{1:T}) = \widehat{L}_T - \min_{k=1,\ldots,K} L_{T,k} = \widehat{L}_T - L_T^*,$$

zdefiniowanego jako różnica pomiędzy sumaryczną stratą algorytmu i stratą najlepszego z ekspertów *z perspektywy czasu* (tzn. po ujawnieniu pełnej sekwencji danych) $L_T^* = \min_{k=1,\ldots,K} L_{T,k}$. Tym samym, jako referencyjną rodzinę strategii predykcyjnych $\mathcal{F}$ wybraliśmy zbiór K ekspertów. Celem algorytmu jest uzyskanie niewielkiej wartości żalu (*subliniowego* z T) dla dowolnej sekwencji wektorów strat.

Najprostszą metodą predykcji jest algorytm *Follow the Leader* (FL, „podążaj za najlepszym"), który w danej chwili t wybiera eksperta z aktualnie najmniejszą sumaryczną stratą. Niestety, żal metody FL może rosnąć liniowo z L_T^* (a tym samym z T), co oznacza porażkę procesu uczenia się. Problemu tego można jednak uniknąć stosując metodę *Weighted Majority* lub jej stratyfikowaną wersję *Hedge* [Littlestone i Warmuth, 1994; Freund i Schapire, 1997], w których, w chwili t, k-ty ekspert wybierany jest z prawdopodobieństwem proporcjonalnym do wykładniczej wagi $\exp(-\eta L_{t-1,k})$ dla pewnej dodatniej *szybkości uczenia* η. Przy odpowiednim dostrojeniu wartości η, algorytm *Hedge* gwarantuje subliniowy żal $O(\sqrt{L_T^* \log K} + \log K)$, co okazuje się wartością optymalną [Cesa-Bianchi i inni, 1997].

FL z perturbacją typu *dropout*. [A_8] Istnieje alternatywna do algorytmu *Hedge* metoda osiągająca subliniowy żal: dodaj do sumarycznej straty każdego z ekspertów losową perturbację i użyj strategii FL na tak zaburzonych stratach: $k_t = \arg\min_k \{L_{t-1,k} + \xi_{t-1,k}\}$, gdzie losowe perturbacje $\xi_{t-1,k}$ są niezależne dla każdego z ekspertów. Ten przepis definiuje ogólną klasę metod nazywanych *Follow the Perturbed Leader* (FPL) [Hannan, 1957; Kalai i Vempala, 2005]. Losując perturbacje z rozkładu wykładniczego z odpowiednio dostrojonym parametrem (jako funkcją L_T^* lub T), uzyskujemy algorytm również gwarantujący optymalny żal $O(\sqrt{L_T^* \log K} + \log K)$.

W pracy zaproponowaliśmy algorytm z klasy FPL, w którym, zamiast dodawać perturbację do sumarycznej straty, losowo zaburzamy wektor strat w każdej iteracji za pomocą procedury nazwanej przez nas *BDP* (*binarized dropout perturbation*): dla dowolnego $\alpha \in (0, 1)$, zaburzenie straty $\ell_{t,k}$ definiujemy jako:

$$\widetilde{\ell}_{t,k} = \begin{cases} 1 & \text{z prawdopodobieństwem } (1-\alpha)\ell_{t,k}, \\ 0 & \text{w przeciwnym przypadku.} \end{cases}$$

Niech $\widetilde{L}_{t,k} = \sum_{i=1}^{t} \widetilde{\ell}_{i,k}$ oznacza sumaryczną zaburzoną w ten sposób stratę k-tego eksperta na pierwszych t obserwacjach. W chwili t, algorytm BDP wybiera: $k_t = \arg\min_k \widetilde{L}_{t-1,k}$, losowo rozwiązując potencjalne remisy. Algorytm ten jest bardzo prosty koncepcyjnie i wyjątkowo łatwy do implementacji. Co ciekawe, używana przez nas perturbacja okazuje się podobna do tej stosowanej w technice *dropout* uczenia sieci neuronowych [Hinton i inni, 2012]. Co zaskakujące, ten prosty algorytm uzyskuje tą samą, *optymalną* gwarancję na żal dla dowolnej wartości parametru $\alpha \in (0, 1)$ (bez konieczności strojenia), *równocześnie* dla danych najgorszego przypadku i danych losowanych niezależnie z rozkładu prawdopodobieństwa:

Twierdzenie 1 ([A_8], twierdzenia 3.1 i 4.1) *Dla dowolnego $\alpha \in (0,1)$, algorytm BDP zapewnia:*

$$R(\widehat{\boldsymbol{p}};\, \boldsymbol{\ell}_{1:T}) = O(\sqrt{L_T^* \log K} + \log K),$$

co jest optymalne z dokładnością do stałej. Gdy wektory straty losowane są niezależnie z tego samego rozkładu i oczekiwana strata najlepszego eksperta jest ściśle mniejsza od oczekiwanych strat pozostałych ekspertów, żal algorytmu BDP redukuje się do $O(\log K)$, również optymalnego z dokładnością do stałej.

Większość algorytmów uczących się posiada parametry, które muszą być uważnie dostrojone, aby osiągnąć dobre gwarancje na żal. Aby osiągnąć te gwarancje równocześnie dla danych najgorszego przypadku i danych stochastycznych, trzeba użyć jeszcze bardziej złożonych metod strojenia, a otrzymane w ten sposób metody predykcji są nadmiernie skomplikowane. Co zaskakujące, nasza metoda osiąga optymalne gwarancje w obu przypadkach bez jakiegokolwiek strojenia parametrów.

Algorytm FL dla danych stochastycznych. [A_{11}] Problem predykcji z pomocą ekspertów nie nakłada żadnych założeń na dane, a celem jest minimalizacja żalu dla najtrudniejszej sekwencji. Strategia osiągająca minimum żalu w najgorszym przypadku nazywana jest strategią *minimaksową*. Jest ona odporna nawet na najtrudniejsze dane, ale również nadmiernie konserwatywna, gdyż nie wykorzystuje sytuacji w których napływające dane są znacznie łatwiejsze do przewidywania. W naszej pracy staraliśmy się znaleźć strategię minimaksową w bardziej ograniczonym przypadku, gdy dane mają naturę *stochastyczną*. W szczególności, założyliśmy, że straty każdego z ekspertów generowane są niezależnie z (nieznanego) rozkładów P_k, $k = 1, \ldots, K$. Ponieważ (oczekiwany) żal nie jest jedyną możliwą miarą jakości predykcji w przypadku stochastycznym, rozważyliśmy również dwie inne miary: oczekiwaną redundancję (*expected redundancy*) oraz nadwyżkę ryzyka (*excess risk*).

W celu znalezienia strategii minimaksowej, zdefiniowaliśmy pojęcie *niezmienniczości permutacyjnej* algorytmów, które mówi, że algorytm nie bierze pod uwagę indeksów (numerów) ekspertów, a jedynie wartości ich strat. Jest to bardzo naturalny warunek, spełniany zasadniczo przez wszystkie sensowne strategie predykcyjne (udowodniliśmy, że jeśli strategia nie spełnia tego warunku, to można utworzyć jej niezmienniczą permutacyjnie wersję, która jest ściśle lepsza w najgorszym przypadku). Głównym, zaskakującym wynikiem pracy jest pokazanie, że prosty algorytm FL jest algorytmem minimaksowym (w bardzo silnym sensie) dla wszystkich trzech rozważanych miar jakości przy binarnych wektorach strat:

Twierdzenie 2 ([A_{11}], wniosek 1) *Dla dowolnych rozkładów $P_1, \ldots, P_K$ na binarnych stratach $\ell_{t,k} \in \{0,1\}$, strategia FL ma najmniejszy oczekiwany żal, redundancję i nadwyżkę ryzyka spośród wszystkich strategii niezmienniczych permutacyjnie. Implikuje to minimaksowość strategii FL ze względu na powyższe miary.*

Pokazaliśmy również, że dla ciągłych wartości strat w przedziale $[0,1]$, strategia FL nie jest minimaksowa; własność tę posiada natomiast jej modyfikacja, *zbinaryzowana strategia FL* ([A_{11}], twierdzenie 3).

4.3.2 Przyrostowe uczenie się z logarytmiczną funkcją straty

Motywacją do badania logarytmicznej funkcji straty jest fundamentalna równoważność problem predykcji z tą funkcją straty i problemu kompresji danych [Rissanen, 1984; Grünwald, 2007]: strategia predykcyjna z niewielkim żalem może zostać użyta jako metoda kompresji danych z niewielką nadwyżką długości kodu, i odwrotnie. Zainteresowanie tym problemem sięga prac Kołmogorowa [1965] i Sołomonowa [1964], i ma bezpośrednie zastosowanie, poza wspomnianą kompresją danych, w estymacji gęstości rozkładów [Cesa-Bianchi i Lugosi, 2006], grach hazardowych czy analizie finansowej [Cover i Thomas, 1991].

Niech $y_{1:T} = y_1, y_2, \ldots, y_T \in \mathcal{Y}$ będzie sekwencją danych. W chwili t, po zaobserwowaniu $y_{1:t-1} = y_1, y_2, \ldots, y_{t-1}$, algorytm uczący się wybiera rozkład prawdopodobieństwa na $\mathcal{Y}$, oznaczany $\widehat{p}_t$ (przestrzenią akcji $\mathcal{A}$ jest tu więc zbiór wszystkich rozkładów prawdopodobieństwa na $\mathcal{Y}$; zauważmy też, że brak tu informacji wejściowej x). Następnie ujawniane jest y_t, a algorytm otrzymuje *logarytmiczną stratę* $\ell(y_t, \widehat{p}_t) = -\log \widehat{p}_t(y_t)$. Jakość algorytmu na całej sekwencji mierzona jest względem referencyjnej rodziny $\mathcal{F}$, która jest rodziną rozkładów na $\mathcal{Y}$, za pomocą żalu:

$$R(\widehat{p};\, y_{1:T}) = \sum_{t=1}^{T} \ell(y_t, \widehat{p}_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell(y_t, f).$$

Ponieważ $\widehat{p}_t$ jest rozkładem prawdopodobieństwa na $\mathcal{Y}$ zależnym od $y_{1:t-1}$, może być interpretowany jako *warunkowy rozkład prawdopodobieństwa* y_t pod warunkiem $y_{1:t-1}$. Rozkład *łączny* na całej sekwencji $y_{1:T}$ indukowany z rozkładów warunkowych ma postać $\widehat{p}(y_{1:T}) = \prod_{t=1}^{T} \widehat{p}_t(y_t)$. Zachodni również własność odwrotna: każdy łączny rozkład $\widehat{p}$ na całej sekwencji $y_{1:T}$ określa strategię predykcyjną indukowaną przez jego rozkłady warunkowe [Cesa-Bianchi i Lugosi, 2006].

Będziemy rozważali rodziny strategii $\mathcal{F}$, które są *rodzinami wykładniczymi rozkładów*. Niech $\mathcal{Y}$ będzie zbiorem przeliczalnym lub podzbiorem przestrzeni euklidesowej. Definiujemy n-parametrową rodzinę wykładniczą $\mathcal{F} = \{f_\theta \colon \theta \in \Theta \subseteq \mathbb{R}^n\}$ [Barndorff-Nielsen, 1978] jako rodzinę rozkładów prawdopodobieństwa na $\mathcal{Y}$ o gęstościach postaci $f_\theta(y) = e^{\theta^\top \phi(y) - \psi(\theta)} h(y)$, gdzie funkcja $\phi(y)$ nazywana jest *statystyką dostateczną*, h – *nośnikiem*, a $\psi(\theta)$ – *logarytmiczną sumą statystyczną*. Zbiór $\Theta = \{\theta \in \mathbb{R}^n : \psi(\theta) < \infty\}$ nazywany jest *naturalną przestrzenią parametrów*. Bez straty ogólności, zakładamy $\phi(y) \equiv y$, tzn. rodzina wykładnicza jest w *postaci naturalnej*. Gradient funkcji $\psi(\theta)$, oznaczony $\mu(\theta)$, jest wektorem wartości oczekiwanych y, $\mu(\theta) = \mathbb{E}_\theta[y]$, gdzie $\mathbb{E}_\theta[\cdot]$ oznacza wartość oczekiwaną względem $f_\theta \in \mathcal{F}$. Funkcja $\mu(\theta)$ jest odwracalna i dlatego pozwala na reparametryzację rodziny wykładniczej za pomocą wektora wartości oczekiwanych μ (*parametryzacja wartością oczekiwaną*). Przykładami rodzin wykładniczych są rozkłady Poissona, geometryczny, wielomianowy, wykładniczy, gamma, beta, Pareto, wielowymiarowe rozkłady normalne i wiele innych. Będziemy często rozważać rodziny wykładnicze $\mathcal{F} = \{f_\theta \colon \theta \in \Theta_0\}$ ograniczone do podzbioru przestrzeni parametrów $\Theta_0 \subseteq \Theta$. Powodem jest stosowalność niektórych naszych wyników tylko do rodzin ograniczonych do wypukłych i zwartych podzbiorów przestrzeni Θ_0, ponieważ bez tego ograniczenia każda strategia predykcyjna może mieć nieskończony żal już w pierwszej iteracji.

Jeśli horyzont czasowy T (całkowita długość sekwencji) jest znana, można bezpośrednio wyznaczyć strategię minimaksową (minimalizującą żal w najgorszym przypadku). Strategia ta znana jest pod nazwą *Normalized Maximum Likelihood (NML)* [Shtarkov, 1987; Rissanen, 1996] i zdefiniowana jako:

$$\widehat{p}_{\mathrm{nml}}(y_{1:T}) = \frac{\sup_{f \in \mathcal{F}} f(y_{1:T})}{\int_{\mathcal{Y}^T} \sup_{f \in \mathcal{F}} f(z_{1:T}) dz_1 \dots dz_T} \tag{1}$$

Żadna inna strategia predykcyjna nie będzie lepsza od strategii NML w najgorszym przypadku. Wiadomo [Rissanen, 1996; Grünwald, 2007], że gdy $\mathcal{F} = \{f_\theta \colon \theta \in \Theta_0\}$ jest n-parametrową rodziną wykładniczą ograniczoną do wypukłego i zwartego Θ_0, żal strategii NML jest dany w postaci $\frac{n}{2} \log T + O(1)$ dla *każdej sekwencji danych*, co wyznacza również minimaksową wartość żalu. Niestety, NML służy raczej jako optymalny punkt odniesienia niż praktyczna strategia predykcji, ponieważ wymaga znajomości horyzontu czasowego T (co jest bardzo nienaturalnym założeniem w większości zastosowań) i, co gorsza, obliczenia rozkładu łącznego i wyznaczenia rozkładów warunkowych, co w ogólności może mieć złożoność wykładniczą z rozmiarem $\mathcal{Y}$. Poniżej przedstawimy jednak strategie predykcyjne z gwarancjami żalu bardzo bliskimi wartości minimaksowej, a jednocześnie znacznie prostsze obliczeniowo.

Suboptymalność strategii *plug-in*. [A_1] Dla rodziny rozkładów $\mathcal{F} = \{f_\theta \colon \theta \in \Theta_0\}$, definiujemy strategię typu *plug-in* jako $\widehat{p}_t(\cdot) = f_{\widehat{\theta}_t}(\cdot)$, dla pewnego estymatora $\widehat{\theta}_t = \widehat{\theta}_t(y_{1:t-1})$. Innymi słowy, strategia *plug-in* zawsze przewiduje zgodnie z jednym z rozkładów z $\mathcal{F}$, wybranym poprzez pewien estymator parametru $\widehat{\theta}_t$ będący funkcją poprzednich obserwacji $y_{1:t-1}$. Najbardziej popularną strategią tego typu jest strategia *największej wiarygodności* (*Maximum Likelihood*, ML), przewidująca zgodnie z rozkładem, który przydziela największe prawdopodobieństwo zaobserwowanym wcześniej danym, tzn. z rozkładem określonym przez *estymator największej wiarygodności* $\widehat{\theta}_t = \arg\max_{\theta \in \Theta_0} f_\theta(y_{1:t-1})$. Zaletą strategii ML jest jest prostota: wiadomo [Grünwald, 2007], że estymator największej wiarygodności w parametryzacji wartości oczekiwanej jest średnią arytmetyczną statystyk dostatecznych. Niestety, pokazano [Grünwald i de Rooij, 2005], że strategia ML jest suboptymalna, nawet w przypadku gdy dane generowane są stochastycznie. Postawiono więc hipotezę badawczą [Grünwald, 2007], czy podobna suboptymalność dotyczy *wszystkich* strategii typu *plug-in*. Nasze wyniki pokazały, że ta hipoteza jest prawdziwa w najbardziej ogólnym sensie i *żadna strategia typu plug-in nie może osiągnąć optymalnej gwarancji na żal* $\frac{n}{2} \log T + O(1)$, nawet dla 1-parametrowej rodziny wykładniczej i danych generowanych stochastycznie:

Twierdzenie 3 ([A_1], wniosek 1) *Niech $\mathcal{F} = \{f_\theta \colon \theta \in \Theta\}$ będzie 1-parametrową rodziną wykładniczą, a $\widehat{p}$ dowolną strategią* plug-in. *Dla prawie wszystkich $\theta \in \Theta$ (wszystkich z wyjątkiem zbioru miary (Lebesgue'a)*

zero), istnieje rozkład prawdopodobieństwa P z $\mathbb{E}_P[y] = \mathbb{E}_\theta[y] = \mu(\theta)$, taki, że dla $y_1, \ldots, y_T \sim P$ zachodzi:

$$\mathbb{E}_P[R(\widehat{p}; y_{1:T})] \geqslant c \log T,$$

gdzie $c > 1/2$ może być dowolnie duże.

Ponieważ optymalna wartość żalu dla $n = 1$ wynosi $\frac{1}{2}\log T + O(1)$, wszystkie strategie *plug-in* są suboptymalne dla prawie każdego wyboru parametru $\mu(\theta)$; w istocie, żadna strategia *plug-in* nie może być znacząco lepsza od strategii ML.

Ograniczenie żalu dla strategii ML. [A_3] Pomimo popularności algorytmu ML, gwarancje na jego żal w najgorszym przypadku otrzymano tylko dla kilku szczególnych rozkładów: normalnego, dwupunktowego oraz gamma [Freund, 1996; Azoury i Warmuth, 2001]. Udowodniliśmy ogólne ograniczenie na żal dla strategii ML, które zachodzi dla *dowolnej* rodziny wykładniczej:

Twierdzenie 4 ([A_3], twierdzenie 1) *Niech $\widehat{p}$ będzie strategią ML, a $\mathcal{F} = \{f_\theta : \theta \in \Theta_0\}$ rodziną wykładniczą ograniczoną do wypukłego i zwartego podzbioru Θ_0. Dla każdej ograniczonej sekwencji $y_1, \ldots, y_T$ ($\|y_t\| \leqslant B$ dla każdego t) zachodzi:*

$$R(\widehat{p}; y_{1:T}) \leqslant C \log T + O(1),$$

gdzie C jest zależne od B i Θ_0.

O ile strategia ML nadal gwarantuje żal logarytmiczny z T, to stała przed $\log T$ zależy od długości wektorów danych i może być znacznie większa niż optymalna stała $\frac{n}{2}$. Założenie o ograniczeniu normy danych do B i rodziny wykładniczej do zwartego Θ_0 są konieczne, ponieważ bez nich strategia ML może mieć żal nawet liniowy z T [Dasgupta i Hsu, 2007]. Pokazaliśmy też, że gwarancja w powyższym twierdzeniu jest w zasadzie nie do polepszenia dla jakiejkolwiek strategii *plug-in* ([A_3], twierdzenie 2).

Ograniczenie żalu dla strategii SNML. [A_3] Jak pokazano, strategia ML jest suboptymalna. Udowodniliśmy jednak, że problem ten może zostać rozwiązany poprzez dodanie aktualnie przewidywanej obserwacji (jeszcze nie ujawnionej przez środowisko) do wyznaczenia estymatora największej wiarygodności, a następnie normalizację tak otrzymanego rozkładu. Strategia ta nazywa się *Sequential Normalized Maximum Likelihood* (SNML) [Rissanen i Roos, 2007], i jest zdefiniowana jako:

$$\widehat{p}_t(y_t) = \frac{1}{Z} f_{\widehat{\theta}_{t+1}}(y_t),$$

gdzie $\widehat{\theta}_{t+1} = \arg\max_{\theta \in \Theta_0} f_\theta(y_{1:t})$ jest estymatorem największej wiarygodności na poprzednich obserwacjach, *włączając też przewidywaną obserwację* y_t, a Z normalizuje rozkład: $Z = \int_{\mathcal{Y}} f_{\widehat{\theta}_{t+1}}(y_t) dy_t$. Stała Z nie jest równa jedności, ponieważ $f_{\widehat{\theta}_{t+1}}(y_t)$ nie normalizuje się poprawnie ($\widehat{\theta}_{t+1}$ zależy od y_t). Tak zdefiniowany rozkład zwykle *nie będzie* należał do rodziny $\mathcal{F}$. Strategia SNML została odkryta przez Rissanena i Roosa [2007], jednak jej zachowanie dla ogólnych rodzin wykładniczych nie było wcześniej badane. Uzyskaliśmy następujący rezultat:

Twierdzenie 5 ([A_3], twierdzenie 4) *Niech $\widehat{p}$ będzie strategią SNML, $\mathcal{F} = \{f_\theta : \theta \in \Theta_0\}$ rodziną wykładniczą ograniczoną do wypukłego i zwartego Θ_0. Dla dowolnej sekwencji danych:*

$$R(\widehat{p}; y_{1:T}) \leqslant \frac{n}{2} \log T + O(1).$$

Tym samym, strategia SNML osiąga, z dokładnością do $O(1)$, minimaksowy żal dla *wszystkich rodzin wykładniczych*, *bez żadnych założeń o danych*. Rissanen i Roos [2007] zauważyli, że predykcje SNML są równoważne predykcjom minimaksowej strategii NML (równanie 1) przy założeniu, że aktualna iteracja jest ostatnią. Tym samym, strategia SNML może być uznana za przybliżenie strategii NML gdy horyzont czasu T nie jest znany, i w ten sposób uniknąć kosztownej obliczeniowo marginalizacji rozkładów względem wszystkich możliwych przyszłych podsekwencji. Określenie gwarancji na żal SNML było więc istotne

nie tylko z powodów praktycznych (jako, że strategia SNML jest efektywnym algorytmem predykcyjnym), ale również z powodów koncepcyjnych, prowadzi bowiem do następującego pytania: jak wiele tracimy gdy opieramy nasze decyzje w danym momencie czasu patrząc tylko jeden krok naprzód, zamiast patrzeć na całą dalszą przyszłość? Co ciekawe, pokazaliśmy, że działając krótkowzrocznie tracimy bardzo niewiele: zaledwie $O(1)$ w stosunku do minimaksowej wartości żalu.

Wymienialność stategii SNML. [A_6] Naturalnym pytaniem jest czy są przypadki, w których patrzenie jeden krok naprzód w procesie predykcyjnym jest *dokładnie najlepszym, co możemy zrobić*, nawet jeśli horyzont czasowy jest znany? Innymi słowy, kiedy strategie SNML i NML są dokładnie tożsame? Odpowiedź na to pytanie ma fundamentalne znacznie dla rozważanych problemów predykcyjnych z dwóch następujących powodów. Po pierwsze, wiemy, że w ogólnym sekwencyjnym procesie decyzyjnym uzyskanie optymalnej strategii wymaga rekursywnego rozwiązania równania Bellmana poprzez indukcję wsteczną. Pozytywna odpowiedź na powyższe pytanie oznacza, że możemy uniknąć indukcji wstecznej, ponieważ optymalna strategia staje się niezależną od horyzontu czasowego: mamy tę samą, optymalną strategię niezależnie od tego, jak daleko patrzymy w przyszłość. Możemy więc patrzeć tylko jeden krok naprzód. Po drugie, pokazano [A_3] [Hedayati i Bartlett, 2012a,b; Harremoës, 2013], że gdy strategie NML i SNML są równoważne, wtedy obie stają *strategiami bayesowskimi* z prawdopodobieństwem a priori określonym przez *rozkład Jeffreysa* [Grünwald, 2007]. Innymi słowy, jeśli strategia NML jest niezależna od horyzontu czasowego, strategia bayesowska z rozkładem a priori Jeffreysa jest strategią minimaksową. Hedayati i Bartlett [2012a,b] pokazali, że ma to miejsce wtedy i tylko wtedy gdy strategia SNML jest *wymienialna*, tzn. przydziela to samo prawdopodobieństwo do sekwencji danych niezależnie od kolejności ich wystąpienia. Testowanie wymienialności jest jednakże trudne i nie daje prostej charakteryzacji rodzin wykładniczych dla których zachodzi równość NML=SNML. W pracy przedstawiliśmy kompletną odpowiedź na to pytanie dla przypadku 1-parametrowych rodzin wykładniczych:

Twierdzenie 6 ([A_6], twierdzenie 11) *Jedyne trzy naturalne, 1-parametrowe rodziny wykładnicze z wymienialną strategią SNML to: (1) Pełna rodzina rozkładów normalnych z ustaloną wariancją $\sigma^2 > 0$; (2) Pełna rodzina rozkładów gamma z ustalonym parametrem kształtu; (3) Pełna rodzina Tweediego rzędu $\frac{3}{2}$.*

Dodatkowo, wszystkie rodziny otrzymane z powyższych trzech przez jedno-jednoznaczną transformację zmiennej y mają wymienialną strategię SNML (np. rozkłady Pareto, Laplace'a, Rayleighta, odwrotny normalny i wiele innych). Tym samym, tylko w powyższych trzech naturalnych rodzinach wykładniczych predykcje strategii SNML są tożsame z predykcjami minimaksowej NML i strategii bayesowskiej (z rozkładem Jeffreysa). Oznacza to, że tylko w tych rodzinach strategia NML nie zależy od horyzontu czasowego i patrzenie jeden krok naprzód jest równoważne z patrzenie t kroków naprzód dla dowolnego $t \geqslant 1$.

Strategia Flattened ML. [A_1] [A_2] Widzieliśmy już, że wszystkie strategie, które przewidują za pomocą jednego z rozkładów w $\mathcal{F}$ (strategie *plug-in*), choć łatwe do wyznaczenia, są koniecznie suboptymalne. Z drugiej strony, strategie optymalne, jak SNML, mogą być kosztowne obliczeniowo (np. obliczenie stałej normalizacyjnej SNML wymaga całkowania po $\mathcal{Y}$). Grünwald [2007] postawił pytanie, czy istnieje modyfikacja strategii *plug-in*, której predykcje wykraczają nieznacznie poza $\mathcal{F}$, a która zarazem posiada optymalne gwarancje na żal. Wykazaliśmy, że hipoteza ta jest prawdziwa, proponując strategię *prawie* typu *plug-in*, nazwaną *Flattened Maximum Likelihood* (FML), która jest równie prosta do wyznaczenia co strategia ML, a zarazem gwarantuje minimaksowy żal $\frac{n}{2}\log T$ z dokładnością do stałej. FML przewiduje zgodnie z rozkładem największej wiarygodności (ML) $f_{\widehat{\theta}_{t-1}}$, dodatkowo „spłaszczając" rozkład poprzez przemnożenie przez człon rzędu $1 + O(1/n)$ zawierający wektor wartości średnich i macierz kowariancji (wyznaczone dla estymatora największej wiarygodności). Spłaszczenie to powoduje zwiększenie wariancji rozkładu, co daje strategii FML odporność na dane najgorszego przypadku.

Twierdzenie 7 ([A_2], twierdzenie 11) *Niech $\mathcal{F}$ będzie n-parametrową rodziną wykładniczą i niech $y_{1:T}$ będzie ograniczoną sekwencją danych ($\|y_t\| \leqslant B$ dla wszystkich t), dla której dla wszystkich $t \geqslant t_0$, estymator największej wiarygodności $\widehat{\theta}_{t-1} \in \Theta_0$, gdzie Θ_0 jest dowolnym wypukłym i zwartym podzbiorem. Wtedy:*

$$R(\widehat{p};\, y_{1:T}) = \frac{n}{2}\log T + O(1),$$

gdzie $\widehat{p}$ jest strategią FML, a stała w $O(1)$ zależy od B, Θ_0 i t_0.

Warunek o przynależności estymatora największej wiarygodności do Θ_0 ma znaczenie techniczne i jest praktycznie zawsze spełniany (ma tylko na celu eliminację patologicznych, np. rozbieżnych, sekwencji). Aby uzyskać optymalną gwarancję na żal, FML potrzebuje założenia o ograniczeniu danych (podczas gdy SNML tego nie potrzebuje), co jest najwyraźniej ceną za prostotę obliczeniową. W praktyce, jednakże, ten warunek jest również zawsze spełniony. Na koniec zauważamy, że strategia FML może być wyznaczona jako przybliżenie drugiego rzędu strategii SNML [A_2].

4.3.3 Przyrostowe uczenie się macierzy

W problemach uczenia się macierzy zarówno dane wyjściowe $\boldsymbol{Y}_t$ jak i akcje strategii predykcyjnych $\widehat{\boldsymbol{P}}_t$ ($t = 1, \ldots, T$) mają postać macierzy. W ostatnich latach problemy te zyskały popularność [Tsuda i inni, 2005; Warmuth i Kuzmin, 2008], ponieważ licznie występują w praktycznych zastosowaniach: systemach rekomendacyjnych [Bennett i Lanning, 2007], grupowaniu spektralnym [von Luxburg, 2007], klasyfikacji wieloetykietowej [Langford i inni, 2009], czy nawet w kwantowej tomografii [Nielsen i Chuang, 2000; Paris i Rehacek, 2004].

Macierzowa logarytmiczna funkcja straty. [A_4] W pracy rozszerzyliśmy na przypadek macierzowy problem uczenia się rozkładu wielomianowego na n elementach. W klasycznym przypadku rozkład jest parametryzowany przez n-wymiarowy wektor prawdopodobieństwa $\widehat{\boldsymbol{p}}$. Można interpretować wektory bazowe $\boldsymbol{e}_k$ jako zdarzenia elementarne (gdzie $\boldsymbol{e}_k$ jest wektorem jednostkowym z $e_{k,k} = 1$ i $e_{k,j} = 0$ dla $j \neq k$), a wektor prawdopodobieństwa jako ich mieszaninę, $\widehat{\boldsymbol{p}} = \sum_k \widehat{p}_k \boldsymbol{e}_k$. Zaproponowaliśmy macierzowe uogólnienie rozkładu wielomianowego na *macierz gęstości* $\widehat{\boldsymbol{P}}$ [Nielsen i Chuang, 2000], symetryczną, dodatnio określoną macierz o jednostkowym śladzie, $\widehat{\boldsymbol{P}} \succeq \boldsymbol{0}$, $\mathrm{tr}(\widehat{\boldsymbol{P}}) = 1$. O ile wektory prawdopodobieństwa reprezentują niepewność algorytmu odnośnie n wektorów bazowych, macierze gęstości reprezentują niepewność odnośnie nieskończenie wielu *iloczynów diadycznych*, tj. iloczynów zewnętrznych postaci $\boldsymbol{u}\boldsymbol{u}^\top$, $\boldsymbol{u} \in \mathbb{R}^n$, które mogą być utożsamione z jednowymiarowymi podprzestrzeni. Macierz gęstości jest mieszaniną takich iloczynów.

W przypadku klasycznym (wektorowym) akcją algorytmu jest wektor prawdopodobieństw $\widehat{\boldsymbol{p}}_t$, środowisko ujawnia wartość wyjścia $y_t \in \{1, \ldots, n\}$ i algorytm jest karany stratą $-\log(\widehat{p}_{t,y_t})$. W przypadku macierzowym, akcją algorytmu $\widehat{\boldsymbol{P}}_t$ i wartością wyjścia $\boldsymbol{Y}_t$ są macierze gęstości, a funkcja straty jest macierzową wersją logarytmicznej straty $\ell(\boldsymbol{Y}_t, \widehat{\boldsymbol{P}}_t) = -\mathrm{tr}(\boldsymbol{Y}_t \mathbf{log}(\widehat{\boldsymbol{P}}_t))$, gdzie $\mathbf{log}(\cdot)$ oznacza logarytm macierzowy. Ta strata sprowadza się do klasycznej logarytmicznej straty, gdy macierz $\widehat{\boldsymbol{P}}_t$ jest diagonalna (jako, że wartości na diagonali tworzą wektor prawdopodobieństw na n elementach), a $\boldsymbol{Y}_t$ jest iloczynem diadycznym $\boldsymbol{e}_k \boldsymbol{e}_k^\top$ dla pewnego $k = 1, \ldots, n$. Podczas gdy w przypadku wektorowym przestrzenią akcji $\mathcal{A}$ i referencyjną rodziną strategii $\mathcal{F}$ są rodziny rozkładów wielomianowych, w przypadku macierzowym $\mathcal{A}$ i $\mathcal{F}$ są zbiorami macierzy gęstości.

Dwie najpopularniejsze strategie predykcyjne w klasycznym przypadku to estymatory Laplace'a oraz Kryczewskiego-Trofimowa (KT) [Cesa-Bianchi i Lugosi, 2006], które można podsumować jako reguły „plus c": predykcja algorytmu $\widehat{\boldsymbol{p}}_t$ jest otrzymywana przez zliczanie przeszłych wystąpień poszczególnych elementów i dodanie c do tych zliczeń; formalnie: $\widehat{p}_{t,k} = \frac{m_k + c}{t-1+nc}$, gdzie m_k jest liczbą wystąpień k-tego elementu w poprzednich $t-1$ iteracjach. Estymator Laplace'a używa $c = 1$, zaś KT wybiera $c = \frac{1}{2}$. Oba algorytmy są strategiami bayesowskimi, z bardzo dobrymi ograniczeniami na żal, rosnącymi logarytmicznie z T, a estymator KT osiąga wręcz żal minimaksowy z dokładnością do stałej [Cesa-Bianchi i Lugosi, 2006]. Interesowało nas, jak zmieni się żal, gdy algorytmy te uogólnimy na przypadek macierzowy. Zdefiniowaliśmy więc macierzową wersję reguły „plus c": $\widehat{\boldsymbol{P}}_t = \frac{\sum_{i=1}^{t-1} \boldsymbol{Y}_i + c\boldsymbol{I}}{t-1+nc}$, otrzymując macierzowy estymator Laplace'a dla $c = 1$ i macierzowy estymator KT dla $c = \frac{1}{2}$. Co zaskakujące, pokazaliśmy, że żal najgorszego przypadku dla macierzowych estymatorów jest *taki sam* jak dla estymatorów klasycznych:

Twierdzenie 8 ([A_4], twierdzenia 1 and 2) *Wartości żalu najgorszego przypadku dla macierzowych i klasycznych wersji estymatorów Laplace'a i KT są identyczne.*

Pokazaliśmy również, że inny algorytm, zwany *last-step minimax* [Takimoto i Warmuth, 2000], który jest szczególnym przypadkiem wcześniej omawianego algorytmu SNML, ma również to samo ograniczenie

na żal w przypadku klasycznym i macierzowym ([A_4], twierdzenie 3). Nasze wyniki po razy pierwszy identyfikują podstawowy problem predykcji, w którym żal osiągany w macierzowej wersji problemu jest identyczny z tym osiąganym w wersji klasycznej. Ponieważ klasyczny rozkład na n elementach odpowiada wartościom własnym macierzy gęstości, oznacza to, że algorytmy macierzowe uczą się systemu wektorów własnych „za darmo", bez ponoszenia dodatkowego żalu.

Przyrostowe PCA i jego uogólnienie. [A_9] *Analiza głównych składowych* (*Principal Component Analysis*, PCA) [Hotelling, 1933] jest jedną z najbardziej popularnych metod analizy, kompresji i wizualizacji danych. Formalnie, w problemie PCA[3] mamy zbiór T n-wymiarowych obserwacji $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$, które chcemy rzutować na podprzestrzeń o wymiarze k ($k \ll n$), reprezentowaną przez macierz rzutowania rzędu k, oznaczaną $\boldsymbol{P}$, w ten sposób, aby maksymalizować całkowity kwadrat długości rzutowanych obserwacji $\sum_{t=1}^{T} \|\boldsymbol{P}\boldsymbol{y}_t\|^2$. Problem ten jest równoważny znalezieniu wektorów własnych $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ stowarzyszonych z k największymi wartościami własnymi „macierzy kowariancji" $\sum_{t=1}^{T} \boldsymbol{y}_t\boldsymbol{y}_t^\top$, i wyborze macierzy rzutowania jako $\boldsymbol{P} = \sum_{i=1}^{k} \boldsymbol{u}_i\boldsymbol{u}_i^\top$.

W *przyrostowym* problemie PCA [Warmuth i Kuzmin, 2008], algorytm otrzymuje dane w sposób sekwencyjny. W chwilach $t = 1, \ldots, T$, algorytm wybiera macierz rzutowania rzędu k oznaczaną $\widehat{\boldsymbol{P}}_t$, a po ujawnieniu następnej obserwacji $\boldsymbol{y}_t$ jest karany *błędem (stratą) kompresji* $\widehat{\ell}_t = \|\boldsymbol{y}_t - \widehat{\boldsymbol{P}}_t\boldsymbol{y}_t\|^2 = \mathrm{tr}((\boldsymbol{I} - \widehat{\boldsymbol{P}}_t)\boldsymbol{y}_t\boldsymbol{y}_t^\top)$. Żalem algorytmu nazywamy różnicę między jego sumarycznym błędem kompresji a sumarycznym błędem kompresji najlepszej (z perspektywy czasu) macierzy rzutowania rzędu k (która jest dokładnie rozwiązaniem standardowego problemu PCA). Otrzymujemy w ten sposób problem uczenia się macierzy, w którym macierze wyjścia mają postać $\boldsymbol{Y}_t = \boldsymbol{y}_t\boldsymbol{y}_t^\top$, funkcja straty jest *liniowa* względem $\widehat{\boldsymbol{P}}_t$ i $\boldsymbol{Y}_t$, a akcją algorytmu uczącego się ($\widehat{\boldsymbol{P}}_t$) jest *randomizowany* wybór macierzy rzutowania rzędu k. Ten randomizowany wybór odpowiada powłoce wypukłej zbioru macierzy rzutowania rzędu k, oznaczanej $\mathcal{F}_k$, czyli zbiorowi macierzy dodatnio określonych ze śladem równym k, i wartościami własnymi nie większymi od 1. Uogólniliśmy również problem przyrostowego PCA na przypadek, w którym obserwowane macierze $\boldsymbol{Y}_t$ są dowolnymi dodatnio określonymi macierzami z wartościami własnymi ograniczonymi w przedziale $[0, 1]$. Ten przypadek nazwaliśmy przypadkiem danych L_∞-*ograniczonych*, w przeciwieństwie do L_1-*ograniczonych* danych $\boldsymbol{y}_t\boldsymbol{y}_t^\top$ w oryginalnym problemie przyrostowego PCA (terminologia ta pochodzi od odpowiednich macierzowych norm Schattena ograniczających w obu przypadkach dane).

W poprzednich pracach uogólniono znane metody uczenia się wektorów na przypadek macierzowy, co zaowocowało algorytmami *Matrix Exponentiated Gradient* (MEG) [Tsuda i inni, 2005] oraz macierzową wersją *Online Gradient Descent* (GD) [Arora i inni, 2013]. Oba algorytmy uaktualniają swoje parametry poprzez minimalizację przetargu między *dywergencją Bregmana* (wyznaczoną pomiędzy starą i nową wartością parametrów) a wartością straty na aktualnej obserwacji dla nowego wektora parametrów [Cesa-Bianchi i Lugosi, 2006; Shalev-Shwartz, 2012]. Macierzowy GD używa dywergencji Bregmana będącej kwadratową normą Frobeniusa (uogólnieniem kwadratowej normy euklidesowej na przypadek macierzy). MEG używa kwantowej entropii względnej [Nielsen i Chuang, 2000], macierzowego uogólnienia klasycznej entropii względnej. W naszej pracy rozważaliśmy dwie wersje algorytmu MEG: wersja *gain MEG* jest parametryzowana za pomocą opisanej wyżej przestrzeni $\mathcal{F}_k$, podczas gdy wersja *loss MEG* używa alternatywnej parametryzacji, w której zamiast brać powłokę wypukła macierzy projekcji rzędu k bierzemy powłokę wypukłą *dopełnień* tych projekcji $\boldsymbol{I} - \boldsymbol{P}$, co prowadzi do przestrzeni parametrów $\mathcal{F}_m$, gdzie $m = n - k$. (w przypadku GD, obie parametryzacje prowadziłyby do tego samego algorytmu).

W pracy [A_9] wyznaczyliśmy górne ograniczenia na żal obu algorytmów zarówno dla L_1-ograniczonych jak i L_∞-ograniczonych danych, dla dowolnej wartości k. Podsumowujemy nasze wyniki w poniższej tabeli górnych ograniczeń (gdzie $m = n - k$):

	dane L_1-ograniczone		dane L_∞-ograniczone	
	$k \leqslant \frac{n}{2}$	$k \geqslant \frac{n}{2}$	$k \leqslant \frac{n}{2}$	$k \geqslant \frac{n}{2}$
Loss MEG	$\boldsymbol{\sqrt{Tk}}$	$\boldsymbol{\sqrt{Tm\left(\log \frac{n}{m}\right)/\frac{n}{m}}}$	$\sqrt{Tkm}^\dagger$	$\boldsymbol{\sqrt{Tm^2 \log \frac{n}{m}}}^\dagger$
Gain MEG	$\sqrt{Tk \log \frac{n}{k}}$	$\sqrt{Tm}$	$\boldsymbol{\sqrt{Tk^2 \ln \frac{n}{k}}}$	$\sqrt{Tkm}$
GD	$\boldsymbol{\sqrt{Tk}}^\dagger$	$\sqrt{Tm}$	$\sqrt{Tkm}$	$\sqrt{Tkm}$

[3] Dla uproszczenia prezentacji zakładamy, że wektory danych są wycentrowane.

Wszystkie wyniki powinny być czytane w notacji $O(\cdot)$ (nie podaliśmy stałych aby uprościć prezentację). Najlepsze (najmniejsze) ograniczenia w każdej kolumnie oznaczone są tłustym drukiem, a symbol † mówi, że dane ograniczenie było już znane wcześniej. Z tabeli wynika, że któraś z wersji MEG jest zawsze optymalna i stąd sugeruje to, że algorytm MEG jest lepszym wyborem niż GD dla tego typu problemów. Udowodniliśmy również *dolne* ograniczenia na żal dla *dowolnego* algorytmu, które są identyczne z najlepszymi górnymi ograniczeniami z tabelki. Tym samym, ograniczenia wyróżnione tłustym drukiem są optymalne (równe żalowi minimaksowemu) z dokładnością do stałej.

W typowych zastosowaniach PCA istnieje nisko-wymiarowa podprzestrzeń, która obejmuje większość wariancji w danych, stąd można się spodziewać, że sumaryczny błąd kompresji najlepszej podprzestrzeni (L_T^*) będzie nieduży. Tym samym skupianie się na danych najgorszego przypadku (w których błąd kompresji L_T^* będzie rósł liniowo z T) może być zbyt pesymistyczne. Z tego względu rozważaliśmy również ograniczenia na żal parametryzowane przez L_T^*, potencjalnie znacznie mniejsze od T. Pokazaliśmy, że w takim przypadku, o ile $k \leqslant \frac{n}{2}$ (co odpowiada typowemu zastosowaniu PCA), algorytm MEG jest optymalny, osiągając żal $O(\sqrt{L_T^* k})$, i ściśle lepszy od algorytmu GD (o wartość rzędu $\sqrt{k}$), który może dla niektórych danych ponosić żal co najmniej $\Omega(k\sqrt{L_T^*})$. To utwierdza nas w przekonaniu, że w rozważanych problemach MEG jest lepszą metodą uczenia się od GD.

Kernelizacja algorytmów macierzowych [A_7] Następując pomysł (znany jako *kernel trick*) został spopularyzowany przez metody wektorów podpierających (*support vector machines*) [Boser i inni, 1992] i stał się jednym z bardziej użytecznych narzędzi w uczeniu maszynowym: każdy algorytm, którego działanie można zredukować wyłącznie do obliczania iloczynów skalarnych na wektorach $\boldsymbol{x} \in \mathbb{R}^n$, może zostać rozszerzony poprzez przekształcenie przenoszące wektory $\boldsymbol{x}$ do wzbogaconej przestrzeni $\phi(\boldsymbol{x}) \in \mathbb{R}^N$, o ile tylko dostępna jest funkcja jądrowa (*kernel function*) $k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}')$ umożliwiająca efektywne wyznaczanie iloczynów skalarnych we wzbogaconej przestrzeni. Takie algorytmy nazywa się *kernelizowalnymi*. W naszej pracy scharakteryzowaliśmy kernelizowalność algorytmów macierzowych poprzez ich niezmienniczość ze względu na pewne typy transformacji ortogonalnych.

Wpierw rozważaliśmy dane macierzowe będące *asymetrycznymi* iloczynami zewnętrznymi postaci $\boldsymbol{x}\boldsymbol{z}^\top$, gdzie $\boldsymbol{x} \in \mathbb{R}^n$ i $\boldsymbol{z} \in \mathbb{R}^m$. Pokazaliśmy ([$A_7$], wniosek 2.7), że algorytm jest kernelizowalny wtedy i tylko wtedy gdy jego predykcje są niezmiennicze względem transformacji postaci $\boldsymbol{x}_t \mapsto \boldsymbol{U}\boldsymbol{x}_t$, $\boldsymbol{z}_t \mapsto \boldsymbol{V}\boldsymbol{z}_t$ ($t = 1, \ldots, T$) dla dowolnych macierzy ortogonalnych $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ i $\boldsymbol{V} \in \mathbb{R}^{m\times m}$. Co więcej, pokazaliśmy ([A_7], twierdzenie 2.9), że jeśli kernelizowalny algorytm jest *liniowy*, tzn. dokonuje w chwili t predykcji danej wyrażeniem $\boldsymbol{x}_t^\top \widehat{\boldsymbol{P}}_t \boldsymbol{z}_t$ dla pewnej macierzy wag $\widehat{\boldsymbol{P}}_t \in \mathbb{R}^{n\times m}$, to $\widehat{\boldsymbol{P}}_t$ może być rozwinięta w sumę iloczynów zewnętrznych, $\widehat{\boldsymbol{P}}_t = \sum_{i,j=1}^{t-1} c_{ij} \boldsymbol{x}_i \boldsymbol{z}_j^\top$, gdzie współczynniki $\{c_{ij}\}_{i,j=1}^{t-1}$ zależą tylko od iloczynów skalarnych postaci $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ i $\boldsymbol{z}_i^\top \boldsymbol{z}_j$, $i, j = 1, \ldots, t-1$.

Następnie rozważaliśmy *symetryczne* iloczyny zewnętrzne $\boldsymbol{x}\boldsymbol{x}^\top$, gdzie $\boldsymbol{x} \in \mathbb{R}^n$. Pokazaliśmy ([$A_7$], wniosek 2.11), że algorytm jest kernelizowalny wtedy i tylko wtedy gdy jego predykcje są niezmiennicze względem transformacji $\boldsymbol{x}_t \mapsto \boldsymbol{U}\boldsymbol{x}_t$ dla dowolnych macierzy ortogonalnych $\boldsymbol{U} \in \mathbb{R}^{n\times n}$. Co więcej, pokazaliśmy ([A_7], twierdzenie 2.13), że jeśli kernelizowalny algorytm jest liniowy, to jego symetryczna macierz wag może zostać rozwinięta w sumę iloczynów zewnętrznych *oraz* wielokrotności macierzy jednostkowej $\boldsymbol{I}$, tzn. $\widehat{\boldsymbol{P}}_t = \sum_{i,j=1}^{t-1} c_{ij} \boldsymbol{x}_i \boldsymbol{x}_j^\top + c\boldsymbol{I}$, gdzie współczynniki $\{c_{ij}\}_{i,j=1}^{t-1}$ i c zależą tylko od iloczynów skalarnych postaci $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ $i, j = 1, \ldots, t-1$. Wyniki te nie były wcześniej znane.

Mając powyższą charakteryzację, możemy użyć przekształceń $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$ oraz $\boldsymbol{z} \mapsto \psi(\boldsymbol{z})$, i tak długo, jak potrafimy efektywnie policzyć iloczyny skalarne (funkcje jądrowe) $\phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}')$ oraz $\psi(\boldsymbol{z})^\top \psi(\boldsymbol{z}')$, możemy zaimplementować kernelizowalny macierzowy algorytm przyrostowy używając wyłącznie obliczeń za pomocą funkcji jądrowych. Użyliśmy powyższych wniosków do kernelizacji dwóch popularnych algorytmów przyrostowego uczenia się macierzy, MEG i GD. Kernelizacja MEG była wynikiem dość zaskakującym, ponieważ wiadomo, że klasyczna wersja MEG dla wektorów (algorytm EG) nie jest kernelizowalna. Macierzowy MEG jest jednak niezmienniczy ze względu na transformacje ortogonalne danych, a tym samym – kernelizowalny. Podaliśmy również optymalne ograniczenia na żal dla rozważanych algorytmów ([A_7], rozdział 4).

Jednym ze sposobów zapewnienia kernelizowalności jest użycie Twierdzenia o Reprezentacji (*Representer Theorem*) [Kimeldorf i Wahba, 1971], które mówi, że w problemie minimalizacji sumarycznej straty na zbiorze danych (która zależy tylko od iloczynów skalarnych między wektorem wag i wektorami ob-

serwacji) wraz z funkcją kary będącą niemalejącą funkcją normy euklidesowej, optymalne rozwiązanie (wektor wag) można rozwinąć do kombinacji liniowej wektorów obserwacji. Twierdzenie to zostało później rozszerzone na przypadek zewnętrznych iloczynów macierzowych [Abernethy i inni, 2009; Argyriou i inni, 2009]. Co ciekawe, nasza geometryczna charakteryzacja kernelizowalności prowadzi natychmiast do prostszej wersji Twierdzenia o Reprezentacji dla danych macierzowych:

Twierdzenie 9 ([A_7], twierdzenie 3.1) *Niech $\mathcal{D} = \{(\boldsymbol{x}_t \boldsymbol{z}_t^\top, y_t)\}_{t=1}^T$ będzie sekwencją etykietowanych danych. Rozważmy problem minimalizacji $\min_{\boldsymbol{P}} \mathcal{L}(\boldsymbol{P}, \mathcal{D})$, który dla dowolnego $\mathcal{D}$ ma jednoznaczne rozwiązanie i jest* niezmienniczy*: dla każdego $\mathcal{D}$ i macierzy ortogonalnych $\boldsymbol{U}$ i $\boldsymbol{V}$, $\mathcal{L}(\boldsymbol{P}, \mathcal{D}) = \mathcal{L}(\boldsymbol{U}\boldsymbol{P}\boldsymbol{V}^\top, \boldsymbol{U}\mathcal{D}\boldsymbol{V}^\top)$, gdzie $\boldsymbol{U}\mathcal{D}\boldsymbol{V}^\top = \{(\boldsymbol{U}\boldsymbol{x}_t \boldsymbol{z}_t^\top \boldsymbol{V}^\top, y_t)\}_{t=1}^T$. Wtedy optymalne rozwiązanie $\widehat{\boldsymbol{P}}$ można rozwinąć w sumę $\widehat{\boldsymbol{P}} = \sum_{i,j=1}^T c_{ij} \boldsymbol{x}_i \boldsymbol{z}_j^\top$, gdzie $\{c_{ij}\}_{i,j=1}^T$ zależą tylko od iloczynów skalarnych między wektorami obserwacji.*

Nasze metody prowadzą również do nieznanej wcześniej wersji Twierdzenia o Reprezentacji dla macierzy symetrycznych, kluczowej dla algorytmów typu MEG ([A_7], twierdzenie 3.2).

4.3.4 Przyrostowe uczenie się niezmiennicze ze względu na skalę [A_{13}]

Rozważaliśmy następujący wariant tzw. *przyrostowej optymalizacji wypukłej* (*online convex optimization*) [Shalev-Shwartz, 2012]. W chwilach $t = 1, \ldots, T$, algorytm otrzymuje wektor wejściowy $\boldsymbol{x}_t \in \mathcal{X} = \mathbb{R}^n$, na którym przewiduje $\widehat{p}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t$ za pomocą wektora wag $\boldsymbol{w}_t \in \mathbb{R}^n$. Następnie ujawniane jest wyjście $y_t \in \mathcal{Y}$ i algorytm karany jest stratą $\ell(y_t, \widehat{p}_t)$ wypukłą względem $\widehat{p}_t$. Jakość algorytmu oceniana jest przez żal, będący różnicą między sumaryczną stratą algorytmu a sumaryczną stratą predykcji uzyskanych przez ustalony wektor wag $\boldsymbol{w} \in \mathbb{R}^n$. Tym samym rodzina referencyjnych strategii predykcyjnych $\mathcal{F} = \{p(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{w} \colon \boldsymbol{w} \in \mathbb{R}^n\}$ jest rodziną funkcji liniowych. Problem ten obejmuje wiele zadań uczenia maszynowego, takich jak np. liniowa klasyfikacja (z zastępczymi funkcjami straty) lub liniowa regresja.

Większość wcześniejszych prac w tej dziedzinie zakłada, że wektory wejściowe i referencyjne wektory wag są *ograniczone*, a ograniczenie to jest znane algorytmowi uczącemu się. W praktyce założenia takie są rzadko kiedy uzasadnione: algorytm nie ma zwykle informacji o wielkości wektorów wejściowych, a założenie znajomości ograniczenia na referencyjne wektory wag (w tym optymalny z perspektywy czasu wektor) jest jeszcze mniej realistyczne. Stąd w ostatnim czasie pojawiło się kilka prac, w których starano się odrzucić część z tych założeń [Streeter i McMahan, 2012; McMahan i Orabona, 2014; Orabona i Pál, 2016; Luo i inni, 2016].

W naszej pracy idziemy krok dalej, całkowicie odrzucają wszystkie ograniczenia nałożone na dane i referencyjne wektory wag, pokazując, że efektywne uczenie się nadal jest możliwe. W tym celu wykorzystujemy naturalną symetrię przy braku ograniczeń, dotyczącą *niezmienniczości ze względu na skalę*: predykcje optymalnego wektora wag dla danej sekwencji danych są niezmiennicze względem dowolnych transformacji liniowych danych wejściowych, ponieważ optymalny wektor wag zostanie wtedy przeskalowany przez transformację odwrotną. Naszym celem była konstrukcja algorytmów również posiadających własność takiej niezmieniczości. Rozpoczęliśmy od przypadku transformacji po cechach, w których współrzędne wektorów wejściowych (cechy) mogą być dowolnie przeskalowane: $x_{t,i} \mapsto a_i x_{t,i}$ dla wszystkich t i dowolnych dodatnich a_i, $i = 1, \ldots, n$. Zaproponowaliśmy algorytm, którego predykcje są niezmiennicze ze względu na takie transformacje (który nazywamy po prostu *Algorytmem 1*). Jest to własność wyjątkowo użyteczna, gdyż nasz algorytm nie wymaga wstępnej normalizacji danych, w odróżnieniu od typowych algorytmów przyrostowych, jak np. *Stochastic Gradient Descent*, których działanie krytycznie zależy od normalizacji. Nasz algorytm potrzebuje tylko $O(n)$ obliczeń na iterację, nie wymaga strojenia żadnych parametrów i osiąga zasadniczo optymalne gwarancje na żal, wyrażone poprzez niezmienniczą funkcję danych i referencyjnego wektora wag:

Twierdzenie 10 ([A_{13}], twierdzenie 4) *Dla dowolnej sekwencji danych $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$ i dowolnego referencyjnego wektora wag $\boldsymbol{w} \in \mathbb{R}^n$, Algorytm 1 osiąga ograniczenie na żal postaci:*

$$R(\widehat{p};\, \boldsymbol{x}_{1:T}, y_{1:T};\, \boldsymbol{w}) \leqslant \sum_{i=1}^n |w_i| s_{T,i} \sqrt{\alpha \log(1 + \alpha d^2 T^2 w_i^2 s_{T,i}^2)} + O(\log T),$$

gdzie $\alpha = 9/8$, a wielkości $s_{T,i} = \sqrt{\sum_{t=1}^T x_{t,i}^2}$ mierzą rozrzut na poszczególnych cechach.

Przemnożenie każdej ze współrzędnych w_i przez $s_{T,i}$ zapewnia, że ograniczenie na żal nie zmienia się przy skalowaniu cech (jako, że referencyjny wektor zostałby wtedy przeskalowany przez transformację odwrotną aby prowadzić do tych samych predykcji).

Następnie rozważaliśmy ogólną klasę transformacji linowych danych wejściowych postaci $\boldsymbol{x}_t \mapsto \boldsymbol{A}\boldsymbol{x}_t$ dla wszystkich t. Wpierw udowodniliśmy negatywny wynik, mówiący że żaden algorytm nie uzyska nietrywialnego niezmienniczego ograniczenia na żal w tym przypadku. Następnie zaproponowaliśmy jednak algorytm, którego predykcje są niezmienniczego ze względu na dowolne transformacje liniowe (nazwany po prostu *Algorytmem 2*), który „prawie" osiąga założone niezmiennicze ograniczenie na żal, z dodatkowym członem w ograniczeniu, który tylko logarytmicznie zależy od wielkości danych wejściowych:

Twierdzenie 11 ([A_{13}], twierdzenie 9) *Dla dowolnej sekwencji danych $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$ i dowolnego wektora wag $\boldsymbol{w} \in \mathbb{R}^n$, Algorytm 2 osiąga ograniczenie na żal postaci:*

$$R(\widehat{p};\, \boldsymbol{x}_{1:T}, y_{1:T};\, \boldsymbol{w}) \leqslant \|\boldsymbol{w}\|_{\boldsymbol{S}_T} \sqrt{\alpha \log(1 + \alpha \|\boldsymbol{w}\|^2_{\boldsymbol{S}_T}) + \Gamma_T} + 1,$$

gdzie $\alpha = 9/8$, $\|\boldsymbol{w}\|_{\boldsymbol{S}_T} = \sqrt{\boldsymbol{w}^\top \boldsymbol{S}_T \boldsymbol{w}}$ jest seminormą indukowaną przez „macierz kowariancji" $\boldsymbol{S}_T = \sum_{t=1}^T \boldsymbol{x}_t \boldsymbol{x}_t^\top$, a Γ_T zależy logarytmicznie od wielkości wektorów wejściowych.

Żal wyrażony jest poprzez seminormę $\|\boldsymbol{w}\|_{\boldsymbol{S}_T}$, która jest niezmiennicza ze względu na dowolnego transformacje liniowe wektorów wejściowych (ponieważ wektor wag $\boldsymbol{w}$ przeskaluje się wtedy poprzez transformację odwrotną aby zachować te same predykcje). Zaproponowany algorytm potrzebuje $O(n^2)$ obliczeń na iterację i nie wymaga strojenia żadnych parametrów.

4.3.5 Uczenie się funkcji monotonicznych

Niech $x_1 \leqslant x_2 \leqslant \ldots \leqslant x_T$ będzie zbiorem T liniowo uporządkowanych punktów (np. na prostej), oznaczonym przez X. Funkcję $f\colon X \to \mathbb{R}$ nazywamy *izotoniczną* (niemalejącą) na X gdy $f(x_i) \geqslant f(x_j)$ dla dowolnych $x_i \geqslant x_j$. Mając dane $\{(x_t, y_t)\}_{t=1}^T$, *problem izotonicznej regresji* dotyczy znalezienia funkcji f minimalizującej całkowity błąd kwadratowy na danych, $\sum_{t=1}^T (y_t - f(x_t))^2$ [Ayer i inni, 1955; Robertson i inni, 1998]. Taka funkcja nazywana jest *izotoniczną funkcją regresji*. Izotoniczna regresja ma fundamentalne znaczenie w statystyce i uczeniu maszynowym, ponieważ ograniczenia izotoniczne występują naturalnie w wielu problemach statystycznych (np. estymacja monotonicznych funkcji gęstości [Robertson i inni, 1998], kalibracja prawdopodobieństw warunkowych klas [Zadrozny i Elkan, 2002], skalowanie wielowymiarowe [Kruskal, 1964]), jak również w wielu zastosowaniach, np. w biologii, medycynie, psychologii. Mimo prostoty i dużego znaczenia praktycznego, izotoniczna regresja jest przykładem problemu *nieparametrycznego*, w którym liczba parametrów rośnie liniowo z liczbą danych.

Przyrostowa regresja izotoniczna [A_{10}]. W naturalny sposób nasuwa się pytanie: czy istnieją dla problemu izotonicznej regresji efektywne algorytmy przyrostowe z dobrymi ograniczeniami na żal? W przyrostowej wersji problemu izotonicznej regresji środowisko wybiera zbiór $X = \{x_1, \ldots, x_T\}$ i przekazuje go algorytmowi na starcie. Następnie, w każdej chwili $t = 1, \ldots, T$, środowisko wybiera jeden z jeszcze nie etykietowanych punktów x_{i_t}, $i_t \in \{1, \ldots, T\}$, na którym algorytm przewiduje $\widehat{p}_{i_t} \in [0, 1]$. Wtedy ujawniona jest prawdziwa wartość wyjścia („etykieta") $y_{i_t} \in [0, 1]$, a algorytm otrzymuje wartość kwadratowej straty $(y_{i_t} - \widehat{p}_{i_t})^2$. Tym samym algorytm przewiduje ostatecznie wartości na wszystkich punktach $x_1, \ldots x_T$, ale w kolejności wyznaczonej przez środowisko. Celem algorytmu jest niska wartość żalu:

$$R(\widehat{p};\, y_{1:T}) = \sum_{t=1}^T (y_{i_t} - \widehat{p}_{i_t})^2 - \min_{f \in \mathcal{F}} \sum_{t=1}^T (y_{i_t} - f(x_{i_t}))^2,$$

gdzie $\mathcal{F}$ jest rodziną wszystkich funkcji izotonicznych na X. Zauważmy, że człon odejmowany w wyrażeniu na żal jest dokładnie wartością błędu optymalnej izotonicznej funkcji regresji w klasycznym problemie izotonicznej regresji. Podkreślamy, że wartości wyjścia nie muszą być izotoniczne względem X. Czytelnik może zastanawiać się, dlaczego zbiór X jest ujawniony algorytmowi zawczasu? Otóż pokazaliśmy, że bez tej wstępnej informacji problem jest zbyt trudno i dowolny algorytm będzie obarczony liniowym żalem w najgorszym przypadku.

Ponieważ problem przyrostowej regresji izotonicznej dotyczy minimalizacji wypukłej funkcji straty na wypukłym zbiorze funkcji izotonicznych, może być analizowany przy użyciu narzędzi przyrostowej optymalizacji wypukłej [Shalev-Shwartz, 2012]. Niestety, pokazaliśmy, że większość popularnych algorytmów przyrostowych zawodzi w tym problemie, dając liniową wartość żalu lub, w najlepszym przypadku, suboptymalną wartość $O(\sqrt{T})$ ([A_{10}], rozdział 3). Czyni to naszym zdaniem problem przyrostowej regresji izotonicznej szczególnie interesującym i wymagającym.

W pracy przedstawiliśmy algorytm, który działa dla tego problemu i osiąga optymalną gwarancję na żal. Nasz algorytm jest wersją metody *Exponential Weights* [Cesa-Bianchi i Lugosi, 2006], uruchomionej na *dyskretnej* siatce funkcji izotonicznych. Podstawowym pomysłem jest tutaj dyskretyzacja zbioru wszystkich funkcji izotonicznych $\mathcal{F}$ z rozdzielczością rzędu $\frac{1}{K}$, dla pewnej wartości całkowitej K (dostrojonej później), co daje skończoną siatkę pokrywającą $\mathcal{F}$:

$$\mathcal{F}_K := \Big\{ f \in \mathcal{F} \colon f(x_t) = \frac{k_t}{K} \text{ dla pewnych } k_t \in \{0, \dots, K\},\, k_1 \leqslant \dots \leqslant k_T \Big\}.$$

Na tym zbiorze uruchamiamy algorytm *Exponential Weights*, będący bayesowską strategią uczenia się, która rozpoczyna z jednostajnym prawdopodobieństwem a priori $\pi_1(f) = \frac{1}{|\mathcal{F}_K|}$ na $f \in \mathcal{F}_K$ i uaktualnia rozkład a posteriori $\pi_t(f)$ poprzez przemnażanie prawdopodobieństw funkcji proporcjonalnie do wykładniczej wartości ich negatywnej straty. Predykcją algorytmu jest średnia predykcja funkcji z $\mathcal{F}_K$, ważona rozkładem π_t:

$$\widehat{p}_{i_t} = \sum_{f \in \mathcal{F}_K} f(x_{i_t}) \pi_t(f), \qquad \text{gdzie} \quad \pi_t(f) \propto e^{-\frac{1}{2}\sum_{j=1}^{t-1} (f(x_{i_j}) - y_{i_j})^2}.$$

Mimo, że algorytm wydaje się trudny do implementacji, pokazaliśmy, że rozkład a posteriori może być uaktualniany w czasie $O(TK)$ poprzez użycie programowania dynamicznego ([A_{10}], rozdział 4.1). Udowodniliśmy, że:

Twierdzenie 12 ([A_{10}], twierdzenie 4) *Biorąc $K = \Theta(T^{1/3} \log T^{-1/3})$, algorytm* Exponential Weights *uruchomiony na dyskretnej siatce $\mathcal{F}_K$ osiąga ograniczeni na żal:*

$$R(\widehat{p};\, y_{1:T}) \leqslant O\big(T^{1/3} (\log T)^{2/3}\big).$$

Ograniczenie to okazało się optymalne z dokładnością do czynników logarytmicznych. Pokazaliśmy również ([A_{10}], twierdzenie 8), że podobne ograniczenie (również optymalne) można uzyskać bardzo podobnym algorytmem dla innych funkcji strat, np. logarytmicznej funkcji straty.

Klasyfikacja porządkowa z ograniczeniami monotonicznymi [A_5]. W pracy rozważaliśmy problemy predykcji z funkcjami izotonicznymi, w których etykiety należą do zbioru K *uporządkowanych* indeksów klas, $y_t \in \{1, \dots, K\}$. Problemy takie nazywa się *klasyfikacją porządkową z ograniczeniami monotonicznymi*. Pojawiają się w sposób naturalny w przypadkach, gdy dostępna jest wiedza dziedzinowa na temat zależności monotonicznych między wartościami na wejściu (wektorami cech), a etykietami klas (np. „im wyższy dług firmy, tym większa szansa jej zbankrutowania", itp.) Tego typu wiedza jest szczególnie istotna w problemach decyzyjnych dotyczących preferencji, np. w teorii społecznego wyboru, wielokryterialnym podejmowaniu decyzji lub w uczeniu się preferencji [Greco i inni, 2001; Fürnkranz i Hüllermeier, 2011].

W celu uwzględnienia ogólniejszego modelu, zakładamy teraz, że dane wejściowe są opisane za pomocą n cech, $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, i istnieje częściowy porządek, zwany *relacją dominacji* $\succeq$, zdefiniowany następująco: dla dowolnych $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, $\boldsymbol{x} \succeq \boldsymbol{x}'$, gdy $x_j \geqslant x'_j$ dla wszystkich $j = 1, \dots, n$. Funkcję $f \colon \mathcal{X} \to \{1, \dots, K\}$ nazywamy izotoniczną gdy dla dowolnych $\boldsymbol{x}, \boldsymbol{x}'$ takich, że $\boldsymbol{x} \succeq \boldsymbol{x}'$, zachodzi $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}')$. Zwróćmy uwagę, że jest to naturalne uogólnienie poprzedniego problemu liniowo uporządkowanych wejść. W naszej pracy skupiliśmy się na stochastycznym wariancie problemu, w którym obserwacje $(\boldsymbol{x}, y)$ generowane są z nieznanego rozkładu prawdopodobieństwa $P(\boldsymbol{x}, y)$. Aby uwzględnić wynikające z wiedzy dziedzinowej związki monotoniczne, zakładamy, że rozkład $P(\boldsymbol{x}, y)$ jest *ograniczony monotonicznie*, tzn. $\boldsymbol{x} \succeq \boldsymbol{x}'$ implikuje $P(y \geqslant k|\boldsymbol{x}) \geqslant P(y \geqslant k|\boldsymbol{x}')$ dla wszystkich $k = 1, \dots, K$. Warunek ten równoważny jest *relacji stochastycznej dominacji (pierwszego rzędu)* [Levy, 1998].

Mając $t-1$ poprzednich obserwacji $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{t-1}$, celem jest predykcja etykiety nowego punktu $\boldsymbol{x}_t$ przez $\widehat{p}_t = f(\boldsymbol{x}_t)$, używając pewnej funkcji monotonicznej (klasyfikatora) $f \colon \mathcal{X} \to \{1, \dots, K\}$. Trafność

predykcji mierzona jest za pomocą *oczekiwanej wartości straty* klasyfikatora f, $L(f;P) = \mathbb{E}_P\left[\ell(y, f(\boldsymbol{x}))\right]$, dla pewnej *macierzy straty* $\ell(y, y')$ rozmiaru $K \times K$. Aby wziąć pod uwagę porządek na etykietach, zakładamy, że L jest *V-kształtne*, tzn. $\ell(y, y')$ nie powinno maleć gdy y' oddala się od y. Niech $f^* = \arg\min_f L(f;P)$ będzie optymalną funkcją predykcji. Pokazaliśmy ([A_5], twierdzenie 1), że f^* jest izotoniczne wtedy i tylko wtedy gdy macierz strat spełnia pewne specyficzne własności, całkowicie charakteryzując klasę macierzy strat prowadzących do izotonicznych optymalnych funkcji predykcyjnych. Przykładowo, macierze postaci $\ell(y, y') = |y - y'|^q$ z $q \geqslant 1$ spełniają te własności, jak również *liniowa funkcja straty* o ogólnej postaci $\ell(y, y') = \alpha(y' - y)_+ + (1 - \alpha)(y - y')_+$ dla dowolnego $\alpha \in (0, 1)$, gdzie $(x)_+ = \max\{x, 0\}$ oznacza dodatnią część x.

Dalsze wyniki dotyczą nieparametrycznych metod predykcji za pomocą funkcji izotonicznych. Pierwsza z metod, nazwana *metodą plug-in*, estymuje prawdopodobieństwa warunkowe klas $P(y|\boldsymbol{x})$ za pomocą zbioru K estymatorów $\widehat{p}_{t,1}(\boldsymbol{x}), \ldots, \widehat{p}_{t,K}(\boldsymbol{x})$, gdzie $\widehat{p}_{t,k}(\boldsymbol{x})$, estymator $P(y = k|\boldsymbol{x})$ bazujący na $t-1$ poprzednich danych, jest funkcją izotoniczną. Jeśli takie estymatory są dostępne, predykcja nowej etykiety dla danego $\boldsymbol{x}$ można zostać wyznaczona poprzez wybór etykiety minimalizującej wartość straty oczekiwaną względem estymatora $\widehat{p}_{t,1}(\boldsymbol{x}), \ldots, \widehat{p}_{t,K}(\boldsymbol{x})$ (metody takie znane są właśnie pod nazwą *plug-in* [Devroye i inni, 1996]). Pokazaliśmy, że można taki estymator otrzymać poprzez rozwiązanie ciągu $K-1$ problemów izotonicznej regresji, przy czym w k-tym problemie używamy pomocniczych etykiet równych 1 gdy $y_i \geqslant k$, lub równych 0 gdy $y_i < k$ (dla każdego i). Innymi słowy, rozwiązujemy $K-1$ problemów binarnych estymacji prawdopodobieństw warunkowych przynależności do górnej lub dolnej kumulacji klas $\{y \geqslant k\}$ i $\{y < k\}$. Pomimo, że każdy z problemów rozwiązywany jest oddzielne, udowodniliśmy, że ostateczny estymator jest spójnym rozkładem prawdopodobieństwa ([A_5], twierdzenie 5).

Druga z metod, nazwana *metodą bezpośrednią*, minimalizuje wprost określoną funkcję straty:

$$\begin{array}{rl} \text{min:} & \sum_{i=1}^{t-1} \ell(y_i, f(\boldsymbol{x}_i)) \\ \text{przy ograniczeniach:} & f\colon \mathcal{X} \to \{1, \ldots, K\} \text{ jest izotoniczna.} \end{array}$$

Mimo, że minimalizacja w klasie wszystkich funkcji izotonicznych wydaje się trudna, opierając się na pracy [Chandrasekaran i inni, 2005] pokazaliśmy, że problem ten można rozwiązać za pomocą programowania liniowego. Następnie zaproponowaliśmy dodatkowe metody redukcji rozmiaru problemu bazujące na analizie relacji dominacji ([A_5], twierdzenie 6). Pokazaliśmy też, że oba podejścia – metoda *plug-in* i metoda bezpośrednia – są ze sobą głęboko powiązane:

Twierdzenie 13 ([A_5], twierdzenie 10) *Mając estymatory $\widehat{p}_{t,1}(\boldsymbol{x}), \ldots, \widehat{p}_{t,K}(\boldsymbol{x})$ z metody plug-in, dla liniowej funkcji straty rozwiązanie $\widehat{p}_t(\boldsymbol{x})$ metody bezpośredniej w punktach $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}$ równe jest:*

$$\widehat{p}_t(\boldsymbol{x}_i) = 1 + \sum_{k=2}^{K} 1_{\widehat{p}_{t,k}(\boldsymbol{x}_i) > \alpha}, \quad \textit{lub} \quad \widehat{p}_t(\boldsymbol{x}_i) = 1 + \sum_{k=2}^{K} 1_{\widehat{p}_{t,k}(\boldsymbol{x}_i) \geqslant \alpha},$$

lub jakiejkolwiek etykiecie pomiędzy tymi dwoma rozwiązaniami, gdzie 1_C jest funkcją indykatorową (równą 1 gdy C zachodzi i 0 w przeciwnym przypadku).

Twierdzenie to pokazuje, że metoda bezpośrednia przewiduje zgodnie z progowanymi wartościami estymatorów prawdopodobieństw z metody *plug-in*, przy okazji charakteryzując warunki, przy których rozwiązanie metody bezpośredniej jest jednoznaczne. Pokazaliśmy również, że żal obu metod dla danych stochastycznych maleje do zera dla dużych wartości t ([A_5], twierdzenie 12 i twierdzenie 13).

4.3.6 Redukcje żalu w klasyfikacji binarnej [A_{12}]

W problemie klasyfikacji binarnej celem jest przewidzenie dla wejścia $x \in \mathcal{X}$ binarnej etykiety $y \in \{-1, 1\}$. Jakość klasyfikatora $h\colon \mathcal{X} \to \{-1, 1\}$ na rozkładzie danych $P(x, y)$ zwykle mierzy się za pomocą *trafności klasyfikacji* $P(h(x) = y)$. Trafność klasyfikacji często nie jest jednak adekwatna do rozważanego problemu i stosuje się wtedy bardziej złożone miary jakości. Przykładowo, gdy klasy są *niezrównoważone*, często używa się miary F_β [Lewis, 1995; Nan i inni, 2012] lub miary AM (błąd zrównoważony) [Menon i inni, 2013]. Optymalizacja takich miar jest jednak dość trudna (obliczeniowo i statystycznie), ponieważ nie dekomponują się one do błędów na poszczególnych obserwacjach.

W naszej pracy rozważaliśmy maksymalizację takich uogólnionych miar jakości za pomocą *zastępczych funkcji strat*. Ograniczyliśmy się do ważnej klasy miar Ψ, które są ilorazami liniowych funkcji fałszywie pozytywnych (FP) i fałszywie negatywnych (FN) błędów klasyfikatora h:[4]

$$\Psi(h;\, P) = \frac{a_0 + a_1 \mathrm{FP}(h) + a_2 \mathrm{FN}(h)}{b_0 + b_1 \mathrm{FP}(h) + b_2 \mathrm{FN}(h)},$$

gdzie $\mathrm{FP}(h) = P(h(x) = 1 \wedge y = -1)$ i $\mathrm{FN}(h) = P(h(x) = -1 \wedge y = 1)$, a współczynniki a_i, b_i mogą zależeć od rozkładu $P(x, y)$. Takie funkcje nazywane są *liniowymi funkcjami wymiernymi*. Zawierają one wcześniej wymienione miary F_β oraz AM, jak również współczynnik podobieństwa Jaccarda, ważoną trafność klasyfikacji, i wiele innych [Koyejo i inni, 2014]. Niech $h^* = \arg\max_h \Psi(h;\, P)$ będzie optymalnym ze względu na Ψ klasyfikatorem. Definiujemy *Ψ-żal* klasyfikatora h jako $R_\Psi(h;\, P) = \Psi(h^*;\, P) - \Psi(h;\, P)$. Podobnie jak w modelu przyrostowym, żal mierzy suboptymalność h ze względu na Ψ. Ponieważ Ψ jest złożoną miarą zależną od całego rozkładu danych, bezpośrednia jej maksymalizacja jest trudna. Dlatego zwykle używa się *zastępczych* funkcji strat, tzn. strat potencjalnie mocno różniących się od Ψ, ale łatwiejszych do optymalizacji. Pojawia się jednak pytanie, czy optymalizacja funkcji zastępczej prowadzi do optymalizacji oryginalnej miary Ψ?

Pokazaliśmy, że odpowiedź na to pytanie jest pozytywna. Niech $\ell\colon \{-1, 1\} \times \mathbb{R} \to \mathbb{R}_+$ będzie zastępczą funkcją straty mierzącą jakość *ciągłych* predykcji etykiety $y \in \{-1, 1\}$. Przykładami takich funkcji jest błąd kwadratowy $\ell(y, \widehat{p}) = (y - \widehat{p})^2$ lub błąd logistyczny $\ell(y, \widehat{p}) = \log(1 + \exp(-y\widehat{p}))$ [Hastie i inni, 2003]. Mając funkcję o wartościach rzeczywistych $f\colon \mathcal{X} \to \mathbb{R}$, jej oczekiwana wartość straty zdefiniowana jest jako $L(f;\, P) = \mathbb{E}_P\left[\ell(y, f(x))\right]$. Definiujemy też ℓ-żal funkcji f jako $R_\ell(f;\, P) = L(f;\, P) - \inf_{f'} L(f';\, P)$, gdzie infimum jest względem wszystkich możliwych funkcji o wartościach rzeczywistych. Niech $h_f\colon \mathcal{X} \to \{-1, 1\}$ będzie klasyfikatorem uzyskanym z f w ten sposób, że $h_f(x) = 1$ gdy $f(x) \geqslant \theta$ i $h_f(x) = -1$ gdy $f(x) < \theta$ dla pewnego progu $\theta \in \mathbb{R}$. Pokazaliśmy, że jeśli próg θ wyznaczony jest poprzez optymalizację miary Ψ (co, mając f, można w praktyce łatwo wykonać na osobnym zbiorze walidacyjnym, jednokrotnie przechodząc wszystkie dane i wybierając najlepszą wartość θ), to Ψ-żal wynikowego klasyfikatora jest ograniczony przez ℓ-żal funkcji f.

Twierdzenie 14 ([A_{12}], twierdzenie 1) *Niech ℓ będzie silnie właściwą (strongly proper) funkcją straty [Agarwal, 2014]. Wtedy:*

$$R_\Psi(h_f;\, P) \leqslant C\sqrt{R_\ell(f;\, P)}.$$

gdzie C zależy od własności Ψ i ℓ.

Warunek na silną właściwość funkcji straty jest techniczny i nie będzie tu omawiany; większość popularnych zastępczych funkcji strat, takich jak wymieniony błąd kwadratowy czy logistyczny, spełniają tę własność. Nasz wynik jest przykładem *redukcji żalu* [Balcan i inni, 2008], w której żal (suboptymalność) ze względu na jedno kryterium jakości ograniczony jest żalem ze względu na inne kryterium. Wymieniona tu funkcja f jest dowolna – mogła być nauczona z danych, być odpowiedzią algorytmu przyrostowego, itp. Rozkład danych jest również dowolny – np. wybierając rozkład empiryczny na próbce twierdzenie będzie dawało ograniczenie na błąd na skończonym zbiorze danych.

Rozszerzyliśmy również powyższy rezultat na przypadek, gdy próg w h_f został otrzymany w sposób niedokładny, np. poprzez optymalizację na osobnej próbce walidacyjnej ([A_{12}], twierdzenie 2). Udowodniliśmy również podobne twierdzenia ([A_{12}], twierdzenia 3 i 4) dla *klasyfikacji wieloetykietowej* [Gibaja i Ventura, 2015; Dembczyński i inni, 2012], gdzie celem jest równoczesne przewidywanie wielu etykiet dla pojedynczej obserwacji: pokazaliśmy, że podobne ograniczenia na żal zachodzą w przypadku tzw. makro- i mikro-uśredniania po etykietach [Manning i inni, 2008].

5 Omówienie pozostałych osiągnięć naukowych

Przedstawiono poniżej wybrane osiągnięcia naukowe W. Kotłowskiego uzyskane po otrzymaniu tytułu doktora, które nie weszły do głównego cyklu publikacji. Opis osiągnięć opiera się na następującej liście wybranych publikacji:

[4] Używamy konwencji, w której miary jakości są funkcjami *użyteczności*, tzn. celem jest maksymalizacja Ψ.

[B_1] Dembczyński, K., Kotłowski, W. i Słowiński, R. (2009). Learning rule ensembles for ordinal classification with monotonicity constraints. *Fundamenta Informaticae*, 94(2):163–178.

[B_2] Kotłowski, W. i Słowiński, R. (2009). Rule learning with monotonicity constraints. *Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009)*, strony 537–544. ACM.

[B_3] Guţa, M. i Kotłowski, W. (2010). Quantum learning: asymptotically optimal classification of qubit states. *New Journal of Physics*, 12:123032.

[B_4] Dembczyński, K., Kotłowski, W. i Słowiński, R. (2010). ENDER - a statistical framework for boosting decision rules. *Data Mining and Knowledge Discovery*, 21(1):52–90.

[B_5] Dembczyński, K., Kotłowski, W., Słowiński, R. i Szeląg, M. (2010). Learning of rule ensembles for multiple attribute ranking problems. *Preference Learning*, strony 217–247. Springer-Verlag.

[B_6] Dembczyński, K., Kotłowski, W. i Słowiński, R. (2010). Beyond sequential covering – boosted decision rules. *Advances in Machine Learning I*, wolumen 263 of *Studies in Computational Intelligence*, strony 209–225. Springer-Verlag.

[B_7] Kotłowski, W., Dembczyński, K. i Hüllermeier, E. (2011). Bipartite ranking through minimization of univariate loss. *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, strony 1113-1120. Omnipress.

[B_8] Dembczyński, K., Kotłowski, W. i Hüllermeier, E. (2012). Consistent multilabel ranking through univariate losses. *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, strony 1319-1326. Omnipress.

[B_9] Dembczyński, K., Gaweł, P., Jaszkiewicz, A., Kotłowski, W., Kubiak, M., Susmaga, R., Wesołek, P., Wojciechowski, A. i Zielniewicz, P. (2012). Community traffic: a technology for the next generation car navigation. *Control and Cybernetics*, 41:869–883.

[B_{10}] Dembczyński, K., Jachnik, A., Kotłowski, W., Waegeman, W. i Hüllermeier, E. (2013). Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, wolumen 28 of *Journal of Machine Learning Research Workshop and Conference Proceedings*, strony 1130–1138. JMLR.

[B_{11}] Dembczyński, K., Kotłowski, W., Gaweł, P., Szarecki, A. i Jaszkiewicz, A. (2013). Matrix factorization for travel time estimation in large traffic networks. *International Conference on Artificial Intelligence and Soft Computing (ICAISC 2013)*, wolumen 7895 of *Lecture Notes in Computer Science*, strony 500–510. Springer-Verlag.

[B_{12}] Kotłowski, W. (2015). Consistent optimization of AMS by logistic loss minimization. *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, wolumen 42 of *Journal of Machine Learning Research Workshop and Conference Proceedings*, strony 99–108. JMLR.

[B_{13}] Dembczyński, K., Kotłowski, W., Waegeman, W., Busa-Fekete, R. i Hüllermeier, E. (2016). Consistency of probabilistic classifier trees. *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2016)*, wolumen 9852 of *Lecture Notes in Computer Science*, strony 511–526. Springer-Verlag.

[B_{14}] Dembczyński, K., Kotłowski, W., Koyejo, O. i Natarajan, N. (2017). Consistency analysis for binary classification revisited. *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, wolumen 70 of *Proceedings of Machine Learning Research*, strony 961–969. PMLR.

5.1 Rodziny reguł decyzyjnych [B_1] [B_2] [B_4] [B_5] [B_6]

Od wielu dekad reguły decyzyjne odgrywały istotną rolę w uczeniu maszynowym, z jednym z pierwszych algorytmów indukcji reguł zaproponowanym przez Ryszarda Michalskiego na początku lat 80. [Michalski, 1983]. Główną zaletą reguł decyzyjnych jest ich prostota i łatwość interpretacji, własności często zaniedbywane w głównym nurcie uczenia maszynowego, a równocześnie niezwykle istotne w wielu praktycznych zastosowaniach, w których decydent nie tylko potrzebuje trafne predykcje, ale również predykcje *łatwe do wyjaśnienia*. Reguły są również zdolne do modelowania złożonych interakcji między cechami, a tym samym są w stanie przybliżać bardzo złożone funkcje. Wczesne algorytmy indukowania reguł były

oparte przede wszystkim na metodzie *sekwencyjnego pokrywania* [Grzymala-Busse, 1992; Cohen, 1995; Fürnkranz, 1996; Stefanowski, 1998]. W wyniku rozwoju metod uczenia maszynowego wprowadzono nową technikę, bazującą na metodzie *boosting* (znaną również jako *forward stagewise additive modeling*) [Freund i inni, 2003; Hastie i inni, 2003; Schapire i Freund, 2014], w której klasyfikatory bazowe są kolejno dodawane do rodziny klasyfikatorów, przy czym nowy klasyfikator koncentruje się na przykładach, które były najtrudniejsze dla klasyfikatorów obecnych w rodzinie. Wybór reguły decyzyjnej jako klasyfikatora bazowego doprowadził do nowych, efektywnych metod indukcji reguł [Cohen i Singer, 1999].

W naszych badaniach zaproponowaliśmy podejście do generowania rodzin reguł przy użyciu strategii *boosting*. Wyprowadziliśmy ogólny algorytm nazwany *ENDER* [B_4], który unifikuje indukcję reguł dla klasyfikacji binarnej i regresji w jednym schemacie. Celem algorytmu jest stopniowa minimalizacja zadanej funkcji straty (np. straty logistycznej dla klasyfikacji lub błędu kwadratowego dla regresji), iteracyjnie dodając nowe reguły do rodziny. Powstały klasyfikator jest liniową kombinacją reguł, przy czym współczynniki liniowe mają interesującą interpretację jako siła głosu indywidualnych reguł. Rozważyliśmy szeroki zakresów technik minimalizacyjnych stosowanych w *boostingu*, które okazały się prowadzić do różnych miar czystości w procesie generowania reguł. Przedstawiliśmy teoretyczny wgląd w proces generowania reguł pokazując, że indukowane miary czystości prowadzą do przetargu między jakością klasyfikacji (dyskryminacją) a pokryciem. Pokazaliśmy, że nasz algorytm jest konkurencyjny w stosunku do innych znanych metod indukcji reguł. W pracy [B_6], rozważaliśmy algorytm ENDER w szerszym kontekście, omawiając podobieństwa i różnice między podejściem *boosting* a sekwencyjnym pokrywaniem.

Rozważaliśmy również indukcję reguł w kontekście uczenia się preferencji [Fürnkranz i Hüllermeier, 2011]. W pracy [B_5] zaproponowaliśmy dwa podejście do indukcji reguł z informacji dotyczącej porównań parami obiektów przez decydenta. Pierwsze z podejść dotyczy uczenia się binarnej relacji preferencji na parach, którą można zastosować do utworzenia (liniowego) rankingu obiektów poprzez ewaluację wszystkich par, a następnie użycie metody *Net Flow Score*. Drugie z podejść dotyczy uczenia się funkcji użyteczności, która każdemu obiektowy przypisuje wartość rzeczywistą, na podstawie których można utworzyć ranking obiektów. W obu podejściach do generowania reguł użyliśmy techniki *boosting*. Przeprowadziliśmy obszerny eksperyment na rzeczywistych danych celem porównania obu podejść.

Badaliśmy również indukcję reguł w kontekście klasyfikacji porządkowej przy założeniu ograniczeń monotonicznych ([B_2], [B_1], również: rozdział 4.3.5). Celem jest konstrukcja rodziny reguł, która (traktowana jako funkcja z przestrzeni cech do zbioru uporządkowanych indeksów klas) jest funkcją monotoniczną ze względu na wszystkie cechy. Nie jest to zadanie trywialne, ponieważ standardowe algorytmy *boostingu* nie pozwalają na łatwe narzucenie tego typu ograniczeń monotonicznych. Zaproponowaliśmy więc dwufazową procedurę klasyfikacji, składającą się z fazy *monotonizacji danych* i fazy indukcji reguł. W pierwszej fazie etykiety klas są zastąpione nowymi etykietami, które są monotoniczne względem cech, a procedura stara się tak dobrać nowe etykiety aby zminimalizować liczbę przeetykietowań. W drugiej fazie reguły indukowane są za pomocą techniki *boosting*. Pokazaliśmy, że kontrolowanie znaków współczynników liniowych reguł jest wystarczające do zapewnienia monotoniczności. Użyliśmy dwóch procedur indukcji reguł: minimalizacji sigmoidalnej funkcji straty ze stałym krokiem ([B_1], algorytm *MORE*) oraz metody *linear programming boosting* ([B_2], algorytm *LPRules*). W obu przypadkach dokonaliśmy teoretycznego wglądu w analizowany problem, jak również zaprezentowaliśmy wyniki eksperymentów obliczeniowych wskazujące, że otrzymane algorytmy są konkurencyjne z najlepszymi istniejącymi algorytmami dla tego typu problemu.

5.2 Kwantowe uczenie się [B_3]

Uogólniliśmy problem klasyfikacji binarnej na problem klasyfikacji dwóch nieznanych *stanów kwantowych*. Główny paradygmat kwantowej teorii informacji mówi, że systemy kwantowe są nośnikiem nowego rodzaju informacji z potencjalnymi rewolucyjnymi zastosowaniami, jak np. szybsze obliczenia czy bezpieczna łączność [Nielsen i Chuang, 2000]. Rozwój technologii kwantowych wymaga opracowania nowych narzędzia do sterowania i precyzyjnego pomiaru układów kwantowych, w których walidacja statystyczna stała się częścią standardowej procedury eksperymentalnej [Paris i Rehacek, 2004]. Przedstawiliśmy nowy typ kwantowego problemu statystycznego zainspirowanego teorią uczenia się, a mianowicie problem klasyfikacji stanu kwantowego. Mając „zbiór uczący" składający się n_1 identycznych, nieznanych stanów kwantowych (kubitów) ρ (kubity mogą być reprezentowane przez dodatnio określoną macierz

zespoloną o rozmiarze 2×2 z jednostkowym śladem, zwaną *macierzą gęstości*) i n_2 nieznanych kubitów σ, celem jest opracowanie pomiaru, który jak najlepiej rozróżnia przyszłe kopie ρ i σ. Jeśli ρ, σ i ich prawdopodobieństwa a priori są znane, optymalna strategia rozróżniania między stanami nazywana jest pomiarem Helstroma [Helstrom, 1976] i gra tutaj rolę klasyfikatora bayesowskiego w klasycznej statystycznej teorii uczenia się. W naszym problemie stany ρ i σ są nieznane, musimy się więc nauczyć rozróżniającego je pomiaru. Rozważyliśmy tu dwie metody: (1) Strategię *plug-in*, która wpierw estymuje nieznane stany ρ i σ, a następnie wykonuje pomiar Helstroma na podstawie estymatorów; (2) Strategię *bezpośrednią*, która wykonuje łączny pomiar dla danych uczących i nowej, nieznanej kopii celem podjęcia decyzji. Wykazaliśmy, że ta ostatnia strategia prowadzi do optymalnej metody klasyfikacji i zazwyczaj działa ściśle lepiej od strategii *plug-in*. Naturalną miarą jakości jest tu nadwyżka ryzyka, różnica między oczekiwanym błędem zadanej metody a błędem optymalnego pomiaru Helstroma. Udowodniliśmy, że nadwyżka ryzyka każdej z metod jest rzędu n^{-1} (gdzie $n = n_1 + n_2$), obliczając dokładne stałe. Ponieważ optymalny pomiar klasyfikacji (metoda bezpośrednia) różni się od optymalnej metody estymacji stanów (metoda *plug-in*), nasze wyniki ponownie pokazują, że różne kwantowe problemy decyzyjne nie mogą być jednocześnie rozwiązane optymalnie.

5.3 Klasyfikacja wieloetykietowa [B_{10}] [B_{13}]

W klasyfikacji wieloetykietowej jeden obiekt może być przydzielony do żadnej, jednej lub wielu klas naraz [Gibaja i Ventura, 2015; Dembczyński i inni, 2012]. Traktując przydział do każdej klasy jako binarną etykietę (równą 1 gdy obiekt należy do danej klasy i 0 w przeciwnym przypadku) widzimy, że klasyfikacja wieloetykietowa uogólnia predykcję pojedynczej etykiety (jak w klasyfikacji binarnej) na predykcję wektora etykiet. Problemy takie często występują w praktycznych zastosowaniach, np. w systemach rekomendacyjnch (film lub książka przypisane do więcej niż jednej kategorii), klasyfikacji tekstu (dokumenty oznaczone więcej niż jedną etykietą) lub nawet w bioinformatyce (geny związane z więcej niż jedną klasą funkcjonalną). Z powodu wektorowej natury wyjść, problemy wieloetykietowe doczekały się wielu nowych miar jakości klasyfikacji, takich jak błąd Hamminga, błąd 0/1 na podzbiorach, błąd rangowy czy miara F [Dembczyński i inni, 2012].

W pracy [B_{10}] porównaliśmy dwa podejścia do optymalizacji wieloetykietowej miary F. Pierwsze z nich nazywa się *structured loss minimization* [Petterson i Caetano, 2010, 2011] i wywodzi się ze strukturalnych maszyn wektorów podpierających (*structured support vector machines*, SSVM). Drugie podejście jest metodą typu *plug-in* [Lewis, 1995; Nan i inni, 2012; Dembczyński i inni, 2011], w której wpierw uczymy się modelować prawdopodobieństwa warunkowe klas, a następnie używamy tych prawdopodobieństw w fazie wnioskowania do optymalnej predykcji (przydziału etykiet). Faza wnioskowania została oparta na algorytmie Dembczyńskiego i innych [2011] dokładnej optymalizacji miary F przy użyciu $m^2 + 1$ parametrów łącznego rozkładu warunkowego etykiet (gdzie m jest liczbą etykiet). Rozważaliśmy również prostszy wariant zakładający niezależność etykiet, używając algorytmu wnioskowania z pracy Nan i inni [2012]. Pokazaliśmy w eksperymencie obliczeniowym, że podejście *plug-in* jest skuteczniejsze od podejścia SSVM. Analizowaliśmy również oba typy metod wykazując, że o ile podejście *plug-in* jest spójne statystycznie, podejście SSVM nie posiada tej własności.

W pracy [B_{13}] rozważaliśmy *drzewa klasyfikatorów* dla klasyfikacji wieloetykietowej i wieloklasowej. Metody te reprezentują model predykcyjny w formie drzewa, w węzłach którego znajdują się proste, binarne klasyfikatory określające do którego poddrzewa należy dana obserwacja. Predykcje dokonywane są poprzez (zachłanne) przejście od korzenia do liści drzewa, określających przydział konkretnych etykiet. Podejście to jest bardzo użyteczne, ponieważ potencjalnie pozwala zmniejszyć czas predykcji z $O(m)$ do $O(\log m)$, gdzie m jest liczbą wszystkich możliwych wyjść (liczba klas w klasyfikacji wieloklasowej lub liczba możliwych kombinacji etykiet w klasyfikacji wieloetykietowej). Skupiliśmy się na klasie metod zwanej *probabilistic classifier trees* (PCTs). Dzięki uczeniu klasyfikatorów probabilistycznych w wewnętrznych węzłach drzewa, metody te pozwalają na przeglądanie drzewa w bardziej zaawansowany i mniej zachłanny sposób, potencjalnie dając lepsze predykcje. Naszym głównym wynikiem jest ograniczenie na żal ze względu na błąd 0/1 poprzez błąd algorytmu przeszukiwania i dywergencję Kullbacka-Leiblera (żal ze względu na błąd logistyczny) klasyfikatorów w wewnętrznych węzłach drzewa. Ograniczenie to implikuje statystyczną spójność metody, a zarazem określa przetarg między złożonością obliczeniową a trafnością predykcji. Porównaliśmy również w eksperymencie obliczeniowym algorytmy PCTs z inną

metodą opartą na drzewach klasyfikatorów, *Filter Trees*.

5.4 AUC i błąd rangowy [B_7] [B_8]

Ranking dwudzielny dotyczy problemów rangowania ze zbioru treningowego pozytywnych i negatywnych przykładów. Najbardziej popularnym kryterium jakości jest tu *pole pod krzywą ROC* (*area under the ROC curve*, AUC) [Hanley i McNeil, 1982; Cortes i Mohri, 2003], które określa prawdopodobieństwo, że w losowo wybranej parze pozytywnych i negatywnych obiektów obiekt pozytywny rangowany jest (poprawnie) powyżej obiektu negatywnego. AUC jest funkcją użyteczności („im wyższa, tym lepsza"), a odpowiadającą jej funkcją straty jest *błąd rangowy* (równy $1-$ AUC).

Ponieważ błąd rangowy zdefiniowany jest na parach, większość metod rangowania również używa podejścia na parach, efektywnie redukując problem do klasyfikacji binarnej i traktując każdą parę (x, x') jako pojedynczą obserwację, klasyfikowaną jako pozytywna wtedy i tylko wtedy gdy x poprzedza x' w rankingu [Herbrich i inni, 1999; Freund i inni, 2003; Agarwal i inni, 2005]. Niestety, podejście to skaluje się w ogólności kwadratowo z rozmiarem danych (w niektórych przypadkach możliwe jest obniżenie tej złożoności, wymaga to jednak dedykowanych, bardziej złożonych algorytmów optymalizacji). Wydaje się więc uzasadnione zapytać, czy ta dodatkowa złożoność jest tu rzeczywiście potrzebna, szczególnie zważywszy, że w kilku eksperymentach wykazano, iż proste klasyfikatory oparte na funkcjach rzeczywistych uzyskują bardzo dobre wartości AUC. Klasyfikatory takie mogą zostać użyte do rankingu po prostu sortując obserwacje po przydzielonych im przez klasyfikator rzeczywistych wartościach wyjścia. W pracy [B_7] naszym celem była więc odpowiedź na pytanie, jak wiele możemy osiągnąć w problemie rankingu poprzez uczenie prostych klasyfikatorów minimalizujących funkcje straty na pojedynczych obserwacjach. Wpierw pokazaliśmy, że minimalizacja błędu 0/1 nie jest dobrym pomysłem, ponieważ może prowadzić do dowolnie dużego błędu rangowego. Dalej wykazaliśmy, że lepsze wyniki można uzyskać stosując ważoną wersję błędu 0/1. Największy zysk osiągnęliśmy jednak poprzez minimalizację marginesowych funkcji strat, takich jak eksponencjalna lub logistyczna funkcja straty, dla których byliśmy w stanie pokazać ograniczenia na *żal rangowy*. Nasze wyniki potwierdziliśmy w eksperymencie obliczeniowym.

W pracy [B_8] rozszerzyliśmy nasze wyniki na problem minimalizacji błędu rangowego w klasyfikacji wieloetykietowej, podając ograniczenia na żal rangowy dla prostych klasyfikatorów minimalizujących zastępcze funkcje straty na pojedynczych etykietach, takie jak eksponencjalna lub logistyczna funkcja straty. Poprzednio problem ten był rozwiązywany za pomocą zastępczych wypukłych strat zdefiniowanych na parach etykiet. Podejście to jest jednak nie tylko bardziej złożone obliczeniowo, ale zostało poddane w wątpliwość w świetle negatywnych wyników pokazujących, że typowe funkcje zastępcze na parach nie są nawet statystycznie spójne [Duchi i inni, 2010; Gao i Zhou, 2011]. Co cieawe, nasze funkcje zastępcze na pojedynczych etykietach są nie tylko prostsze obliczeniowo i koncepcyjnie, ale również okazują się statystycznie spójne. Nasze wyniki prowadziły do efektywnych algorytmów dla rankingu wieloetykietowego, które przetestowaliśmy w eksperymencie obliczeniowym.

5.5 Niedekomponowalne miary złożoności w klasyfikacji binarnej [B_{12}] [B_{14}]

Jakość klasyfikatorów mierzona jest tradycyjnie trafnością klasyfikacji. W wielu przypadkach miara ta jest jednak nieadekwatna do rozważanych problemów i liczne zastosowania doprowadziły do propozycji wielu bardziej złożonych miar jakości [Choi i Cha, 2010], takich jak AUC dla niezrównoważonych klas [Menon i inni, 2013], miara F dla wyszukiwania informacji [Lewis, 1995], lub precyzja w czołówce rankingu [Kar i inni, 2014]. Dużym wyzwaniem w optymalizacji takich miar jest jednak ich *niedekomponowalność* – ewaluacja zbioru predykcji nie może zostać zdekomponowana na sumę ewaluacji pojedynczych predykcji. Ta cecha stanowi główną trudność w analizie teoretycznej tych miar i doprowadziła do dwóch różnych podejść do badania spójności statystycznej [Nan i inni, 2012].

Z jednej strony, podejście *Population Utility* (PU) skupia się na *estymacji* – czyli PU-spójny klasyfikator to taki, który poprawnie estymuje wartość miary na całej populacji. Z drugiej strony, podejście *Expected Test Utility* (ETU) skupia się na *uogólnieniu* – czyli ETU-spójny klasyfikator optymalizuje oczekiwaną wartość błędu względem losowo wybieranych zbiorów danych o zadanym rozmiarze. W pracy [B_{14}] zaprezentowaliśmy szereg wyników porównujących podejścia PU i ETU. Pokazaliśmy, że dla dużego zakresu miar, podejścia PU i ETU są *asymptotycznie równoważne* (tj. dla zbiorów danych o dużym rozmiarze),

przy założeniu technicznego warunku p-*lipszycowskości*, spełnianego przez większość miary używanych w praktyce. Podobny rezultat znany był poprzednio tylko dla miary F [Nan i inni, 2012]. Analizowaliśmy też przybliżoną klasyfikację ETU używając rozwinięcia miary w szereg Taylora i pokazując, że przybliżenia mogą być wyznaczone znacznie efektywniej i są równocześnie obarczone małym błędem przy pewnych standardowych założeniach. Rozważaliśmy również efekt błędnej specyfikacji modelu, odkrywając, że podejście ETU jest tu bardziej wrażliwe niż PU i może wymagać dobrej kalibracji prawdopodobieństw warunkowych. Na koniec zaprezentowaliśmy wyniki obliczeniowe na symulowanych i rzeczywistych danych potwierdzające nasze wyniki teoretyczne.

W pracy [B_{12}] rozważyliśmy problem uczenia się klasyfikatora binarnego optymalizującego miarę *approximate median significance* (AMS), której optymalizacja była celem konkursu uczenia maszynowego *Higgs Boson Machine Learning Challenge* (HiggsML) [Adam-Bourdarios i inni, 2014]. Skoncentrowaliśmy się na najbardziej popularnym podejściu do optymalizacji AMS, opartym na uczeniu się funkcji o wartościach rzeczywistych f poprzez minimalizację na próbce uczącej zastępczej funkcji straty dla klasyfikacji binarnej, takiej jak funkcja logistyczna, a następnie utworzeniu klasyfikatora poprzez progowanie f, tzn. wartości funkcji powyżej progu oznaczały klasyfikację pozytywną (klasa *event*), a poniżej – klasyfikację negatywną (klasa *background*). Próg był dobierany na osobnym zbiorze walidacyjnym bezpośrednio optymalizując miarę AMS. Takie podejście stało się bardzo popularne wśród uczestników konkursu głównie ze względu na fakt, że pierwszy etap (uczenie się funkcji f) nie wykorzystuje docelowej miary oceny (AMS) i dzięki temu można bez żadnych modyfikacji stosować standardowe metody klasyfikacji, takie jak regresja logistyczna, *Stochastic Gradient Boosting*, *Random Forest* [Hastie i inni, 2003], itp. Mimo swojej prostoty, procedura ta okazała się bardzo skuteczna. W naszej pracy wykazaliśmy, że takie podejście stanowi spójną metodę optymalizacji AMS. Mówiąc ściślej, wykazaliśmy, że żal wynikowego klasyfikatora (otrzymanego przez progowanie f) mierzony względem kwadratu błędu AMS jest ograniczony przez żal funkcji f mierzony względem logistycznej funkcji straty. Było to pierwsze ograniczenie na żal zastosowane do niedekomponowalnej miary jakości takiej jak AMS.

5.6 Estymacja czasów przejazdu w rzeczywistej sieci drogowej [B_9] [B_{11}]

Problem ten był badany we współpracy z firmą *NaviExpert* zajmującą się nawigacją w ruchu drogowym.

W pracy [B_9] opisaliśmy szereg metod statystycznych używanych w systemie *Community Traffic* (NX-CT) firmy *NaviExpert*. System ten zbiera surowe dane dotyczące czasu i pozycji geograficznej urządzeń GPS (a tym samym samochodów, w których się one znajdują), które są później przeliczane na czasy przejazdu w segmentach sieci drogowej, co pozwala na wnioskowanie o płynności ruchu. Zaproponowaliśmy metodę estymacji czasów przejazdu opartą na modelu składającym się z dwóch komponentów. Komponent *statystyczny* jest odpowiedzialny za predykcje oparte na stabilnych i powtarzalnych trendach (np. „w każdą niedzielę rano ruch na ulicach w centrum jest niewielki"). Stabilność ta stanowi o sile komponentu (pozwala na dalekosiężne predykcji, np. rzędu dni), ale również o jego słabości (brak możliwości reakcji na nagłe zmiany w płynności ruchu). Dlatego komponent *dynamiczny* dostraja predykcje komponentu statystycznego na podstawie ostatnich obserwacji reagując na zmiany, które nie mogą być wyjaśnione długo-okresowym zachowaniem (np. roboty drogowe lub nietypowe zagęszczenie ruchu). System NX-CT umożliwia również użytkownikom aktywny udział w procesie kolekcjonowania danych poprzez wysyłanie informacji o aktualnej sytuacji drogowej. Zaproponowaliśmy metodę estymacji wiarygodności takich wiadomości opartą na algorytmie *Expectation Maximization*. Nasze algorytmy zostały przetestowane w eksperymencie obliczeniowym na rzeczywistych danych pochodzących z systemu NX-CT.

W pracy [B_{11}] użyliśmy metod dekompozycji macierzy (*matrix factorization*) popularnych w systemach rekomendacyjnych [Srebro i inni, 2005; Koren i inni, 2009] do estymacji czasów przejazdu z danych historycznych. Rozważaliśmy dużą macierz czasów przejazdu, w której wiersze odpowiadały krótkim, niepodzielnym odcinkom drogowym, a kolumny – krótkim (15-minutowym) przedziałom czasowym rozpiętym na przestrzeni całego tygodnia. Dekompozycja takiej macierzy daje zbiór ukrytych cech w formie dwóch macierzy o małym rzędzie, których iloczyny przybliżają oryginalną macierz. Model taki pozwala trzymać w pamięci dwie macierze znacznie mniejsze od całej macierzy danych, a dodatkowo może służyć jako efektywna obliczeniowo i trafna metoda predykcji czasów przejazdu na dowolnym odcinku sieci drogowej, w dowolnej chwili czasowej. Przy okazji, ukryte cechy wyznaczone przez dekompozycję macierzy dały interesujący wgląd do analizowanego problemu. Eksperymenty na dużych, rzeczywistych zbiorach

danych wykazały lepsze wyniki zaproponowanego algorytmu w porównaniu do kilku standardowych modeli szacowania czasów przejazdu.

Literatura

Abernethy, J., Bach, F., Evgeniou, T., i Vert, J.-P. (2009). A new approach to Collaborative Filtering: Operator estimation with spectral regularization. *Journal of Machine Learning*, 10:803–826.

Adam-Bourdarios, C., Cowan, G., Germain, C., Guyon, I., Kégl, B., i Rousseau, D. (2014). Learning to discover: the Higgs boson machine learning challenge. `http://higgsml.lal.in2p3.fr/documentation/`.

Agarwal, S. (2014). Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653-1674.

Agarwal, S., Graepel, T., Herbrich, R., Har-Pelend, S., i Roth, D. (2005). Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425.

Argyriou, A., Micchelli, C. A., i Pontil, M. (2009). When is there a Representer Theorem? Vector versus matrix regularizers. *Journal of Machine Learning Research*, 10:2507–2529.

Arora, R., Cotter, A., i Srebro, N. (2013). Stochastic optimization of PCA with capped MSG. *Advances in Neural Information Processing Systems 26 (NIPS)*, strony 1815–1823.

Ayer, M., Brunk, H. D., Ewing, G. M., Reid, W. T., i Silverman, E. (1955). An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 26(4):641-647.

Azoury, K. i Warmuth, M. (2001). Relative loss bounds for on-line density estimation with the exponential family of distributions. *Journal of Machine Learning*, 43(3):211–246.

Balcan, M.-F., Bansal, N., Beygelzimer, A., Coppersmith, D., Langford, J., i Sorkin, G. B. (2008). Robust reductions from ranking to classification. *Machine Learning*, 72(1):139–153.

Barndorff-Nielsen, O. (1978). *Information and Exponential Families in Statistical Theory*. Wiley, Chichester, UK.

Bennett, J. i Lanning, S. (2007). The Netflix Prize. *Proceedings of KDD Cup and Workshop 2007*.

Boser, B. E., Guyon, I. M., i Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings 5th Annual ACM Workshop on Comput. Learning Theory*, strony 144–152. ACM Press, New York, NY.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., i Warmuth, M. K. (1997). How to use expert advice. *Journal of the ACM*, 44(3):427–485.

Cesa-Bianchi, N. i Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press.

Chandrasekaran, R., Ryu, Y. U., Jacob, V. S., i Hong, S. (2005). Isotonic separation. *INFORMS Journal on Computing*, 17(4):462–474.

Choi, S.-S. i Cha, S.-H. (2010). A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, strony 43–48.

Cohen, W. W. (1995). Fast Effective Rule Induction. *Proceedings of the 12th International Conference of Machine Learning (ICML 1995)*, strony 115–123, Tahoe City, USA. Morgan Kaufmann.

Cohen, W. W. i Singer, Y. (1999). A simple, fast, and effective rule learner. *Proceedings of the 16th National Conference on Artificial Intelligence*, strony 335–342, Orlando, USA. AAAI Press / The MIT Press.

Cortes, C. i Mohri, M. (2003). AUC optimization vs. error rate minimization. *Advances in Neural Information Processing Systems 16 (NIPS)*. MIT Press.

Cover, T. M. i Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley & Sons.

Dasgupta, S. i Hsu, D. (2007). On-line estimation with the multivariate gaussian distribution. *Proceedings of the 20th Annual Conference on Learning Theory (COLT 2007)*, strony 278–292.

Dashevskiy, M. i Luo, Z. (2011). Time series prediction with performance guarantee. *IET Communications*, 5(8):1044–1051.

Dembczyński, K., Waegeman, W., Cheng, W., i Hüllermeier, E. (2011). An exact algorithm for F-measure maximization. *Advances in Neural Information Processing Systems 24 (NIPS)*, strony 1404-1412. Curran Associates, Inc.

Dembczyński, K., Waegeman, W., Cheng, W., i Hüllermeier, E. (2012). On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5-45.

Devroye, L., Györfi, L., i Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1st edition.

Duchi, J., Mackey, L., i Jordan, M. (2010). On the consistency of ranking algorithms. *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, strony 327–334.

Freund, Y. (1996). Predicting a binary sequence almost as well as the optimal biased coin. *Proceedings of the 9th Annual Conference on Learning Theory (COLT 1996)*, strony 89–98.

Freund, Y., Iyer, R., Schapire, R. E., i Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933-969.

Freund, Y. i Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139.

Fürnkranz, J. (1996). Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54.

Fürnkranz, J. i Hüllermeier, E., redaktorzy (2011). *Preference Learning*. Springer-Verlag.

Gao, W. i Zhou, Z. (2011). On the consistency of multi-label learning. *Proceedings of the 24th Annual Conference*

on Learning Theory (COLT 2011), Journal of Machine Learning Research Workshop and Conference Proceedings, strony 341–358.
Gibaja, E. i Ventura, S. (2015). A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3):52:1–52:38.
Greco, S., Matarazzo, B., i Słowiński, R. (2001). Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, 129:1–47.
Grünwald, P. D. (2007). *The Minimum Description Length Principle*. MIT Press.
Grünwald, P. D. i de Rooij, S. (2005). Asymptotic log-loss of prequential maximum likelihood codes. *Proceedings of the 18th Annual Conference on Learning Theory (COLT 2005)*, strony 652–667.
Grzymala-Busse, J. W. (1992). LERS — A system for learning from examples based on rough sets. Słowiński, R., redaktor, *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*, strony 3–18. Kluwer Academic Publishers.
Hanley, J. A. i McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.
Hannan, J. (1957). Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139.
Harremoës, P. (2013). Extendable MDL. *The IEEE International Symposium on Information Theory (ISIT 2013)*, strony 1516–1520. IEEE.
Hastie, T., Tibshirani, R., i Friedman, J. (2003). *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag.
Hedayati, F. i Bartlett, P. L. (2012a). Exchangeability characterizes optimality of sequential normalized maximum likelihood and bayesian prediction with jeffreys prior. *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, wolumen 22 *Journal of Machine Learning Research: Workshop and Conference Proceedings*, strony 504–510.
Hedayati, F. i Bartlett, P. L. (2012b). The optimality of jeffreys prior for online density estimation and the asymptotic normality of maximum likelihood estimators. *Proceedings of the 25th Annual Conference on Learning Theory (COLT 2012)*, wolumen 23 *Journal of Machine Learning Research: Workshop and Conference Proceedings*, strony 7.1–7.13.
Helstrom, C. W. (1976). *Quantum Detection and Estimation Theory*.
Herbrich, R., Graepel, T., i Obermayer, K. (1999). Regression models for ordinal data: A machine learning approach. Technical report TR-99/03, Technical University of Berlin.
Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., i Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.
Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 498–520.
Kalai, A. i Vempala, S. (2005). Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307.
Kar, P., Narasimhan, H., i Jain, P. (2014). Online and stochastic gradient methods for non-decomposable loss functions. *Advances in Neural Information Processing Systems 27 (NIPS)*, strony 694–702.
Kimeldorf, G. S. i Wahba, G. (1971). Some results on Tchebycheffian Spline Functions. *Journal of Mathematical Analysis and Applications*, 33:82-95.
Kolmogorov, A. N. (1965). Three approaches to the definition of the concept "quantity of information.". *Problems of Information Transmission*, 1:3–11.
Koren, Y., Bell, R., i Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.
Koyejo, O., Natarajan, N., Ravikumar, P. K., i Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics. *Advances in Neural Information Processing Systems 27 (NIPS)*, strony 2744–2752.
Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.
Langford, J., Zhang, T., Hsu, D. J., i Kakade, S. M. (2009). Multi-label prediction via compressed sensing. *Advances in Neural Information Processing Systems 22 (NIPS)*, strony 772–780. MIT Press.
Levy, H. (1998). *Stochastic Dominance*. Kluwer Academic Publishers.
Lewis, D. (1995). Evaluating and optimizing autonomous text classification systems. *The 18th International ACM SIGIR Conference on Research and Development in Information Retrieval*, strony 246-254.
Littlestone, N. i Warmuth, M. K. (1994). The Weighted Majority algorithm. *Information and Computation*, 108(2):212–261.
Luo, H., Agarwal, A., Cesa-Bianchi, N., i Langford, J. (2016). Efficient second order online learning by sketching. *Advances in Neural Information Processing Systems 29 (NIPS)*, strony 902–910.
Manning, C. D., Raghavan, P., i Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
McMahan, H. B. i Orabona, F. (2014). Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximation. *Proc. of the 27th Conference on Learning Theory (COLT 2014)*, strony 1020–1039.
Menon, A. K., Narasimhan, H., Agarwal, S., i Chawla, S. (2013). On the statistical consistency of algorithms for binary classification under class imbalance. *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, strony 603–611.

Michalski, R. S. (1983). A theory and methodology of inductive learning. Michalski, R. S., Carbonell, J. G., i Mitchell, T. M., redaktorzy, *Machine Learning: An Artificial Intelligence Approach*, strony 83–129. Tioga Publishing, Palo Alto.

Nan, Y., Chai, K. M. A., Lee, W. S., i Chieu, H. L. (2012). Optimizing F-measure: A tale of two approaches. *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, strony 289–296.

Nielsen, M. A. i Chuang, I. L. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press.

Orabona, F. i Pál, D. (2016). Coin betting and parameter-free online learning. *Advances in Neural Information Processing Systems 29 (NIPS)*, strony 577–585.

Paris, M. i Rehacek, J., redaktorzy (2004). *Quantum State Estimation*, wolumen 649 *Lecture Notes in Physics*. Springer-Verlag.

Petterson, J. i Caetano, T. S. (2010). Reverse multi-label learning. *Advances in Neural Information Processing Systems 24 (NIPS)*, strony 1912-1920.

Petterson, J. i Caetano, T. S. (2011). Submodular multi-label learning. *Advances in Neural Information Processing Systems 24 (NIPS)*, strony 1512-1520.

Rissanen, J. (1984). Universal coding, information, prediction and estimation. *IEEE Transactions on Information Theory*, 30:629-636.

Rissanen, J. (1996). Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, IT-42(1):40-47.

Rissanen, J. i Roos, T. (2007). Conditional NML universal models. *Information Theory and Applications Workshop (ITA-07)*, strony 337–341.

Robertson, T., Wright, F. T., i Dykstra, R. L. (1998). *Order Restricted Statistical Inference*. John Wiley & Sons.

Schapire, R. E. i Freund, Y. (2014). *Boosting: Foundations and Algorithms*. MIT Press.

Shalev-Shwartz, S. (2012). Online learning and online convex optimization. *Foundation and Trends in Machine Learning*, 4(2):107–194.

Shtarkov, Y. M. (1987). Universal sequential coding of single messages. *Problems of Information Transmission*, 23(3):3–17.

Solomonoff, R. (1964). A formal theory of inductive inference, part I. *Information and Control*, 7:1–22.

Srebro, N., Rennie, J. D. M., i Jaakola, T. S. (2005). Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems 18 (NIPS)*, strony 1329-1336. MIT Press.

Stefanowski, J. (1998). On rough set based approach to induction of decision rules. Skowron, A. i Polkowski, L., redaktorzy, *Rough Set in Knowledge Discovering*, strony 500–529. Physica Verlag, Heidelberg.

Streeter, M. i McMahan, H. B. (2012). No-regret algorithms for unconstrained online convex optimization. *Advances in Neural Information Processing Systems 25 (NIPS)*, strony 2402–2410.

Takimoto, E. i Warmuth, M. (2000). The last-step minimax algorithm. *Proceedings of the 13th Annual Conference on Computational Learning Theory (COLT 2000)*, strony 100–106.

Tsuda, K., Rätsch, G., i Warmuth, M. K. (2005). Matrix Exponentiated Gradient updates for on-line learning and Bregman projections. *Journal of Machine Learning Research*, 6:995–1018.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17:395-416.

V'yugin, V. V. (2015). The following the perturbed leader algorithm and its application for constructing game strategies.

Warmuth, M. K. i Kuzmin, D. (2008). Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9:2287-2320.

Zadrozny, B. i Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002)*, strony 694–699.

Zamani, M., Beigy, H., i Shaban, A. (2016). Cascading randomized weighted majority: A new online ensemble learning algorithm. *Intelligent Data Analysis*, 20(4):877–889.

Wojciech Kotłowski