УНИВЕРЗИТЕТ У БЕОГРАДУ

ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

ЗАВРШНИ РАД

Јаков Петровић

Београд 2020.

# УВОД

Рад се бави истраживањем техника за налажење најближих суседа над већим скуповима података. Претрага најближег суседа у метричком просотру подразумева проналажење ентитета који има најмању удаљеност од ентитета у односу на којег се претрага врши.

Претрага најближих суседа преставља значајан проблем у областима попут биомедицине, генетике, маркетинга, социјалних медија, машинског учења и оптимизације, при обради слика и снимака, препознавању образаца и облика те анализи и обради просторних и геолошких података. У већини случајева, записи из ових база се могу представити у векторском простору који има неколико десетина па чак и хиљаду димензија. (Indyk & Motwani, 1998)

Алгоритам к- најближох суседа (енг. *k-nearest neighbours*) је предиктивни алгоритам машинског учења, који непознату класу или вредност објекта одређује на основу особина из скупа најсличнијих познатих инстанци и широко је коришћен метод класификације и регресије. Претрага суседа је кључна и временски најкритичнија фаза алгоритма. Из претходно наведеног, намеће се закључак да су класификација и регресија само мали наставак на сложенији проблем претраге најближих суседа (енг. *Nearest neighbour search*).

Предвиђање будућих догађаја је један од кључних изазова са којима се суочавају друштва, предузећа и појединци. У складу са предвиђањима, ентитети могу предузети акције у циљу повећања прихода или смањења штете.

Хоће ли ће нови комитент банке бити у могућности да отплати кредит? Хоће ли ће пацијент имати проблем при уградњи зубног импланта? Који жанр музике ће бити предложен потенцијалном купцу?

Добијање благовремених резултата применом алгоритама машинског учења, уз ефикасно коришћење ресурса, је императив. КНН је један од најједноставнијих метода за класификацију и регресију. Једноставност је врлина КНН алгоритма, али је и узрок слабијих перформанси код података са шумовима, или кад је потребно обрадити велики број димензија и инстанци.

и унапређење у погледу редукције времена и ефикаснијег коришћења ресурса који су неопходни за спровођење алгоритма.

Многобројне иновативне технике проналажења суседа који могу, али не морају бити најближи суседи непознате инстанце, настале су залагањем истраживача из разних научних области. Најзначајнија особина апроксимативног алгоритма је да постигне готово исту прецизност као оригинални приступ са потпуним скенирањем, али за значајно краћи временски интервал. Управо је та чињеница мотивисала су аутора рада на изучавање апроксимативних метода. Резултат истраживања ће приказати рад алгоритма, преглед и анализу његових апроксимативних варијанти. Утврдиће се разлика у перформансама апроксимативних техника над скуповима података различитих величина и димензија, на корист будућим истраживачима.

Ovde ide Reminder, tj- šta je u kom poglavlju.

# Дефиниција проблема и преглед стања у области

Алгоритам к- најближих суседа (КНН) je предиктивни алгоритам машинског учења који непознату инстанцу класификује на основу најсличнијих познатих података. У формалном облику, проблем налажења најближег суседа се састоји у следећем:

Нека је дат скуп тачака у простору . Потребно је претражити скуп , како би се нашле тачке које су најближе задатој тачки то јест:

За разлику од алгоритама који испрва изводе генерализацију података и генеришу класификациони модел, те на основу модела брзо процењују припадност непознате инстанце одређеној класи (енг. *eager learners*), овај метод припада групи лењих алгоритама (енг. *lazy learners*). Одлика лењих алгоритама је да се инстанце просто складиште у фази тренирања или се извршава минимално претпроцесирање података. То значи да ће фаза тренирања алгоритма бити краћа, међутим предвиђање алгоритма ће трајати дуже. Како се фаза тренирања код ових алгоритама састоји у простом чувању инстанци, ова група алгоритама се назива и инстанцно-оријентисаном. Овај алгоритам je често веома рачунски интензиван и захтева ефикасне технике складиштења података, које треба да омогуће паралелно извршавање поступка проналажења суседа (Han, Pei, & Kamber, 2012).

Иако је алгоритам најближих суседа постигао одличне резултате у примени, постоје два изразита проблема у вези са скалабилношћу при раду са великом количином података:

* Време извршавања: Алгоритамска сложеност проналажења само једног најближег суседа је , где је број тачака у простору и број димензија. Алгоритам је рачунски захтевнији када је потребно пронаћи више суседа, што је најчешћи случај. Најзад, цео процес је потребно извести сваки пут када се појави инстанца коју треба класификовати.
* Меморија: За брзо израчунавање удаљености, неопходно је да је цео скуп постојећих тачака учитан у РАМ меморију, што може представљати изазов код великих база података. (Maillo, Ramirez, Triguero, & Herrera, 2017)

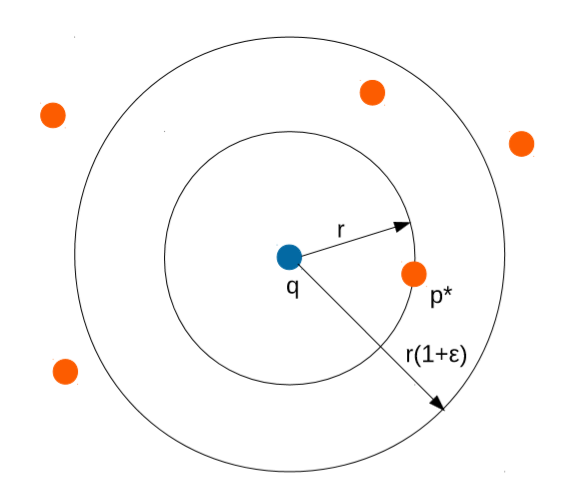
Ефикасно претраживање најближих суседа постаје учестали захтев према великом броју база података. Алгоритми који се директно ослањају на индексне структуре дају задовољавајуће резултате на малим или средњим количинама података, али не задовољавају потребе ефикасности када је реч o великим и високо димензионираним скуповима (Seidl & Kriegel, 1998).

Очигледна тешкоћа да се осмисле алгоритми засновани на претрази суседа који ће бити ефикасни у случају када је број димензја већи (код неких аутора када је број димензија већи од 10), наводи да је потребно размотрити апроксимативне методе које ће проналазити приближно најближе суседе.

У том случају, можемо проширити претходно дату дефиницију:

Нека је дат скуп тачака у простору и нова тачка којој треба одредити суседа. За дато , тачка је приближан сусед ако је:

Где је најближи сусед задате тачке . У општем случају, када je , поступак се своди на проналажење приближних најближих суседа, где за сваког - тог важи да је апроксимација одговарајућег најближег суседа тачке (Arya, Mount, Netanyahu, Silverman, & Wu, 1998).



Илустрација 1: Апроксимативни алгоритам

Проблем налажења суседа био је у пољу интересовања истраживача од педесетих година двадесетог века, међутим, област долази у фокус почетком деведесетих година када је објављено мноштво значајних радова из ове области. Према наводу (Kushilevitz, Ostrovsky, & Rabani, 2000), када је реч о веома димензионираним подацима, проблем је разматран од стране Добкина и Липтона (*Dobkin, D., & Lipton, R. J. (1976). Multidimensional searching problems*) који су генерисали алгоритам чија сложеност претраге експоненцијално зависи од броја димензија . Представљени метод је касније побољшан, али време извршења je још увек било у експоненцијалној зависности од димензије простора.

Крајем века, (Kleinberg, 1997) је представио два алгоритма који дају значајна побољшања у односу на дотадашња достигнућа. Први алгоритам нуди сложеност претраге од и други, чија сложеност тежи линеарној (Kleinberg, 1997).

Другачији приступ решавању проблема донели су (Indyk & Motwani, 1998) и објавили метод који има полиномну сложеност у зависности од и , дакле .

Већина до данас представљених решења подразумева претпроцесирање постојећих инстанци, како би се над њима касније извршила претрага. Читав сет апроксимативних метода предвиђа преуређење скупа података тако што би се формирала структура стабла (*R-tree, X-tree, SR-tree, k-d tree* и друга), чиме се обезбеђује убрзање алгоритма. Друге методе предлажу хеширање инстанци како би се спровело груписање међусобно сличних инстанци. При одабиру апроксимативног алгоритма, пожељно је размотрити време, сложеност, као и меморијске захтеве обраде података. Велики број аутора даје одличан теоријски допринос, међутим, недостатак многих ранијих радова огледа се у изостанку свеобухватне експерименталне анализе. Често се практична примена изводи на вештачки креираним скуповима података, а понекад изостаје упоредна анализа са изворним алгоритмом који се ослања на комплетно скенирање података.

Новија истраживања представљају решења проблема апроксимације ослањајући се на технологије за паралелизацију процеса и дистрибуиране рачунарске архитектуре. Неколико дистрибуираних алтернатива су предложене са циљем да омогуће апроксимативном алгоритму да обради велику количину података. Већина решења се базира на програмској парадигми *MapReduce* и његовој имплементацји отвореног кода – *Hadoop*. На тај начин се паралелизује извршавање алгоритма и ублажавају ефекти меморијских и рачунских ограничења. Недавно је представљен нови дизајн (Maillo, Ramirez, Triguero, & Herrera, 2017) који превазилази стандардне *Hadoop*-*MapReduce* приступе и обезбеђује флексибилну шему за класификацију великог броја непознатих инстанци над великим подацима, користећи *Apache Spark* архитектуру (Triguero, Maillo, Luengo, García, & Herrera, 2016).

# ФОРМУЛАЦИЈА ПРОБЛЕМА

Анализа перформанси и примене апроксимативних техника за претрагу најближих суседа представља срж проблема овог рада. Време је један од кључних ресурса данашњих предузећа. У складу са тим, расте потреба за брзим генерисањем резултата за предвиђање. Осим времена и ефикасног коришћења ресурса, тачност предвиђања је један од критичних захтева. Управо су дате чињенице показатељ вредности изучавања ове области машинског учења.

Најважнији резултат истраживања би могла бити препорука за правилан одабир алгоритма и његових параметара. У раду ће бити имплементирани и промењени одабрани апроксимативни алгоритми. Резултат истраживања обухватиће детаљан приказ неколико врста апроксимативних алгоритама и структура података на које се ослањају. У истраживању ће се спровести упоредна анализа алгоритама и њихових перформанси.

Истраживање треба да одговори на следећа питања:

* Шта је апроксимативни алгоритам најближих суседа?
* Које су најпопуларније апроксимационе технике?
* Колика је разлика у времену потребном за спровођење изворног алгоритма и апроксимативних алгоритама?
* Колика је разлика у резултатима предвиђања апроксимативних решења и изворног КНН-а?
* Да ли је, и у којој мери је могуће паралелизовати извршење алгоритма?
* Колики је утицај структуре података на брзину извршавања и резултате предвиђања апроксимативних варијанти алгоритма?
* Које ће перформансе и резултате апроксимативни алгоритми постићи при раду над скуповима различитих величина и броја атрибута?
* Да ли је могуће извести препоруке за коришћење одговарајућег апроксимативног алгоритма у зависности од података и корисничких захтева?

# МОТИВАЦИЈА: ИНТЕЛЕКТУАЛНА И ЛИЧНА МОТИВИСАНОСТ

# ОРГАНИЗАЦИЈА И МЕТОДЕ ИСТРАЖИВАЊА

Резултати истраживања ће се добити експерименталним путем. Алгоритми ће се примењивати нaд скуповима података различитих димензија под различитим параметрима. Оквирни план истраживања обухвата четири фазе:

* Изучавање одабране литературе о алгоритмима који ће се применити у раду
* Измена постојећих и израда сопствених имплементација алгоритама
* Припрема података и тестирање алгоритма на пресоналном рачунару и на рачунарском кластеру уколико је то могуће
* Квантитативна анализа добијених резултата
* Извођење закључака и препорука

Добру основу за приступ проблему дао је Ханан Самет (Samet, 2006). Аутор пружа теоријску и практичну базу за примену алгоритма над различитим структурама података, превасходно стаблима. Читаво поглавље, посвећено је искључиво двема техникама претраге најближих суседа – инкременталној и техници претраге у дубину. Међу корисним радовима који ће бити проучени је (Liu, Moore, Yang, & Gray, 2005) где се презентују алгоритми базирани на метричким стаблима. Аутори истичу једноставност алгоритама и сугеришу да је могуће постићи резултате компаративне алгоритму дефинисаном од стране (Indyk & Motwani, 1998) уз могућу већу ефикасност. Рад (Abbasifard, Ghahremani, & Naderi, 2014) који пружа поређење различитих функција удаљености и сложености алгоритама ће такође утицати на избор метoда које ће се имплементирати.

Препоруке за избор структуре погодне за претрагу у зависности од карактеристика података који се обрађују дали су (Muja & Lowe, 2009). Утврђено је да два алгоритма од којих се оба заснивају на стаблима (енг. *randomized kd-tree* и *hierarchical k-means tree*) пружају најбоље резултате. Значајно је напоменути да би број инстанци код К-Д стабла морао бити значајно већи у односу на број димензија , јер у супротном не би било побољшања у времену у односу на комплетну претрагу свих инстанци (Arya, Mount, & Narayan, Accounting for boundary effects in nearest-neighbor searching, 1996).

Следећи корак у изради истраживања засниваће се на проучавању литературе које се тиче примене алгоритама у случајевима када подаци над којима се извршавa имају већи број димензија.

Значајно другачији приступ у проналажењу најближих суседа даје рад (Indyk & Motwani, 1998). Идеја је да се пронађе метод којим ће се међусобно слични објекти сврставати у исте категорије. Хеширање на основу положаја или локацијски-сензитивно хеширање (енг. *Locality-sensitive hashing,* у даљем тексту ЛСХ) је назив представљене технике. Дефиниција коју је дао (Samet, 2006) даје ближе објашњење:

Циљ ЛСХ алгоритма је да се пронађе функција хеширања која ће сачувати информацију о удаљености инстанци, или приближне удаљености уз одређену толеранцију. Општи кораци ЛСХ алгоритма представљени су у раду под називом „Претраживање сличности при великом броју димензија путем хеширања“ (Gionis, Indyk, & Motwani, 1999). Наводи се да када је број димензија између 10 или 20, К-Д стабла и њима сличне структуре генеришу резултате претраге тек незнатно брже од линеарне претраге.

Раније је напоменуто да ће време бити кључни идентификатор перформанси. Друге мере евалуације које би се могле дефинисати су:

* Грешка код функције удаљености – разлика у вредности функције удаљености од непознате инстанце између апроксимативних и правих суседа
* Грешка у идентификацији – број идентификованих правих суседа у односу на број тражених
* Трошкови претпроцесирања – размотриће се време, сложености поступка и меморисјки захтеви алгоритма
* Тачност – проценат правилно класификованих инстанци

Садржај мастер рада ће бити подложан променама у зависности од тока истраживања. Следи оквирни садржај:

1. Увод
2. Апроксимативни алгоритми најближих суседа
   1. Преглед стања у предметној области
   2. Алгоритам заснован на К-Д стаблима
      1. Дефиниција К-Д стабла
      2. Попуњавање К-Д стабла
      3. Претраживање К-Д стабла
   3. Алгоритам заснован на хеш функцијама
   4. Дистрибуиране технологије за апроксимацију
3. Имплементација
   1. Анализа скупова података
   2. Тестирање алгоритама
4. Резултати истраживања
   1. Упоредни приказ резултата
   2. Препоруке
5. Закључак
6. Прилози

Алгоритми ће се имплементирати у Пајтон (eng. *Python*) програмском језику. У разматрање ће се узети постојећа решења и библиотеке попут *scipy.spatial.KDTree* или *sklearn.neighbors.kd\_tree* и друге. При истраживању могућности имплементације над *Apache Spark* архитектуром користиће се *PySpark* програмски интерфејс.

# Опис дела света који ће бити изучаван

Претпоставка аутора је да ће се користити скупови података доступни на адреси https://archive.ics.uci.edu/ml/datasets.html (UCI Machine Learning Repository). Међу кандидатима за почетну анализу налазе се мањи скупови који имају више од сто хиљада инстанци попут *Covertype Data Set*. Изазован кандидат за анализу представља скуп под називом *HIGGS,* где задатак разлучивање да ли сигнал означава појаву Хигсовог бозона или не.

# РЕФЕРЕНТНА ЛИТЕРАТУРА, КОНЦЕПТИ И ТЕОРИЈСКИ ФОКУС

Литература коришћена у приступном раду ће се такође користити при изради мастер рада:

Abbasifard, M. R., Ghahremani, B., & Naderi, H. (2014). A survey on nearest neighbor search methods. International Journal of Computer Applications, 95, 39-52.

Arya, S., Mount, D. M., & Narayan, O. (1996). Accounting for boundary effects in nearest-neighbor searching. Discrete & Computational Geometry, 16, стр. 155-176.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., & Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Јournal of the ACM (JACM), 45, 891-923.

Gionis, A., Indyk, P., & Motwani, R. (1999). Similarity search in high dimensions via hashing. Vldb, 99, стр. 518-529. Edinburgh.

Han, J., Pei, J., & Kamber, M. (2012). Data mining: concepts and techniques (3rd). Waltham, Massachusetts, United States of America: Elsevier.

Indyk, P., & Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. ACM.

Kleinberg, J. M. (1997). Two algorithms for nearest-neighbor search in high dimensions. Proceedings of the twenty-ninth annual ACM symposium on Theory of computing (стр. 599--608). El Paso,Texas, USA: АCM.

Kushilevitz, E., Ostrovsky, R., & Rabani, Y. (2000). Efficient search for approximate nearest neighbor in high dimensional spaces. SIAM Journal on Computing, 30, 457-474.

Liu, T., Moore, A. W., Yang, K., & Gray, A. G. (2005). An investigation of practical approximate nearest neighbor algorithms., (стр. 825-832).

Maillo, J., Ramirez, S., Triguero, I., & Herrera, F. (2017). kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. Knowledge-Based Systems, 117, 3-15.

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1), 331-340.

Samet, H. (2006). Foundations of multidimensional and metric data structures. (J. Gray, Ур.) San Francisco, California, United States of America: Morgan Kaufmann.

Seidl, T., & Kriegel, H.-P. (1998). Optimal Multi-Step k-Nearest Neighbor Search. 27. Seattle: ACM SIGMOD Record.

Triguero, I., Maillo, J., Luengo, J., García, S., & Herrera, F. (2016). From big data to smart data with the k-nearest neighbours algorithm.

UCI Machine Learning Repository. (n.d.). Преузето са University of California, Irvine: <https://archive.ics.uci.edu/ml/datasets.html>

Додатна литература обухвата следеће радове:

Andoni, A., & Indyk, P. (2006, October). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on (pp. 459-468). IEEE.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999, January). When is “nearest neighbor” meaningful?. In International conference on database theory (pp. 217-235). Springer, Berlin, Heidelberg.

Datar, M., Immorlica, N., Indyk, P., & Mirrokni, V. S. (2004, June). Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the twentieth annual symposium on Computational geometry (pp. 253-262). ACM.

Hajebi, K., Abbasi-Yadkori, Y., Shahbazi, H., & Zhang, H. (2011, July). Fast approximate nearest-neighbor search with k-nearest neighbor graph. In IJCAI Proceedings-International Joint Conference on Artificial Intelligence (Vol. 22, No. 1, p. 1312).

Kushilevitz, E., Ostrovsky, R., & Rabani, Y. (2000). Efficient search for approximate nearest neighbor in high dimensional spaces. SIAM Journal on Computing, 30(2), 457-474.

Maillo, J., Triguero, I., & Herrera, F. (2015, August). A mapreduce-based k-nearest neighbor approach for big data classification. In Trustcom/BigDataSE/ISPA, 2015 IEEE (Vol. 2, pp. 167-172). IEEE.

Otair, D. (2013). Approximate k-nearest neighbour based spatial clustering using kd tree. arXiv preprint arXiv:1303.1951.

Slaney, M., & Casey, M. (2008). Locality-sensitive hashing for finding nearest neighbors [lecture notes]. *IEEE Signal processing magazine*, *25*(2), 128-131.

# Сврха и циљеви истраживања

Сврха овог истраживања је да прикаже могућности унапређења стандардног алгоритма за проналажење најближих суседа применом метода апроксимације, што ће омогућити истраживачима и другим заинтересованим странама да брже и ефикасније спроводе предиктивне анализе над већим скуповима података. Аутор се нада да ће мастер рад обогатити скроман фонд литературе која се бави поменутом проблематиком на српском језику и олакшати будућим мастер студентима упознавање са темом. Истраживање ће дати преглед и поређење досадашњих резултата и указати на њихове предности и мане. Циљ рада је унапређење и експериментално поређење перформанси апроксимативних алгоритама који се извршавају над великим подацима, као и теоријски осврт на кораке и сложеност таквих алгоритама.

# РЕФЕРЕНЦЕ

Abbasifard, M. R., Ghahremani, B., & Naderi, H. (2014). A survey on nearest neighbor search methods. *International Journal of Computer Applications, 95*, 39-52.

Arya, S., Mount, D. M., & Narayan, O. (1996). Accounting for boundary effects in nearest-neighbor searching. *Discrete & Computational Geometry, 16*, стр. 155-176.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., & Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Јournal of the ACM (JACM), 45*, 891-923.

Gionis, A., Indyk, P., & Motwani, R. (1999). Similarity search in high dimensions via hashing. *Vldb*, *99*, стр. 518-529. Edinburgh.

Han, J., Pei, J., & Kamber, M. (2012). *Data mining: concepts and techniques* (3rd изд.). Waltham, Massachusetts, United States of America: Elsevier.

Indyk, P., & Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. ACM.

Kleinberg, J. M. (1997). Two algorithms for nearest-neighbor search in high dimensions. *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (стр. 599--608). El Paso,Texas, USA: АCM.

Kushilevitz, E., Ostrovsky, R., & Rabani, Y. (2000). Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing, 30*, 457-474.

Liu, T., Moore, A. W., Yang, K., & Gray, A. G. (2005). An investigation of practical approximate nearest neighbor algorithms., (стр. 825-832).

Maillo, J., Ramirez, S., Triguero, I., & Herrera, F. (2017). kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. *Knowledge-Based Systems, 117*, 3-15.

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 331-340.

Samet, H. (2006). *Foundations of multidimensional and metric data structures.* (J. Gray, Ур.) San Francisco, California, United States of America: Morgan Kaufmann.

Seidl, T., & Kriegel, H.-P. (1998). Optimal Multi-Step k-Nearest Neighbor Search. *27.* Seattle: ACM SIGMOD Record.

Triguero, I., Maillo, J., Luengo, J., García, S., & Herrera, F. (2016). From big data to smart data with the k-nearest neighbours algorithm.

*UCI Machine Learning Repository*. (n.d.). Преузето са University of California, Irvine: https://archive.ics.uci.edu/ml/datasets.html