Plan spotkania. #1

eRka

Czwartek, 6 Listopada, 2014

18:00-18:20 Wprowadzenie do eR(k)a.

Bartosz Sękiewicz

Podczas prezentacji chciałbym przedstawić w pigułce historię oraz możliwości R, a także popularność tego środowiska w Polsce (w szczególności w Krakowie). Większość z zaprezentowanych informacji będzie dobrze znana użytkownikom R, jednakże może stanowić punkt startowy dla osób, które mają małą styczność z tym językiem, ale chciałyby zmienić ten stan rzeczy. Na koniec omówimy przyszłość eRka – liczę na owocną dyskusję.

18:20-18:50 Algorytm Cross Entropy Clustering oraz pakiet CEC.

Dr Przemysław Spurek

Klasteryzacja (grupowanie) jest bardzo ważnym zagadnieniem w różnych dziedzinach nauki takich jak biologia, bioinformatyka, medycyna, biznes i marketing, informatyka (software evolution, image segmentation, evolutionary algorithms, recommender systems, Markov chain Monte Carlo (MCMC)) oraz social science. Jest to powodem dynamicznego rozwoju tej teorii oraz ciągłego powstawania coraz lepszych algorytmów grupowania. W środowisku R powstało wiele pakietów przeznaczonych do klasteryzacji takich jak mclust, pdfCluster, mixtools, clues, HDclassif, czy ClustOfVar. Podczas prezentacji postaram się przedstawić, dlaczego warto wybrać algorytm Cross Entropy Clustering oraz pakiet CEC.

Największym problemem związanych z grupowaniem jest dobieranie liczby klastrów. Powstało wiele metod opartych na analizie efektów klastrowania dla różnych parametrów. Niestety posiadają one dużą złożoność obliczeniową (opierają się na kilkukrotnym wykonaniu całego algorytmu klasteryzacji). Metoda CEC sama dobiera ilość klastrów, mówiąc precyzyjnie redukuje zbędne grupy.

18:50-19:10 Optymalizacja R – dlaczego warto przesiąść się na Linuxa?

Zygmunt Zawadzki (https://github.com/zzawadz)

Tematem prezentacji będzie zaskakująco prosta optymalizacja działania R pod Linuxem. Standardowo zainstalowany R korzysta z dosyć wolnych bibliotek macierzowych, jednak dzięki zastosowaniu pewnej sztuczki, można te biblioteki łatwo podmienić na bardziej wydajne. Przykładowo - wykorzystując bibliotekę OpenBlas na Intel Core I7 3770K mnożenie macierzy 2500x2500 zajmuje ok 0,5 sekundy (wykorzystywane są wszystkie rdzenie procesora - w tym przypadku 4), a w wersji standardowej - ok 10 sekund. Również inne działania zostają przyspieszone (svd, eigen, pca, itd.). W trakcie prezentacji zostanie krótko poruszony temat jak realizowane są operacje macierzowe w R (słowa kluczowe - BLAS i LAPACK). Następnie zostanie na żywo przeprowadzony zabieg przyspieszania R, by pokazać, że rzeczywiście nie jest to trudne, a konsola w Linuxie nie jest taka straszna!