# PERTEMUAN 5 DAN 6

## METODE KNN DAN PENINGKATAN HASIL ANALISIS PADA PENGENALAN POLA

### TUJUAN PRAKTIKUM

1. Mahasiswa mampu memahami dan menjelaskan mengenai konsep *k*-*Nearest Neighbour* (k-NN) *Density estimation* dan *Classification*
2. Mahasiswa mampu mengimplementasikan k-NN untuk klasifikasi Data sederhana menggunakan R
3. Mahasiswa mampu memahami dan menjelaskan mengenai salah satu cara peningkatan analisis dalam pengenalan pola

### TEORI PENUNJANG DAN MATERI PRAKTIKUM

### Algoritme k-NN

Algoritme *mechine learning* menyediakan metode/teknik untuk mengklasifikasi-kan sebuah objek ke dalam beberapa grup berdasarkan nilai dari variabel penjelas. Salah satunya adalah Metode *nearest neighbor* (NN) . Metode ini cukup mudah diimplemen-tasikan dan untuk dimengerti. Namun tidak seperti beberapa metode dalam *machine learning*, NN tidak memiliki model yang berasosiasi dengan proses (algoritme) pembelajaran nya, sehingga keasalahan(galat) harus diestimasi secara komputasional, tetapi metode ini menghasilkan 1 solusi sederhana untuk mengklasifikasi sebuah objek varu berdasarkan hasil dari himpunan referensi yang diketahui. *K-NEAREST NEIGHBOUR* (KNN) adalah sebuah generalisasi objek yang memiliki kedekatan kriteria ke suatu himpunan referensi objek, maka dianggap satu grup/kelas. KNN adalah salah satu contoh algoritme dalam machine learning/data mining yang sederhana dan salah satu contoh dari *instanced-based learning*. Dimana data diklasifikasikan berdasarkan himpunan referensi yang menyimpan data sesuai label. Lebih lanjut jarak diantara data tersimpan (himpunan referensi) dan objek baru dikalkulasi menggunakan nilai ukur dari beberapa pengukuran kemiripan. Pengukuran kemiripan ini umumnya diekspresikan dengan pengukuran jarak, seperti *euclidian, cosine,* atau *manhattan distance*. KNN akan menetapkan label atau kelas ke objek baru (*new instance*)

**Review Step Pengenalan Pola dengan Proses Pembelajaran (Klasifikasi) di R**

1. **Prepare Your Work Directory and Get Data**

Langkah dimulai dari men*setting* direktori dan mengumpulkan data observasi.Untuk set direktori dalam R gunakan perintah setwd, atau getwd Untuk proses penelitian dan analisis suatu metode dalam R, kita dapat menggungakan data berupa *built-in datasets* dalam R, atau menggunakan data eksternal di luar R. Untuk *built-in datasets* kita dapat melihat sekumpulan dataset dalam daftar dengan mengetikan:

> library(help="datasets")

> ## contoh mengambil dataset IRIS

> IRIS

Jika ingin mengambil contoh dataset eksternal, dapat mencoba ke beberapa situs repositori penyedia data seperti UC Irvine Machine Learning, dengan mengunduh datanya langsung melalui webpagenya.

>iris<-read.csv(url("http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"), header = FALSE)

* Bagaimana jika header di set menjadi False, dan bagaimana jika diset menjadi True?
* Jika tanpa ada header, bagaimana memunculkan header yang sesuai dengan kita inginkan.

1. **Know Your Data (optional)**

Setelah mendapatkan Data, maka selanjutnya analisis awal data (*initial overview of Data set*), dengan tujuan dapat mengerti dengan bentuk data. Untuk mengetahui bentuk data secara umum dapat menggunakan *scatter plots* untuk melihat seberapa banyak 1 variabel dapat dipengaruhi oleh variabel lain (melihat apakah ada kolerasi diantara dua variabel)

Untuk menampilkan scatter plots bisa menggunakan **ggvis package**

> library (ggvis)

> iris%>%ggvis(~Sepal.length,~sepal.width, fill =~Kelas) %>% layer\_points()

* Lakukan analisis terhadap dua variabel sepal. Length, dan sepal.width untuk ketiga setiap ketiga kelas yang diperlihatkan oleh grafik?
* Lakukan analisis terhadap dua variabel petal. length, dan petal.width untuk ketiga setiap ketiga kelas yang diperlihatkan oleh grafik. Apa hasilnya?

1. **Prepare Your Data**

Sebagai bagian dari preparasi data, adalah memastikan bahwa data konsisten disetiap variabel/fitur sehingga tidak menjadi *over-emphasise*. Salah satunya adalah menggunakan cara berikut:

prepare

prepare(x){

a=(x-min(x))

b= max(x)-min(x)

return(a/b)

}

* Fungsi apakah prepare(x) itu?
* Kenapa diperlukan untuk memastikan bahwa data yang disiapkan dan digunakan tidak *over-emphasise*?
* Bagaimana menerapkan alforitme untuk fungsi tersebut?

Proses preparasi data selanjutnya adalah menetapkan data training dan data testing untuk input bagi algoritme pembelajaran k-NN. Untuk tujuan menilai performansi model nantinya, maka diperlukan. pembagaian data menjad data training dan data testing. Data training digunakan untuk proses pembelajaran, sementara data testing digunakan untuk mengevaluasi hasil dari proses pembelajaran dari data training tersebut. Pada paraktiknya pembagian data haruslah disjoint dengan ukuran umum pada data yang relatif cukup besar adalah 2/3 untuk training set, dan 1/3 untuk testing set. Namun untuk data yang ukurannya relatif kecil dapat digunakan metode *k-fold cross validation* (sudah diberikan pada pertemuan sebelumnya) untuk analisis performansi model.

Lakukan pembagian data (spliting data) untuk data IRIS denngan perbandingan data adalah 0.67 untuk training set, dan 0.33 untuk testing set.

> set.seed(1234)

Merupakan sebuah bilangan dari random number generator, untuk bisa mendapatkan skuens yang sama dari random number generator

> ind<-sample(2, nrow(iris), replace = TRUE, prob=c(0.67,0.33))

> iris.trainset <- iris[ind==1,(1:4)]

> iris.testset <- iris[ind!=1,(1:4)]

> iris.trainlabel <- iris[ind==1,5]

> iris.testlabel <- iris[ind!=1,5]

1. **Building the Clasifier**

Setelah langkah preparasi selesai, dan sudah memastikan bahwa data telah dilakukan normalisasi dan spliting dengan proporsi yang sesuai, langkah selanjutnya adalah menerapkan proses pembelajaran terhadap data. Dalam R, untuk algoritme KNN sudah tersedia dalam *package* class**.** Unduh package untuk menerapkan algoritme pembelajaran k-NN.

> any(grepl("class", installed.packages()))

> library(class) #install.packages("class")

> '?' (knn)

Untuk membangun *classifier* KNN, Fungsi yang digunakan adalah **knn(), dengan argumen berikut:**

**>** iris.pred <- knn(train = iris.trainset, test = iris.testset, cl = iris.trainlabel, k=3)

1. **Evaluation your model**

Tahapan Esensial dalam algoritme pembelajaran adalah evaluasi/ analisis performansi dari metode pembelajaran/ model. Dengan kata lain, kita ingin menganalisis *the degree of correctness* dari prediksi hasil metode pembelajaran, secara sederhana menggunakan matriks konfusi

> table(iris.pred, iris.testlabel)

Untuk analisis lebih dalam dalam menampilkan matriks konfusi, dapat menggunakan package gmodels

> install.packages("gmodels")

> library (gmodels)

> crossTable(x=iris.testlabel, y=iris\_pred, prop.chisq=FALSE)

* Cobakan untuk k yang berbeda, k=5,7,9,11,13
* Perlihatkan dengan matriks konfusi untuk nilai k tersebut

### LATIHAN PRAKTIKUM

### Ikuti prosedur berikut ini dan buatlah script-r untuk melakukan teknik klasifikasi menggunakan R

* Lakukan Prosedur klasifikasi K-NN untuk data "Wisconsin Breast Cancer Dataset" yang dapat diunduh di situs *UC-Irvine Machine Learning Archive*. Jumlah sampel dataset adalah 569, sebagian bening dan sebagian tumor.
* Berikan label untuk data frame tersebut berurutan dengan nama: Id, diagnosis, rad\_mean, texture\_mean, perimeter\_mean, area\_mean, smoothness\_mean, compactness\_mean, concavity\_mean,points\_mean, sym\_mean, dimn\_mean, rad\_se, texture\_se, perimeter\_se, area\_se, smoothness\_se, compactness\_se, concavity\_se, points\_se, sym\_se, dimn\_se, rad\_worst, texture\_worst, perimeter\_worst, area\_worst, smoothness\_worst, compactness\_worst, concavity\_worst, points\_worst, sym\_worst, dimn\_worst
* Organisasi Data dengan lakukan cek jumlah diagnosis, dan cek untuk data missing value dan ubah diagnosis (B, dan M) ke dalam bentuk factor labels ("benign", "malignant")
* Lakukan Normalisasi terhadap data (tambahkan identitas pasien pada rownames)
* Lakukan Pembagian Data (Data untuk training dan validasi) gunakan “sample”
* Lakukan Klasifikasi KNN (Menggunakan Library Class)
* Analisis Hasil Prediksi
* Cobakan untuk beberapa nilai k= 3, 7, 11, 31

**Bagaimana kita dapat meningkatkan hasil analisis KNN?**

2 Faktor yang dapat menjadi pertimbangan untuk peningkatan hasil analisi adalah:

* Set Variabel yang optimum untuk menghasilkan penghitungan jarak yang lebih baik;
* Nilai k yang optimum.

Kendala yang ditemukan adalah untuk melihat parameter mana yang memberikan nilai terbaik (klasifikasi yang lebih baik) pada proses pembelajaran dan validasi hasil.

**Signifikansi Variabel pada Proses Klasifikasi**

Dalam proses pengenalan pola, fitur yang representatif sangat diperlukan untuk algoritme pembelajaran mengenali pola pembeda dari suatu objek. Berikut ini adalah studi terkait melihat pengaruh yang signifikan dalam pengenalan pola

names(wbcd.trainset)

lm\_1 <- lm(rad\_mean ~ wbcd.trainlabel, data = wbcd.trainset)

summary(lm\_1)

Call:

lm(formula = rad\_mean ~ wbcd.trainlabel, data = wbcd.trainset)

Residuals:

Min 1Q Median 3Q Max

-0.30948 -0.06746 0.00181 0.06723 0.50268

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.249109 0.007376 33.77 <2e-16 \*\*\*

wbcd.trainlabelmalignant 0.248215 0.012376 20.06 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1155 on 378 degrees of freedom

Multiple R-squared: 0.5156, Adjusted R-squared: 0.5143

F-statistic: 402.3 on 1 and 378 DF, p-value: < 2.2e-16

names(summary(lm\_1))

summary(lm\_1)$fstatistic

value numdf dendf

402.281 1.000 378.000

##The significance measure we want:

summary(lm\_1)$fstatistic[1]

value

402.281

Tampilkan untuk semua variabel dengan menggunakan vektor

exp\_var\_fstat <- as.numeric(rep(NA, times = 30))

names(exp\_var\_fstat) <- names(wbcd\_normalisasi)

Selanjutnya tampilkan nilai F-Statistic untuk setiap variabel

exp\_var\_fstat["rad\_mean"] <- summary(lm(rad\_mean ~ wbcd.trainlabel, data = wbcd.trainset))$fstatistic[1]

exp\_var\_fstat["texture\_mean"] <- summary(lm(texture\_mean ~ wbcd.trainlabel, data = wbcd.trainset))$fstatistic[1]

exp\_var\_fstat["perimeter\_mean"] <- summary(lm(perimeter\_mean ~ wbcd.trainlabel, data = wbcd.trainset))$fstatistic[1]

exp\_var\_fstat.

**About significance of the variables**

most\_sig\_stats <- sort(exp\_var\_fstat, decreasing = T)

names(most\_sig\_stats) <- exp\_vars

most\_sig\_stats[1:10]

most\_sig\_stats[25:30]

wbcd\_train\_ord <- wbcd.trainset[, names(most\_sig\_stats)]

Dari hasil tersebut telah memberi pandangan, 5 variabel terakhir tidak terlalu signikan. Menambahkan variable tersebut di dalam list fitur model hanya dapat menambahkan varian. Untuk mempersiapkan pembelajaran KNN dengan beberapa variable, *reorder* variabel pada training set sesuai nilai signifikansinya. Langkah selanjutnya untuk menghasilkan berapa banyak variabel yang memberikan nilai terbaik (best fit), dapat menggunakan salah satu versi dari cross validation.

**Menemukan nilai optimal untuk kombinasi fitur dan Parameter k-NN**

**Eksekusi hasil tes dengan loops**

Buat sejumlah nest dari for loops untuk mengkalkulasi errors dari semua training set dan pilihan model parameter. Yang terpenting perlu diingat adalah:

1. Kita harus menyimpan error dan mengasosiasikanya dengan parameter ka dan menghitung erorrnya secara bersamaan. Menyimpannya dalam bentuk data frame memungkinkan untuk menhanalisis signifikansi dan ploting
2. Kita harus menambahkan setiap entries ke data frame didalam nested loop untuk setiap iterasi.

Berikut adalah sebuah fungsi baru untuk menghitung error KNN dengan setiap pilihan parameter. Contoh MC Split, n=5, k=7.

knn\_test <- knn(train = wbcd\_train\_ord[1:250, 1:5], test = wbcd\_train\_ord[251:380, 1:5], cl=wbcd.trainlabel[1:250], k = 7)

tbl\_test <- table(knn\_test, wbcd.trainlabel[251:380])

tbl\_test

err\_rate <- (tbl\_test[1, 2] + tbl\_test[2, 1])/length(knn\_test)

err\_rate

### TUGAS PRAKTIKUM

**Di LMS.**

### DAFTAR PUSTAKA

1. Victor A. Bloomfield. 2014. *Using R for Numerical Analysis in Science and Engineering*. 1 edition. Chapman and Hall/CRC