Jonatan Hoffmann Hanssen Grzegorz Kajda Adrian Duric

TEK5020 Prosjektoppgave 1 – Rapport

Fremgangsmåte

Som beskrevet i oppgaven, har vi brukt nærmeste-nabo klassifikatoren til å estimere feilraten for alle mulige kombinasjoner av egenskaper for et gitt antall dimensjoner, for hvert mulige antall av dimensjoner. For hvert antall av dimensjoner finner vi da den beste kombinasjonen av mulige egenskaper (den med minst estimert feilrate), og bruker de tre klassifikatorene på denne kombinasjonen av egenskaper for å finne den beste klassifikatoren for den egenskapskombinasjonen.

Resultater

Følgende feilrater ble oppnådd ved å utføre nærmeste nabo klassifikasjon på treningsdatasettet for ulike kombinasjoner av egenskaper. De beste (laveste) feilratene per datasett, per antall dimensjoner, er markert i grønt:

Egenskaper	Datasett 1	Datasett 2	Datasett 3
1	22.0%	18.7%	28.5%
2	39.3%	28.7%	30.0%
3	41.3%	46.0%	36.0%
4	33.3%	-	38.0%
(1, 2)	25.3%	2.7%	24.5%
(1, 3)	22.7%	18.7%	17.5%
(1, 4)	14.7%	-	21.5%
(2, 3)	38.0%	37.3%	16.0%
(3, 4)	30.7%	-	19.5%
(1, 2, 3)	21.3%	2.7%	15.5%
(1, 3, 4)	14.7%	-	13.0%
(2, 3, 4)	21.3%	-	11.5%
(1, 2, 3, 4)	16.0%	-	15.0%

Ved å trene de tre klassifikatorene på de beste egenskapene i treningssettet per antall dimensjoner, og deretter predikere på testsettet, fikk vi følgende feilrater. Beste feilrate per datasett, per antall dimensjoner, er markert i grønt.

Én dimensjon

Datasett	Nærmeste nabo	Lineærdiskriminant	Minste feilrate Gauss
1	24.0%	22.00%	18.7%
2	18.0%	52.7%	10.7%
3	32.5%	50.0%	35.5%

To dimensjoner

Datasett	Nærmeste nabo	Lineærdiskriminant	Minste feilrate Gauss
1	16.7%	22.0%	11.3%
2	1.3%	36.7%	2.0%
3	12.0%	50.00%	20.0%

Tre dimensjoner

Datasett	Nærmeste nabo	Lineærdiskriminant	Minste feilrate Gauss
1	12.7%	26.0%	8.7%
2	2.0%	12.0%	2.0%
3	10.5%	16.00%	13.0%

Fire dimensjoner

Datasett	Nærmeste nabo	Lineærdiskriminant	Minste feilrate Gauss
1	9.3%	7.3%	8.0%
2	-	-	-
3	11.5%	12.00%	7.0%

Avsluttende spørsmål

Hvorfor er det fornuftig å benytte nærmeste-nabo klassifikatoren til å finne gunstige egenskapskombinasjoner?

Nærmeste-nabo klassifikatoren er en ikke-parametrisk metode, og derfor antar den ingenting om tetthetsfunksjonene til det underliggende datasettet, og kan derfor være svært nyttig å bruke i tilfeller der ingen logiske antagelser kan bli trukket om dataen. Metoder som nærmeste-nabo egner seg også av den grunn svært godt til approksimering av de sanne tetthetene, særlig når datasettet er stort. Dette fordi feilraten synker asymptotisk for nærmeste-nabo, og gode estimater oppnås som oftest jo flere datapunkter vi har. Med andre ord, gitt et tilstrekkelig stort dataset, er vi nesten alltid garantert å finne egenskapskombinasjoner som i verste fall fører til en feilrate med dobbel så høy amplitude som den bayesiske feilraten. Dessuten er denne metoden, som nevnt i oppgaveteksten, enkel å både forstå intiuitivt og å programmere. Den gir et godt og lettforståelig mål på hvilke egenskapskombinasjoner som inneholder mest variabilitet.

Hvorfor kan det i en praktisk anvendelse være fornuftig å finne en lineær eller kvadratisk klassifikator til erstatning for nærmeste-nabo klassifikatoren?

I praksis har nærmeste-nabo klassifikatoren noen bemerkelsesverdige ulemper som blant annet at den er regnemessig kostbar, og krever som oftest høy minnekapasitet. Vanligvis når vi bruker nærmeste-nabo, er vi nødt til å lagre hele datasettet for å kunne beregne avstanden til hvert eneste punkt, for hvert punkt vi ser på. Ofte kan vi også trekke noen konklusjoner om dataen vi skal analysere på forhånd, som f.eks. at den er

normalt fordelt. Dette forenkler estimering av a priori og a posteriori sannsynligheter, og tillater bruk av teknikker som minimum feilrate og bayesisk estimering, som er kjent for å ha lavere hardware-krav.

I situasjoner der datasetet er ganske lite, og ujevnt, kan vi også risikere at feilraten blir betydelig høy, og ligge tett opp mot det dobbelte av optimal feilrate. I ekstreme tilfeller der den bayesiske feilraten er høy i seg selv, kan det hende at NN vil ha dobbelt så høy feilrate, noe som åpenbart er uønskelig. Ikke minst baseres også nærmeste-nabo kun på et mål på avstand, som kriteriet for tildeling av klasse tilhørighet for punkter i datasetet. Det kan derfor hende at vi går glipp av viktig latent informasjon som ikke kan oppdages ved kun bruk av så enkle tilnærminger. Dette fører til at nærmeste-nabo klassifikatoren er utsatt for overfitting; den er altså svært sensitiv for støy i datasettet. Lineære og kvadratiske klassifikatorer kan både lede til færre utregninger, og potensielt bidra til å redusere overfitting og gi en bedre balanse mellom bias og varians.

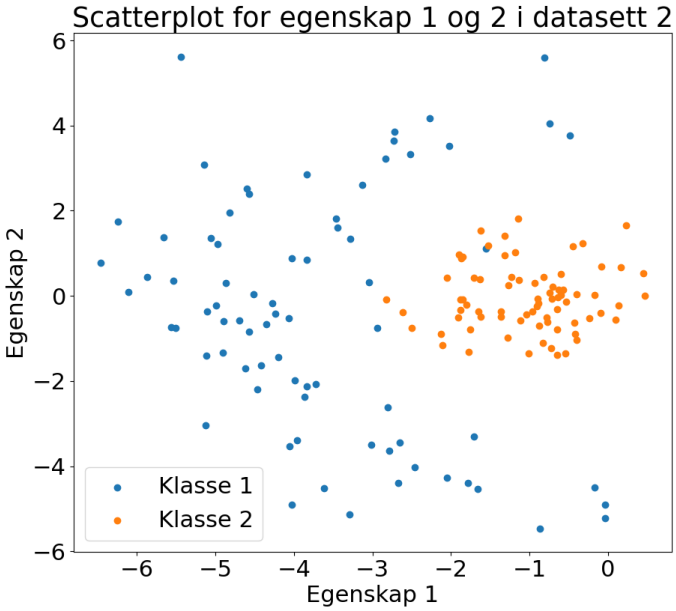
Hvorfor er det lite gunstig å bruke samme datasettet både til trening og evaluering av en klassifikator?

Å bruke samme datasett til trening og evaluering av en klassifikator gjør at når man evaluerer presisjonen av klassifikatoren på testsettet (som nå også er treningssettet), er ikke denne presisjonen representativ for hvilken presisjon klassifikatoren hadde fått på vilkårlige data samplet fra samme distribusjon som datasettet.

Dette er fordi klassifikatoren har sett, og trent på, dataen den blir evaluert på. Når vi evaluerer en klassifikator, ønsker vi å måle hvor presis den er på usette data fra samme distribusjon som treningsdataene. Klassifikatoren er laget for å optimalisere desisjonsgrensen for dataene den har sett, og når den trener på testsettet, blir den unaturlig god på å klassifisere akkurat testsettet, i forhold til hvordan den hadde vært på usette data.

Hvorfor gir en lineær klassifikator dårlige resultater for datasett 2?

Ved å plotte egenskapskombinasjonen av datasett 2 som ga best resultat for nærmeste-nabo klassifikatoren, ser vi at datasettet i disse dimensjonene ikke er lineært separabelt (se figuren nedenfor). Det kreves altså en ikke-lineær desisjonsgrense for å optimalisere grensen, som lineære klassifikatorer ikke er i stand til å produsere.



Jonatan Hoffmann Hanssen Grzegorz Kajda Adrian Duric