Kohonen Net

Peter Johansen

27. november 1998

Vektorkvantisering

Et område hvor klyngeanalyse har stor praktisk betydning, er til signaltransmission. Hvis et signal, for eksempel et digitalt billede, skal transmitteres anvendes hyppigt en metode som kaldes vektorkvantisering. *kvantisering* betyder at metoden opfattes som generalisering af at kvantisere et analogt signal. Hvis man har et fast antal bit til rådighed til hver stikprøve af et signal kan man spørge hvordan man skal kvantisere den analoge værdi for at fejlen bliver så lille som muligt. Med andre ord hvordan konstruerer vi en *kodebog* som for hver bitkombination oplyser hvilket analogt interval den repræsenterer? I stedet for et analogt signal kan man også anvende denne terminologi hvis signalet er digitalt og vi ønsker at repræsentere det ved færre bit. Generaliseringen fra kvantisering af en skalar værdi til *vektor*kvantisering fremkommmer ved at opfatte et diskret signal som et antal blokke eller vektorer af diskrete værdier. Anvender man en bloklængde n skal man derfor repræsentere et antal vektorer af dimension n. Hvordan skal man nu kvantisere en vektor? Hvordan vælger vi et repræsentativt sæt af vektorer for signalet? Hvis det kan lykkes for os, og vi skal transmittere signalet kan vi nøjes med at transmittere repræsentanten for en vektor i stedet for vektoren selv og på den måde transmittere er mindre antal bit per vektor. Med denne metode bortkastes selvfølgelig information. Med en lille kodebog fås en stor bitreduktion ved transmission men en dårlig rekonstruktion når modtageren, ved at slå op i kodebogen, skal rekonstruere signalet ud fra repræsentanter for signalets vektorer. Metoden blev først indført af Shannon for at vise at en prefix kode kan konstrueres så entropien kan tilnærmes vilkårligt godt. K-means algoritmen er en metode til at finde en kodebog.

Kohonen net

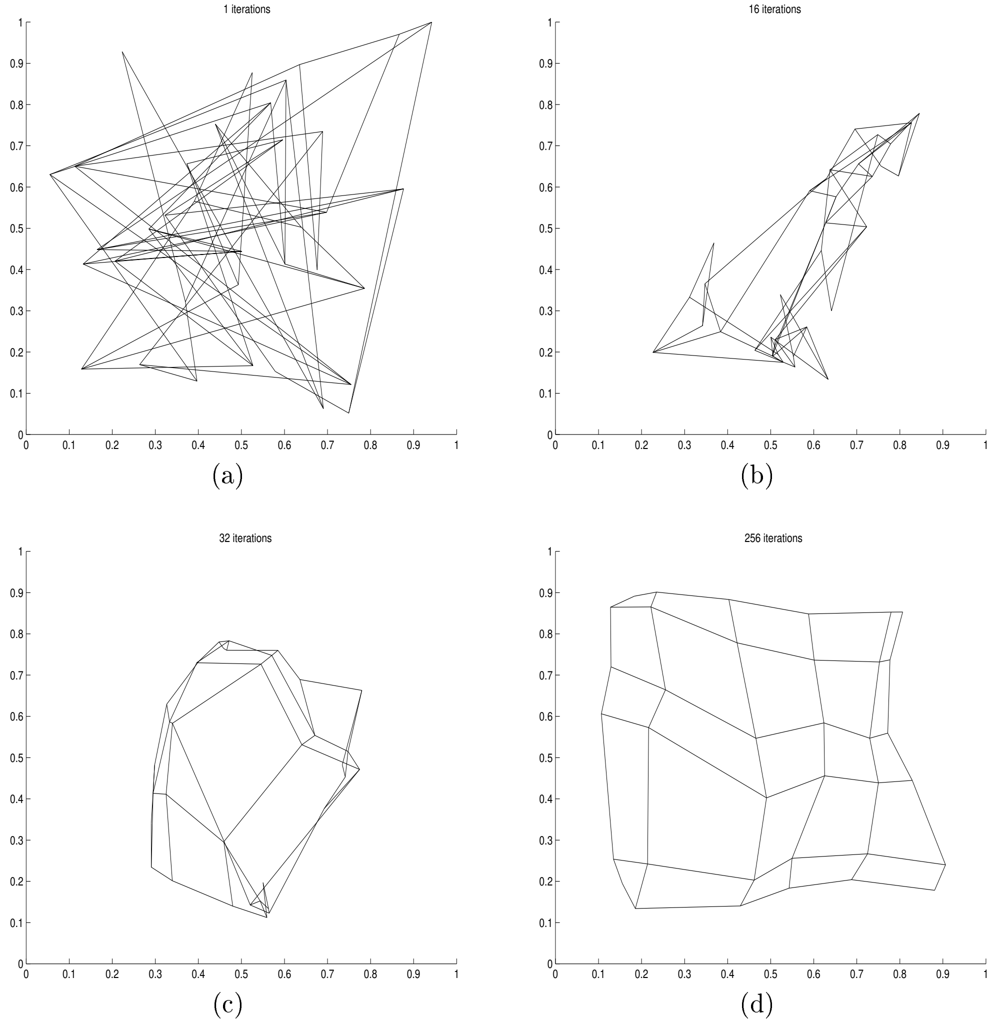
En metode til at finde klyngecentre er foreslået af Teuvo Kohonen[3]. Kohonens oprindelige formål var at konstruere selvorganiserende systemer og det er først senere at han har foreslået anvendelsen til klyngeanalyse.

Kohonen's algoritme Som ved k-means algoritmen foretages der et antal iterationer som gradvis forbedrer de oprindelige positioner af klyngecentrene. I sin enkleste version vælges elementerne $x \in M$ (M er mængden af featurepunkter) et ad gangen i en tilfældig rækkefølge. Elementet x finder det nærmeste klyngecentrum k og rykker det hen i retning mod x efter formlen

$$k \quad := \quad (1-\alpha)k + \alpha x \tag{1}$$

$0 < \alpha(t) < 1$ er en konstant som kan afhænge af iterationsnummeret t. α vil typisk vælges som en aftagende funktion af t. En forskel fra k-means algoritmen er at Kohonen's algoritme opdaterer et klyngecenters position hver eneste gang der er bestemt en nærmeste nabo. I k-means algoritmen flyttes klyngecentret først når alle korteste afstande er bestemt. Det flyttes derefter til tyngdepunktet for de nærmeste elementer $x \in M$.
For at forklare Kohonen's algoritme i sin fuldstændige version skal vi indføre en ekstra struktur på klyngecentrene. Hvert klyngecentrum knyttes til et punkt i en graf. Grafen er ofte et gitter i 1, 2 eller 3 dimensioner. Til hver pixel c er knyttet en omegn af centre N_c. Når et klyngecentrum opdateres efter formel (1) opdateres også klyngecentrene i N_c. $N_c(t)$ er en funktion af iterationsnummeret, til at begynde med en stor omegn og til sidst en lille omegn, eventuelt til sidst kun selve pixlen c.
For inituitivt at forstå virkningen af den aftagende funktion α kan man foretage en analogi med et fysisk system. Her fortolkes iterationsnummeret t som en temperatur. Klyngecentrene fortolkes som partikler der tiltrækkes af punkterne $x \in M$. Partiklernes kinetiske energi er større ved høj temperatur end ved lav temperatur. De bevæger sig derfor længere ved høj temperatur end ved lav temperatur. Først er temperaturen høj og systemet afkøles derefter langsomt. Denne analogi udnyttes også i den optimeringsmetode som kaldes *simulated annealing*. Resultatet vil være at den resulterende beliggenhed af klyngecentrene vil tendere til at fordele sig på samme måde som punkterne $x \in M$.
Virkningen af omegnen N_i som aftager med tiden er, at klyngecentrene vil bevæge sig i grupper, hvis størrelse aftager. I den fysiske analogi vil sammenhængen mellem klyngecentrene eller med et andet udtryk: viscositeten først være høj for derefter at aftage. Effekten bliver at den endelige fordeling af klyngecentrene vil ligne fordelingen af punkterne $x \in M$. Effekten af den måske tilfældige initiale placering af klyngecentrene bliver minimal. Man kan også sige at påvirkningen af klyngecentrene går fra at være global mod at være lokal. Skalaen for vekselvirkning formindskes.

Eksempel Algoritmen skal lære et regelmæssigt gitter. Begyndelsespositionerne af centrene er tilfældigt valgt. Datapunkter vælges tilfældigt i hele kvadratet.
Figur 1 viser resultatet efter og 1,16,32 og 256 iterationer. I hver iteration flyttes en klynge af centre henimod et tilfældigt valgt læredatum. Figuren viser hvorledes positionen af klyngecentrene ændrer sig. Der er tegnet linier mellem centre for naboer i getteret. Beskrivelse af dette og lignende eksempler kan findes i [2].



Figur 1: Fra tilfældigt beliggende centre til et kvadratisk gitter. Kohonens algoritme med α=0.25 i de første to trediedele af iterationerne og derefter $\alpha = .05$. I den første trediedel af iterationerne rykkes centre i en omegn som er 5×5, I den næste trediel er omegnen 3×3 og i den sidste trediedel er omegnen et enkelt centrum. (a) 1 iteration. (b) 16 iterationer. (c) 32 iterationer. (d) 256 iterationer.

Biologisk analogi Hjernebarken indeholder et sammenhængende område af nerveceller, den *motoriske cortex.* Hvis en nervecelle stimuleres elektrisk trækker muskler sig sammen på bestemt område af legemets overflade. Hvis en nabocelle stimuleres påvirkes muskler i nærheden af de forrige. Man kan sige at afbildningen af den motoriske cortex på legemets overflade er een-entydig og kontinuert. Som etikette på et lille område af hjernebarken kan man anbringe et billede af den del af kroppens overflade som påvirkes af de pågældende nerveceller. På denne måde afbildes et menneskes overflade på et stykke af hjernebarken. Man kan derfor tegne en *homunculus* på hjernebarken, hvor et område er stort hvis det har særlig fin motorisk kontrol (det gælder for eksempel tungen). Kohonen's algoritme kan danne en tilsvarende afbildning. Her er analogien mellem punktmængden M og punkter på legemets overflade og mellem de faste positioner for klyngecentrene og positioner på hjernebarkens overflade.

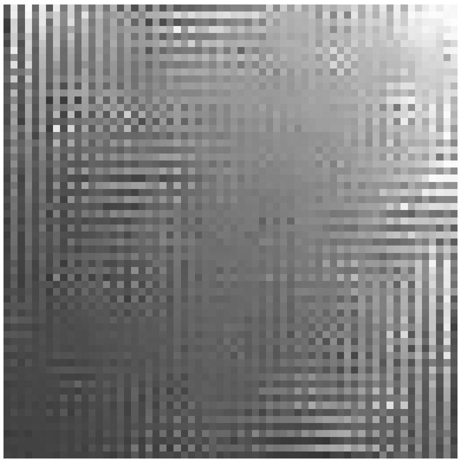
Søgning efter nærmeste nabo Det ekstra krav at klyngecentrene skal ligge i en rumlig struktur kan vi drage fordel af til klyngeanalyse. Når vi nemlig skal finde det nærmeste klyngecentrum til et punkt fra M kan vi søge ved at anvende gitterstrukturen, idet vi ved, at nabocentre sandsynligvis har værdier som er tæt ved hinanden. Gitteret anvendes som *index* i datamængden.
Vi underdeler gitret i 2^d "kvadranter" og finder afstanden til den midterse klynge i hver. Vi vælger det kvadrant hvis klynge har den korteste afstand til det ukendte datapunkt. Dette underdeles rekursivt indtil det består af en enkelt indgang i kodebogen. Denne vælges som kode for datapunktet.

Eksempel Figur 2 er et standard 256×256 billede. Det opdeles i små billeder på hver 2×2 pixels og indeholder derfor 128×128 småbilleder. Hvert småbillede indkodes ved det nærmeste småbillede i en kodebog som bygges med Kohonens algoritme.
Figur 3 viser kodebogen. Kodebogen er vist som et billede som består af 32*32 små billeder hver på 2*2 pixels, som er anbragt ved siden af hinanden i et 32×32 gitter. Hvert billede på 2*2 pixels er en indgang i kodebogen. Der er således $32 \times 32 = 1024$ indgange i kodebogen dvs en indgang i kodebogen er et kvadruple af gråtoneværdier. I kan lægge mærke til at de små billeder i nærheden af hinanden ligner hinanden. Det afspejler det forhold at kyngecentre for nabogitteret ligger tæt ved hinanden i $\{0, \ldots, 256\}$. Derfor kan man hurtigt søge efter det småbillede som mest ligner et givet 2*2 billede. Det kan gøres top-down ved på hvert trin at inddele i fire kvadranter og vælge den kvadrant hvis centerbillede mest ligner det givne billede [2]. Billederne er fra en phd afhandling [1].

Litteratur

[1] Ken Jakobsen. *Mobile Image Coding.* PhD thesis, Technical University of Denmark, Department of Electronics, 2800 Lyngby, Denmark, 1993.

Figur 2: Et 256×256 billede. Det opdeles i små billeder på hver 2×2 pixels og indeholder derfor 128×128 småbilleder. Hvert småbillede indkodes ved det nærmeste småbillede i kodebogen i næste figur.



Figur 3: Billedet viser en kodebog som er bygget ved hjælp af Kohonens algoritme som skitseret. Kodebogen er vist som et billede som består af 32*32 små billeder hver på 2*2 pixels, som er anbragt ved siden af hinanden i et 32×32 gitter. Hvert billede på 2*2 pixels er en indgang i kodebogen. Der er således $32 \times 32 = 1024$ indgange i kodebogen dvs en indgang i kodebogen er et kvadruple af gråtoneværdier. I kan lægge mærke til at de små billeder i nærheden af hinanden ligner hinanden. Det afspejler det forhold at kyngecentre for nabogitteret ligger tæt ved hinanden i $\{0, \ldots, 256\}$. Derfor kan man hurtigt søge efter det småbillede som mest ligner et givet 2*2 billede. Det kan gøres topdown ved på hvert trin at inddele i fire kvadranter og vælge den kvadrant hvis centerbillede mest ligner det givne billede [2].

[2] T. Kohonen. Clustering, taxonomy, and topological maps of patterns. In *Proceedings of the 6th International Conference on Pattern Recognition*, October 1982.

[3] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.