Big Data eller privacy?

Henglein, Fritz

Publication date:
2015

Document Version
Tidlig version også kaldet pre-print

Citation for published version (APA):
Henglein, F. (2015). Big Data eller privacy?. Paper præsenteret ved Copenhagen Fintech Innovation and Research (CFIR) Nordisk Fintech HUB, Danmark.

Big Data eller privacy?*

Fritz Henglein
Datalogisk Institut, Københavns Universitet (DIKU)

2015-07-15

Abstract

I dette positionspapir argumenteres der for, at Big Data, privacy og juridisk regulering skal samtænkes teknisk og policymæssigt, hvis de ikke skal stå i vejen for hinanden.

1 Big Data

Big Data dækker over behandlingen af ekstremt store datamængder i mange forskellige formater, der ydermere produceres med stor hastighed. Formålet er at opdage mønstre og uddrage informationer, så fortiden kan bruges til at forudsige fremtiden. Big Data er muliggjort af tre faktorer:

- eksponentielt stigende mængder af netværksforbundne digitale data,
- massiv regnekraft i alt fra *smartphones* til store *cloud computing*-centre og
- sofistikerede nye dataanalysemetoder, herunder statistisk maskinlæring (eng. *machine learning*).

Selvom om begrebet *Big Data* er på sit højeste i Gartner Groups *hype index* og kan forventes at få samme medfart som *eBusiness* og *dotcom*, så bygger det på en solid videnskabelig og teknologisk historie med et langsigtet anvendelsespotentiale, hvor kun fantasien – og måske regulering – sætter grænser. F.eks. blev maskinlæring, herunder forgængerne til de tidsaktuelle neurale netværk og *deep learning* metoder, udviklet i 1950erne med forskningsbevillinger opnået med løfter om udvikling af robotsoldater.

Der findes allerede nu en hel del anvendelser af Big Data, f.eks. automatisk billedgenkendelse, automatisk sprogoversættelse, automatisk match af datingprofiler, forudsigelser af sygdomsudvikling så som Alzheimer's og brystkræft. Inden for finans og forsikring kombineres finansielle data med sociale netværksdata og tilsyneladende irrelevante data fra sociale netværk for at analysere, hvor

*Tidligere version omdelt ved årsmødet for foreningen Copenhagen Fintech Innovation and Research (CFIR), maj 2015

kreditværdige potentielle lånekunder er, eller hvor stor deres skadesrisiko er. Nye virksomheder etablerer således i stigende grad lånemarkedspladser uden om det traditionelle banksystem. Betalingstransaktioner analyseres rutinemæssigt for at afsløre mulig misbrug af kreditkort eller for at identificere lånekunder, der har stor risiko for at komme i betalingsvanskeligheder. Netbankkunder kan i princippet få deres forbrug og investeringer analyseret og sammenlignet med andre kunder. Systemer med adgang til bankers pengetransaktioner ville forholdsmæssigt nemt i realtid kunne foreslå en kunde alternative udbydere af en vare, hun er i gang med at købe; eller de vil kunne danne ad-hoc indkøbsklubber, der automatisk starter en indkøbsauktion for varen. Umiddelbart er det dog ikke tilladt for en finansiel virksomhed.

2 Privacy

Big Data-anvendelser bygger nemlig typisk på personfølsomme oplysninger, og her kolliderer Big Data med beskyttelsen af privacy (da. *privatlivets fred*): Brystkræftanalyse kræver adgang til individuelle patienters sundhedsdata såsom sygdomsforløb, DNA-profiler, røntgen- og MR-billeder. Det ville fremme sygdomsforskningen, hvis dette materiale frit kunne postes på *World Wide Web*, men der er næppe mange patienter, som gerne vil se deres sygdomsdata lagt ud, så alle kan se dem i al fremtid.

Privacy er retten til være i fred. Privacy opnås ultimativt, når kun ejeren af personspecifikke data kan håndtere dem: Ingen anden kan vide, om de overhovedet eksisterer. Det betyder i yderste konsekvens, at dataene ikke kan have nogen direkte eller indirekte konsekvenser, der er observerbare for andre end ejeren. En insisteren på denne ekstreme form for privacy ville umuliggøre stort set alle Big Data-anvendelser inden for sundhed og finans. Udfordringen er derfor at finde en balance mellem Big Data og privacy: at opnå fordelene ved specifikke Big Data-anvendelser, dog med så lidt privacy-"lækage" som muligt. Hvordan kan en høj grad af privacy opnås og, hvis muligt, garanteres?

2.1 The Bad

De dårlige nyheder først: Privacy er meget svært at opnå.

2.1.1 Anonymisering

I en Big Data-verden med mange store datasæt er anonymisering, hvor personidentificerende oplysninger fjernes, utilstrækkelig; de kan ofte rekonstrueres fra andre oplysninger. En browser overfører f.eks. selv i *private mode* så mange attributer til en webserver, at de oftest er tilstrækkelige til at identificere brugeren — det kan testes på hjemmesiden for forskningsprojektet *Panopticlick* — og følge alle hendes forskellige netbesøg. Det ville være forholdsvist nemt for f.eks. Facebook at vise målrettede reklamer for hash, selv når hun tidligere kun har søgt efter det i private mode. (Når det ikke gøres, er det nok udelukkende,

fordi hun ville opleve det som ubehageligt.) Selv uden sådanne attributer kan anonymisering brydes ved at kombinere flere datasæt og krydskorrelere deres indbyrdes relationer. I den famøse Netflix-challenge blev alle brugerdata erstattet af tilfældige tal, men ved at sammenligne hvilke film disse brugere kunne lide med blogindlæg og andre datakilder kunne en del af brugernes identitet afsløres. I et andet eksempel blev patientjournalen tilhørende guvernøren af Massachusetts identificeret blandt anonymiserede patientdata.

2.1.2 Kryptering

Kryptering er en teknik, der skal sikre dataintegritet, -autenticitet og -privacy. Men kryptering løser ikke problemet med, hvordan man lækker *lidt* privacy uden at risikere at lække alt. Kryptering er groft sagt det samme som et pengeskab med hjul på. Man kan låse sine data inde og herefter placere skabet vilkårlige steder, uden at nogen uden nøgle kan få adgang til dataene. Men hvis de skal bruges til noget — og det skal de jo, ellers ville det være endu sikrere er slette dem med det samme — så er eneste måde at få adgang til data at låse hele pengeskabet op, hvorefter alle data i alle detaljer er synlige for hvem end måtte være til stede. En kendt sikkerhedsforsker fra Harvard University formulerede det således: "Kryptering er en mur, der er 6 meter høj og 2 meter tyk — og 3 meter bred. Man kan ofte bare gå uden om den." En bruger kan kun læse en krypteret besked ved at dekryptere den i sin helhed, og herefter kan hun i princippet sende den i klartekst videre som email. Efter at have låst data ind og ud af pengeskabet, er vi altså principielt tilbage ved udgangspunktet.

2.2 The Good

Det er derfor nødvendigt at rette blikket ikke kun mod data, men mod den *software*, der får adgang til fortrolige data: Hvilke programmer har adgang til fortrolige data, hvordan behandler de dem, hvilke resultater sender de videre og til hvem? Ikke alle programmer har de samme privacy-egenskaber. Et program der offentliggør, hvor mange kreditkortransaktioner der er blevet gennemført i en forretning, lækker meget lidt information om de bagvedliggende transaktioner i forhold til et program, der viser alle individuelle transaktioner på nettet.

Her er nogle forsigtigt optimistiske nyheder.

2.2.1 Software-baseret sikkerhed

Vi kan starte med at bruge effektive softwaresikkerhedsmetoder:

- Afvikl software i en "sandkasse", dvs. under et andet programs kontrol, der holder øje med afviklingen og stopper eller ændrer den, når noget suspekt sker. Det er meget brugt i *cloud computing*. Problemet er, at det kun fanger ret grovkornede sikkerheds- og privacyproblemer. Det vil ikke se forskel mellem et program, der viser pengetransaktionernes gennemsnit, og et, der viser selve pengetransaktionerne.

- Eliminer softwarefejl. Mange indbrud i serversystemer, såsom tyveri af kreditkortoplysninger, skyldes snedig udnyttelse af programmeringsfejl, der er svært at opdage ved afprøvning, f.eks. *buffer overflows*, *race conditions* i samtidigt kørende processer og utilsigtet udførsel af inputdata som programtekst (*SQL injection* og *cross-site scripting*-angreb). Disse fejl kan i deres helhed elimineres ved brugen af programmeringssprog med "stærke" typesystemer og/eller statisk sikrede domænespecifikke sprog og biblioteker. Det er overraskende, at sådanne teknikker ikke bruges i større omfang i nuværende praksis, da brud i sikkerhed kan have hurtige og omfattende negative konsekvenser. Som vicedirektøren for Microsoft India formulerede det engang: "In the internet age, the worst case is the average case; a single obscure security hole can spread like wildfire and be exploited instantaneously." Det er endvidere svært at koble sikkerhed på efterfølgende – den skal være tænkt ind i softwarekonstruktionen fra begyndelsen.

- Brug sprogbaseret sikkerhed (eng. *language-based security, LBS*). Det dækker over metoder, der med matematikkens og logikkens magt garanterer, at programmer opfylder bestemte sikkerhedsegenskaber, før de eksekveres. I *proof-carrying code (PCC)* ledsages softwaren af et logisk certifikat, der garanterer det; i nogle domænespecifikke sprog (eng. *domain-specific languages, DSL*) er programmeringsrepertoiret begrænset til kun at konstruere sikre programmer.

2.2.2 Software-baseret privacy

Men selv efter sikkerhedsfejl er rettet, er der intet til hinder for, at programmer stadigvæk lækker mere privacy end tilsigtet. Her er vi ude i spændende teknikker, som kan delvist kan betragtes som forskningsmæssigt modnet, men som endnu ikke er standardpraksis i alle udviklingsafdelinger.

- I *Secure Multiparty Computation (SMC)* har flere aktører private data, de ikke vil udlevere til hinanden, men som skal bruges til at beregne et fælles resultat. Et illustrerende eksempel er, når to millionærer vil afgøre, hvem af dem, der er rigere uden at oplyse hinanden eller andre om, hvor mange penge de hver især har.

- I *Information Flow Analysis* analyseres programmer, der er afhængige af private og offentlige data, til at sikre, at deres offentlige uddata hverken direkte eller indirekte afslører private inddata; det garanterer således hemmeligholdelse af alle private data.

- *Differential Privacy* dækker over metoder, der understøtter statistiske beregninger uden at lække mere end det minimale om individuelle private data, de afhænger af. Dette gøres ved at beregne resultatet, måle hvor meget resultatet ændrer værdi afhængig af ændring af private inddata og ved at føje tilstrækkelig meget statistisk "støj" til resultatet for at skjule de individuelle inddatas bidrag.

- *Data Provenance* dækker over metoder til at repræsentere og bevare, hvor data kommer fra. Her indfarves data konceptionelt med oplysninger om, hvor de stammer fra, og programberegninger implementeres til at bevare farverne af inddata i deres uddata. På denne måde gengiver farvemikset i resultatet, hvilke data der er blevet brugt til at beregne det, inden man sender det videre.

Der er en del uløste problemer med at få disse metoder udviklet til praksis. Blandt andet kræves sikre og praktisk effektive softwarekonstruktionsmetoder og -teknikker, der understøtter Big Data-applikationsudvikling.

2.3 The Ugly

En ikke ubetydelig risiko for fremskridt i denne retning udgøres af juridisk regulering i omgangen med personspecifikke oplysninger, hvis denne ikke bruger robuste koncepter eller ignorerer grundlæggende *trade-offs* mellem privacy and dataudnyttelse. Hvis f.eks. "videregivelse af oplysninger" i *lov om finansiel virksomhed* fortolkes som forbud mod direkte eller indirekte videregivelse af så meget som *en enkelt bit* af informationen i de sensitive data, så kan interessante Big Data-anvendelser kun realiseres af ikke-finansielle virksomheder. Relaterede risici opstår, hvis lovgivningen regulerer beskyttelse af bestemte data i stedet for den information, de repræsenterer; eller hvis programoptimeringsmetoder såsom *caching* (midlertidig lagring af data eller beregningsresultater) underkastes kunstige juridiske klassifikationer som set tidligere i forbindelse med ophavsretslovgivningen.

3 Konklusion

Der er gode grunde til at tro, at Big Data og hensyn til privacy kan forliges med hinanden, hvis Big Data, privacy og regulering samtænkes i stedet for at angribe dem isoleret fra hinanden.