Come le macchine imparano ad apprendere.

Liceo Scientifico Statale E. Fermi

02/04/2019

fabrizio.malfanti@kellify.com

Machine Learning

- Cosa significa ?
- Cosa imparano le macchine ?
- Come imparano le macchine ?
- Quanto imparano le macchine ?
- Come si danno i voti alle macchine ?

Data Mining

- **Il processo di estrazione di informazione valida, utilizzabile e precedentemente sconosciuta, da grandi basi di dati e l'utilizzo di queste informazioni per prendere cruciali informazioni di business.**

Cosa è il Machine Learning

(o apprendimento automatico)

- Il Machine Learning coinvolge gran parte delle nostre abitudini:
- Spotify che ci suggerisce che brani ascoltare
- Netflix che ci consiglia un film
- Amazon che ci mostra I prodotti che ci possono essere utili
- Instagram che ordina gli aggiornamenti da mostrarci
- Facebook che ci suggerisce cosa vedere, cosa leggere, cosa ascoltare, quali amici considerare ecc ecc ecc

- Ma anche:
- L’algoritmo che costruisce I piani di cura dei malati oncologici in base all’analisi istologica
- I veicoli che si guidano da soli
- L’algoritmo che vi guida nel traffico
- Il sistema che ottimizza la frenata del nostro veicolo
- L’algoritmo che ordina l’accesso alla sala operatoria per ottimizzare la riuscita degli interventi
- L’algoritmo che riordina il magazzino di Amazon per permettere una delivery più efficiente e veloce

Algoritmi

- **Un algoritmo è una sequenza di istruzioni che dice a un computer cosa fare.**
- **Non è solo un insieme di istruzioni, deve essere talmente preciso, dettagliato e univoco da poter essere eseguito da un computer.**
- **Una ricetta di cucina non è un algoritmo, non è abbastanza preciso e dà per scontate una serie di conoscenze che il computer non può avere.**
- Seguire una ricetta porta a piatti squisiti o a schifezze immani, seguire un algoritmo porta sempre allo stesso risultato

TRIS

- Se l'avversario ha occupato due caselle di fila, occupate la casella mancante
- Altrimenti, se c'è una mossa che crea due coppie di caselle adiacenti in un colpo solo scegliete quella
- Altrimenti, se la casella centrale è libera occupate quella
- Altrimenti, se l'avversario ha occupato una casella d'angolo occupate quella opposta
- Altrimenti, se c'è un angolo vuoto , occupatelo
- Altrimenti occupate una casella vuota qualsiasi

NON PERDE MAI

Algoritmi

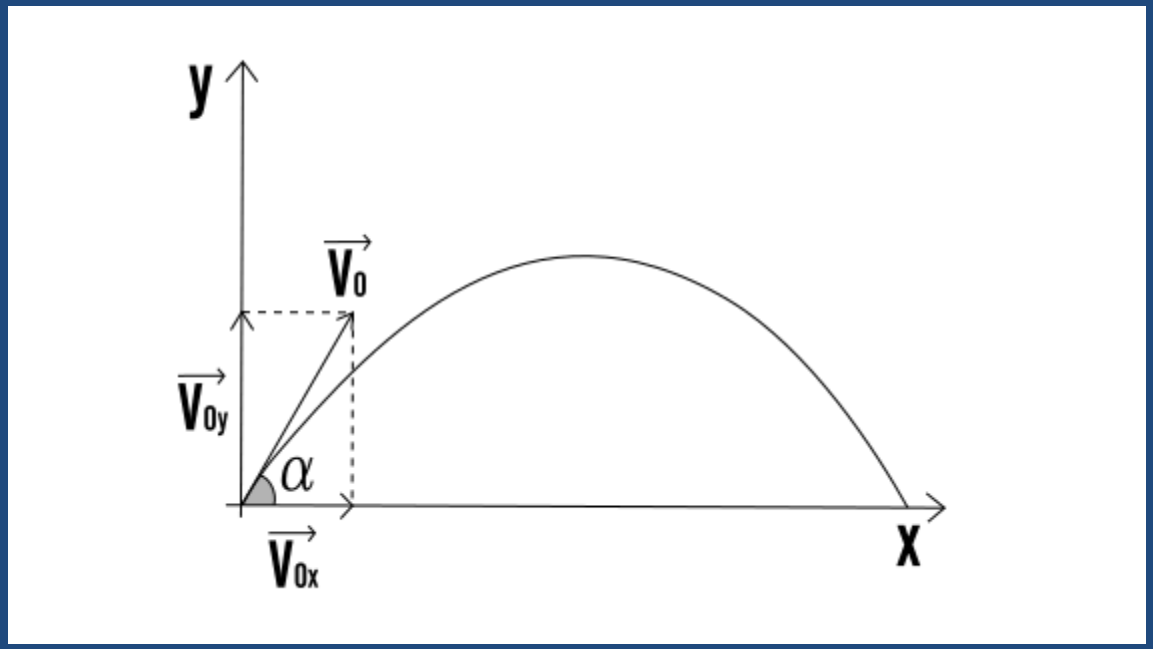
Ogni algoritmo ha un input e un output: Si inseriscono i dati nel computer, si fa girare l'algoritmo e si ottiene un risultato.

L'algoritmo è scritto da un programmatore che implementa su un computer la sequenza di istruzioni.

Moto del proiettile

Se dobbiamo lanciare un sasso e centrare un buco e dobbiamo far fare il lavoro a un computer, possiamo impostare un algoritmo per calcolare la gittata in funzione dell'angolo di lancio e della velocità iniziale.

Moto del proiettile



Moto del proiettile

Questa è la formula della traiettoria

$$\begin{cases} t = \dfrac{x - x_0}{v_{0x}} \\ y = -\dfrac{g}{2v_0^2\cos^2(\alpha)}x^2 + \left(\dfrac{gx_0}{v_0^2\cos^2(\alpha)} + \tan(\alpha)\right)x - \dfrac{gx_0^2}{2v_0^2\cos^2(\alpha)} - \tan(\alpha)x_0 + y_0 \end{cases}$$

Moto del proiettile

E questa è la formula della gittata:

$$x_G = \frac{2v_0^2 \cos(\alpha) \sin(\alpha)}{g}$$

Moto del proiettile

Per quanto la formula possa essere complicate è possibile sviluppare un algoritmo:

$$x_G = \frac{2v_0^2 \cos(\alpha) \sin(\alpha)}{g}$$

Se devo mandare un oggetto a 100 metri di distanza, se l'angolo di lancio è di 30°, la velocità iniziale deve essere di 33 m/s.

Moto del proiettile

Se poi tengo conto del vento, di altre possibili interferenze, della forma dell'oggetto che lancio, dell'attrito dell'aria ecc ecc ecc, la formula diventa sempre più complicata, ma ci sarà sempre un modo per trasformarla in un algoritmo da implementare su un computer.

PERO’ …

Moto del proiettile

Un bambino di 6 anni riesce a centrare una buca con una biglia da 5 metri di distanza e non ha la minima idea di quanto valga *g* (ne che esista) e di cosa sia il cos(α).

Principio del Machine Learning

E’ evidente che c’è un altro modo di imparare le cose diverso dall’implementare un algoritmo, che è il modo in cui noi, da bambini all’età adulta, impariamo a fare le cose senza sapere precisamente come le facciamo.

Principio del Machine Learning

Questo è il principio di base del Machine Learning. Anziché programmare un computer con un algoritmo che prendo dei dati in input, fa i conti e produce un output, noi programmiamo i computer affinché imparino dai dati a effettuare un compito.

Principio del Machine Learning

Il Machine Learning è quello che succede quando si insegna ai computer ad essere creativi.

Principio del Machine Learning

In questo modo il computer costruisce il suo algoritmo, a partire dai dati di input e dai risultati.

In questo modo si possono implementare processi che non sarebbero convertibili in algoritmi perché troppo complicati.

Principio del Machine Learning

Nessun programmatore saprebbe scrivere un algoritmo per guidare l'auto, ci sono troppe variabili in gioco, troppi elementi da considerare e troppi eventi da gestire.

Ma sono invece riusciti a costruire un computer a cui dire: «Impara a guidare guardando quello che facciamo noi»

Principio del Machine Learning

Insegnare a un programma ha diversi vantaggi:

- Se impara qualcosa di sbagliato posso spegnerlo e iniziare da zero
- Una volta che il programma è addestrato lo posso replicare in migliaia di esemplari in tempo zero
- Non si dimentica quello che ha imparato

Machine Learning

Cosa può imparare un computer e come ?

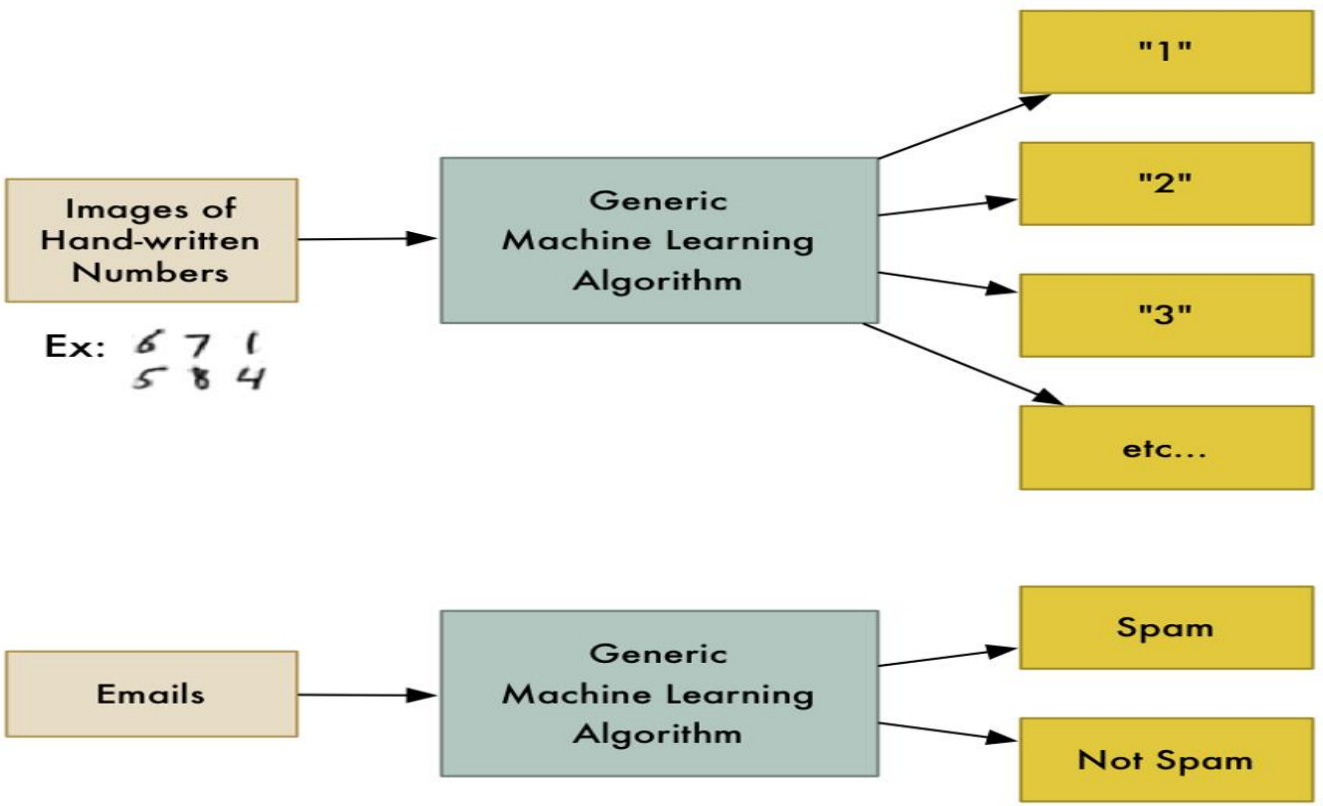
- Poche cose e in pochi modi diversi.
- Riesce a capire a che classe appartiene un oggetto/immagine/video o a che valori può corrispondere.
- Riesce a individuare gli oggetti che si assomigliano
- Riesce a individuare le mosse successive da fare.

Classificazione

Un algoritmo di classificazione può inserire i dati in diversi gruppi basandosi sulle caratteristiche in comune che è in grado di determinare in autonomia.

Lo stesso algoritmo si usa per classificare e identificare cose diverse, dai numeri scritti a mano al fatto che una mail sia spam o non spam.

Classificazione



Classificazione

Sei un agente immobiliare vuoi migliorare la funzionalità della tua agenzia. Puoi immagazzinare i dati di tutti le case che hai venduto negli ultimi tempi e costruire un software che ti aiuta a stimare il prezzo delle nuove case.

Classificazione

Vani	Mq	Quartiere	piano	Posto auto	Prezzo
6	100	Sampierdarena	5	si	250000
5	80	Sampierdarena	6	no	180000
9	180	Castelletto	4	si	400000
4	50	Rivarolo	1	No	100000
8	160	Castelletto	2	No	300000
10	220	Nervi	6	SI	600000

Usando questa tabella di dati vogliamo costruire un sistema che sia in grado di prevedere il prezzo di qualunque altra casa a Genova

Classificazione

Vani	Mq	Quartiere	piano	Posto auto	Prezzo
7	120	Sampierdarena	3	no	?
5	80	Castelletto	6	no	?

Classificazione

E’ come avere un foglio con gli esercizi di algebra e le risposte dove tutti i simboli matematici sono stati cancellati

Math Quiz #1 - Teacher's Answer Key

1) 2 4 5 = 3
2) 5 2 8 = 2
3) 2 2 1 = 3
4) 4 2 2 = 6
5) 6 2 2 = 10
6) 3 1 1 = 2
7) 5 3 4 = 11
8) 1 8 1 = 7

Classificazione

E dover risolvere il problema:

23 12 7 = ?

Classificazione

Possiamo davvero considerare la capacità di calcolare il prezzo di una casa come **Apprendimento** ?.

Classificazione

Negli esseri umani, il cervello può approcciare qualsiasi situazione e imparare come farvi fronte senza istruzioni esplicite. Se vendi case per lungo tempo, sicuramente avrai la “sensazione” sul giusto prezzo per la casa, sul modo migliore per commercializzarla, sul tipo di cliente che potrebbe essere interessato, ecc.

L’obiettivo della ricerca sull’Intelligenza Artificiale è quello di essere in grado di replicare questa capacità nei computer.

Classificazione

Ad oggi, gli algoritmi di Machine Learning non sono ancora così avanzati — funzionano solo quando hanno di fronte un problema molto specifico, limitato.

In questo caso, forse una definizione più appropriata di "apprendimento" sarebbe "**capire un'equazione, per risolvere un problema specifico, sulla base di alcuni dati di esempio**".

Classificazione

Purtroppo *“Automazione per capire un’equazione che risolve un problema specifico, sulla base di alcuni dati di esempio”* non è un grande bel nome.

Così ci siamo ritrovati invece con il nome **Machine Learning**.

House Pricing

Proviamo a scrivere questo algoritmo.

Senza il machine learning potrei pensare di fare così:

House Pricing

Vani	Mq	quartiere	piano	Posto auto	Prezzo
6	100	Sampierdarena	5	si	180000
5	80	Sampierdarena	6	no	140000
9	180	Castelletto	4	si	400000
4	50	Voltri	1	No	60000
8	160	Castelletto	2	No	300000
10	220	Nervi	6	SI	600000

House Pricing

```
def stima_prezzo_casa(vani, mq2, quartiere,piano,posto auto):
prezzo = 0
# a sampierdarena il costo medio delle case per mq2 è € 2388
prezzo_per_mq2 = 1777
if quartiere== "sampierdarena":
# ma alcune aree costano un pò di più
prezzo_per_mq2 = 2058
elif quartiere== "castelletto":
prezzo_per_mq2 = 2727
elif quartiere== "nervi":
# alcune aree costano un pò di meno
prezzo_per_mq2 = 1400
# si inizia a stimare un prezzo base in base alla grandezza della casa
prezzo = prezzo_per_mq2 * mq2
# adesso aggiusti la stima in base al numero di vani
if num_di_vani == 4:
# i 4 vani  sono economici
prezzo= prezzo — 10000
else:
# casa con più vani viene valutate di più
prezzo = prezzo + (num_di_vani* 1000)
Aggiungo altre regole per piano, posto auto ….
return prezzo
```

House Pricing

- Puoi spenderci sopra ore e ore, e arrivare alla fine ad avere qualcosa che funzioni. Ma il programma non sarà mai perfetto e sarà difficile mantenerlo aggiornato dato che i prezzi cambiano in continuazione.
- Non sarebbe meglio se il computer riuscisse a capire come implementare questa funzione al posto tuo? A noi non importa cosa fa esattamente la funzione, ma che dia come risultato il valore corretto:

House Pricing

```
def stima_prezzo_casa(num_di_vani, mq2, quartiere,piano,posto auto):
prezzo = <computer, per favore fai qualche calcolo per me>
return prezzo
```

House Pricing

- Potete pensare a questo problema come se il **prezzo** fosse un delizioso stufato e gli ingredienti fossero il **numero di vani**, la **metratura,** il **quartiere,** il **piano** e la presenza di un **posto auto**. Se riuscite a capire quanto ogni ingrediente impatta sul prezzo finale, probabilmente c'è un rapporto esatto tra gli ingredienti e il prezzo finale.
- Questo ridurrebbe la funzione originale (con tutte quei 'se' e 'altrimenti') a qualcosa di molto più semplice come questo:

House Pricing

```
def stima_prezzo_casa(num_di_vani, mq2, quartiere,piano,posto auto):
prezzo = 0
# un pizzico di questo
prezzo += num_di_vani* 0.841231951398213
# una grande manciata di questo
prezzo += mq2 * 1231.1231231
# forse un pugnetto di questo
prezzo += quartiere * 2.3242341421
# e un pugnetto di questo
prezzo += piano * 1.12334200
# e forse un pugnetto di questo (un PI GRECO non fa mai male)
prezzo += (posto auto) * 3.14159265358979323846264338327950288419 71
# e alla fine, ci aggiungiamo un pò di sale qb
prezzo += 201.23432095
return prezzo
```

House Pricing

- I numeri magici in grassetto **0.841231951398213**, **1231.1231231**, **2.3242341421**, **1.12334200 , 3.14159265358979323846264338327950288 41971** e **201.23432095** sono i nostri **pesi**. Se fossimo in grado di capire i pesi perfetti per usarli per ogni casa, la nostra funzione sarebbe in grado di predire i prezzi delle case!

House Pricing

- Un semplice modo per capire i migliori pesi da usare sarebbe qualcosa di questo tipo:

House Pricing

Step 1:
Inizia con ogni peso impostato a 1.0:

```
def stima_prezzo_casa(num_di_vani, mq2, quartiere,piano,posto auto):
prezzo= 0
# un pizzico di questo
prezzo += num_di_vani* 1.0
# una grande manciata di questo
prezzo += mq2 * 1.0
# forse un pugnetto di questo
prezzo += quartiere * 1.0
# e un pugnetto di questo
prezzo += piano * 1.0
# e forse un pugnetto di questo (un PI GRECO non fa mai male)
prezzo += (posto auto) * 1.0

# e alla fine, ci aggiungiamo un pò di sale qb
prezzo += 1.0
return prezzo
```

House Pricing

Step 2:

Usa la funzione per ogni casa che conosci e guarda quanto lontano si scosta il risultato della funzione dal prezzo corretto reale:

House Pricing

Vani	Mq	quartiere	piano	Posto auto	Prezzo	Prezzo Calcolato
6	100	Sampierdarena	5	si	180000	200000
5	80	Sampierdarena	6	no	140000	135000
9	180	Castelletto	4	si	400000	420000
4	50	Voltri	1	No	60000	72000
8	160	Castelletto	2	No	300000	320000
10	220	Nervi	6	SI	600000	550000

House Pricing

Step 3:

Ripeti il secondo passaggio più e più volte con **ogni possibile combinazione di pesi**. Qualunque sia la combinazione di pesi che rende il costo più vicino allo zero, è quella da usare. Quando trovi il peso che funziona, hai risolto il problema!

House Pricing

Quello che abbiamo fatto è banale, abbiamo semplicemente alimentato 5 insiemi di numeri e abbiamo costruito l'algoritmo per calcolare il prezzo di una casa a partire da alcune sue caratteristiche.

House Pricing

La ricerca in molti campi (come la linguistica / la traduzione) negli ultimi 40 anni ha dimostrato che questi algoritmi di apprendimento generici che “mescolano i numeri del minestrone” hanno approcci migliori di quelli usati dalle persone reali che usano esplicite regole. L’approccio “sciocco” del Machine Learning alla fine batte gli esperti umani.

House Pricing

La funzione che abbiamo ottenuto è molto sciocca. Non sa nemmeno cosa sono i metri quadrati o i vani ne tantomeno conosce Genova. Tutto quello che sa è che ha bisogno di mescolare questi numeri per ottenere la risposta corretta.

House Pricing

Immaginate che invece di prendere come parametri i metri quadri e vani, la funzione prenda un range di numeri.

Diciamo che ogni numero rappresenta la luminosità di un pixel di un'immagine catturata dalla telecamera montata sulla parte superiore della vostra auto. Ora diciamo che invece di cercare un valore chiamato "prezzo", la funzione emetta una previsione chiamata "gradi_per_girare_sterzo_ruote".

Avete appena fatto una funzione che può guidare l'auto da sola!

House Pricing

La ricerca in molti campi (come la linguistica / la traduzione) negli ultimi 40 anni ha dimostrato che questi algoritmi di apprendimento generici che “mescolano i numeri del minestrone” hanno approcci migliori di quelli usati dalle persone reali che usano esplicite regole. L’approccio “sciocco” del Machine Learning alla fine batte gli esperti umani.

House Pricing

Che dire sul fatto di "provare ogni numero" indicato nello Step 3?

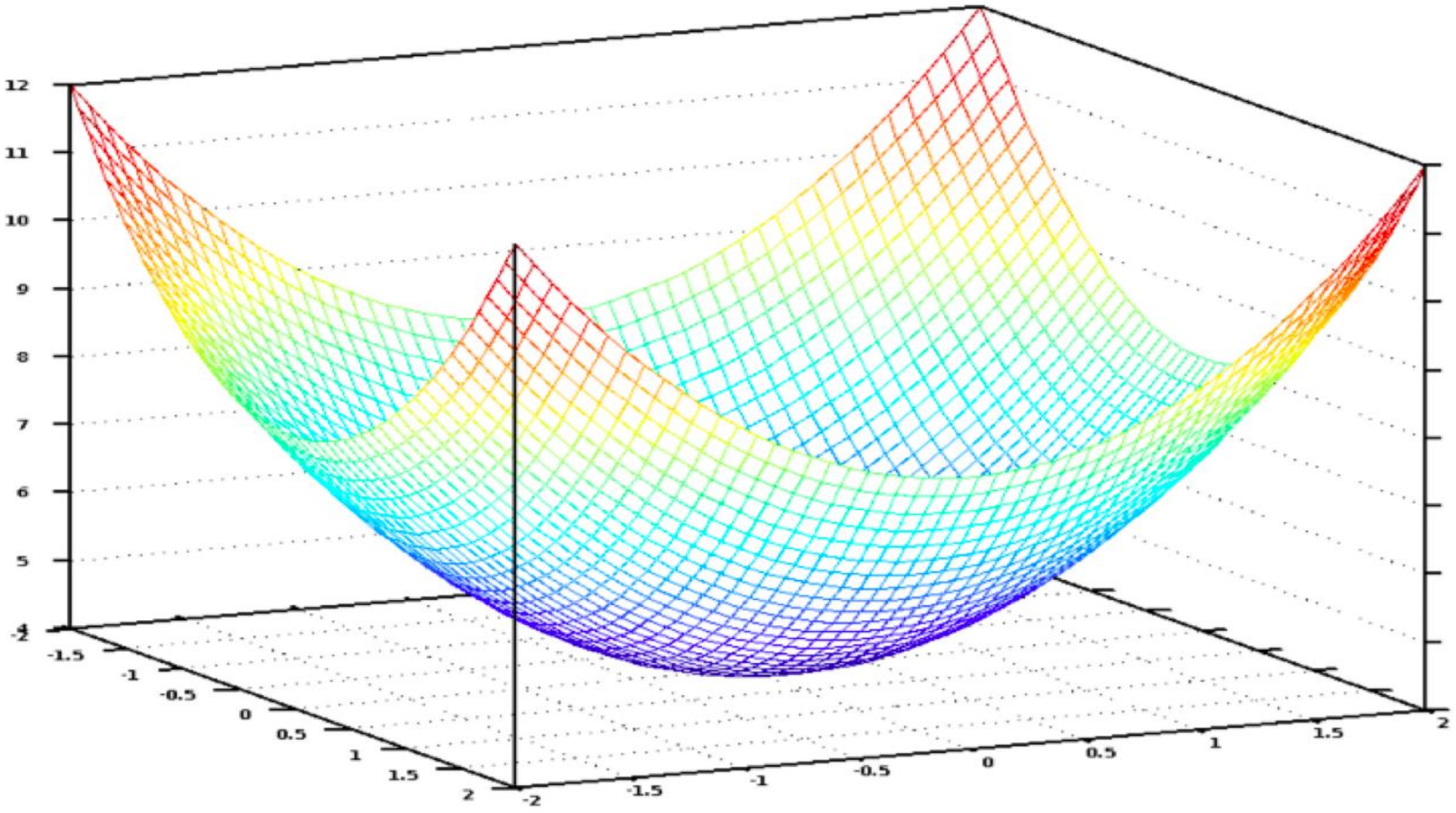
Ok, ovviamente non possiamo provare tutte le combinazioni di tutti i pesi possibili per trovare la combinazione che funziona meglio. Se lo prendiamo alla lettera continueremmo all'infinito perché non saremmo mai a corto di numeri da provare.

Per evitare questo, i matematici hanno trovato molti modi intelligenti per trovare rapidamente dei buoni valori per quei pesi senza dover fare tantissimi tentativi.

House Pricing

$$\text{Cost} = \frac{\sum_{i=1}^{500} \left(\text{MyGuess}(i) - \text{RealAnswer}(i)\right)^2}{500 \cdot 2}$$

House Pricing



House Pricing

In questo grafico, il punto più basso in blu è dove il nostro costo è il più basso — quindi la nostra funzione sarà la meno sbagliata. I punti più alti sono dove ci sono margini di errore più alti. Quindi, se siamo in grado di trovare i pesi che ci fanno arrivare al punto più basso di questo grafico, il gioco è fatto!

House Pricing

Abbiamo solo bisogno di regolare i nostri pesi per fare in modo di spostarci verso la parte centrale, più bassa del grafico. Se continuiamo a fare piccoli aggiustamenti sui nostri pesi rimanendo diretti verso il punto più basso, arriveremo alla meta senza dover provare tanti altri pesi.

House Pricing

L'algoritmo in tre fasi che ho descritto sopra si chiama **regressione lineare multivariata**. Sta stimando l'equazione di una linea che si adatta a tutti i punti dei dati delle casa. Quando utilizziamo questa equazione per stimare il prezzo di vendita delle case, non sappiamo bene dove apparirà sulla linea. Questa è davvero un'idea molto potente con cui si riescono a risolvere i problemi "reali".

House Pricing

Abbiamo solo bisogno di regolare i nostri pesi per fare in modo di spostarci verso la parte centrale, più bassa del grafico. Se continuiamo a fare piccoli aggiustamenti sui nostri pesi rimanendo diretti verso il punto più basso, arriveremo alla meta senza dover provare tanti altri pesi.

House Pricing

- L'approccio che ho mostrato può funzionare nei casi più semplici, ma non in tutti i casi. Una ragione è che i prezzi delle case non sono sempre abbastanza semplici e lineari da seguire una linea continua.
- Ma per fortuna ci sono molti modi per gestire questo problema. Ci sono molti altri algoritmi di "Machine Learning" in grado di gestire i dati non lineari

Il Machine Learning è magico ?

Una volta che si inizia a vedere la facilità con cui tecniche di Machine Learning possono essere applicate a problemi che sembrano davvero difficili (come il riconoscimento della scrittura a mano), si inizia ad avere la sensazione che si potrebbe usare il Machine Learning per risolvere qualsiasi problema e ottenere una risposta fino a quando si hanno abbastanza dati. Basta inserire dei dati e guardare il computer calcolare magicamente l'equazione che c'è tra gli stessi!

Il Machine Learning è magico ?

- Ma è importante ricordare che il Machine Learning funziona solo se il problema è veramente risolvibile con i dati che si hanno a disposizione.
- Ad esempio, un modello che prevede i prezzi delle case in base al tipo di piante in vaso di ogni casa, non potrà mai funzionare. Non c'è nessuna relazione tra le piante in vaso in ogni casa e il prezzo di vendita della casa. Quindi, non importa quanto si provi, il computer non potrà dedurre una relazione tra i due.

Il Machine Learning è magico ?

Quindi ricordate, se un umano esperto non può utilizzare i dati per risolvere il problema manualmente, anche un computer probabilmente non sarà in grado di risolverlo. Invece, concentratevi su problemi che un essere umano è in grado di risolvere, ma dove sarebbe meglio se un computer potesse risolverli molto più velocemente.

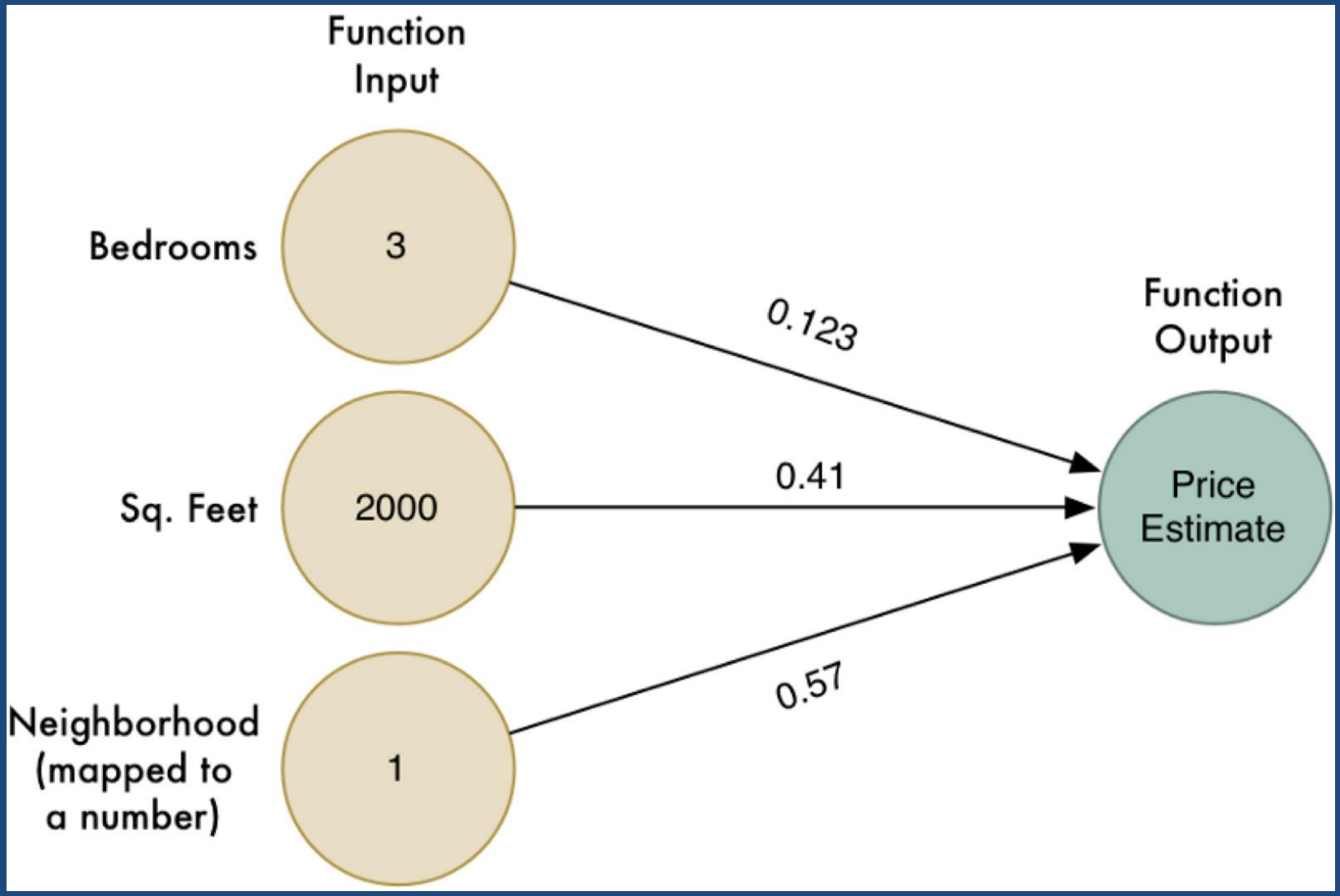
House Pricing

- Torniamo ancora per un attimo al problema dei prezzi delle case.
- Semplifichiamolo ancora per rendere più facile i grafici.
- Diciamo che il prezzo della casa dipende dai metri quadri, dal numero di camere da letto e dal quartiere.

House Pricing

```
def stima_prezzo_casa(num_di_stanze_da_letto, mq2, quartiere):
prezzo= 0
# un pizzico di questo
prezzo += num_di_stanze_da_letto * 0.123
# una grande manciata di questo
prezzo += mq2 * 0.41
# forse un pugnetto di questo
prezzo += quartiere * 0.57
return prezzo
```

House Pricing



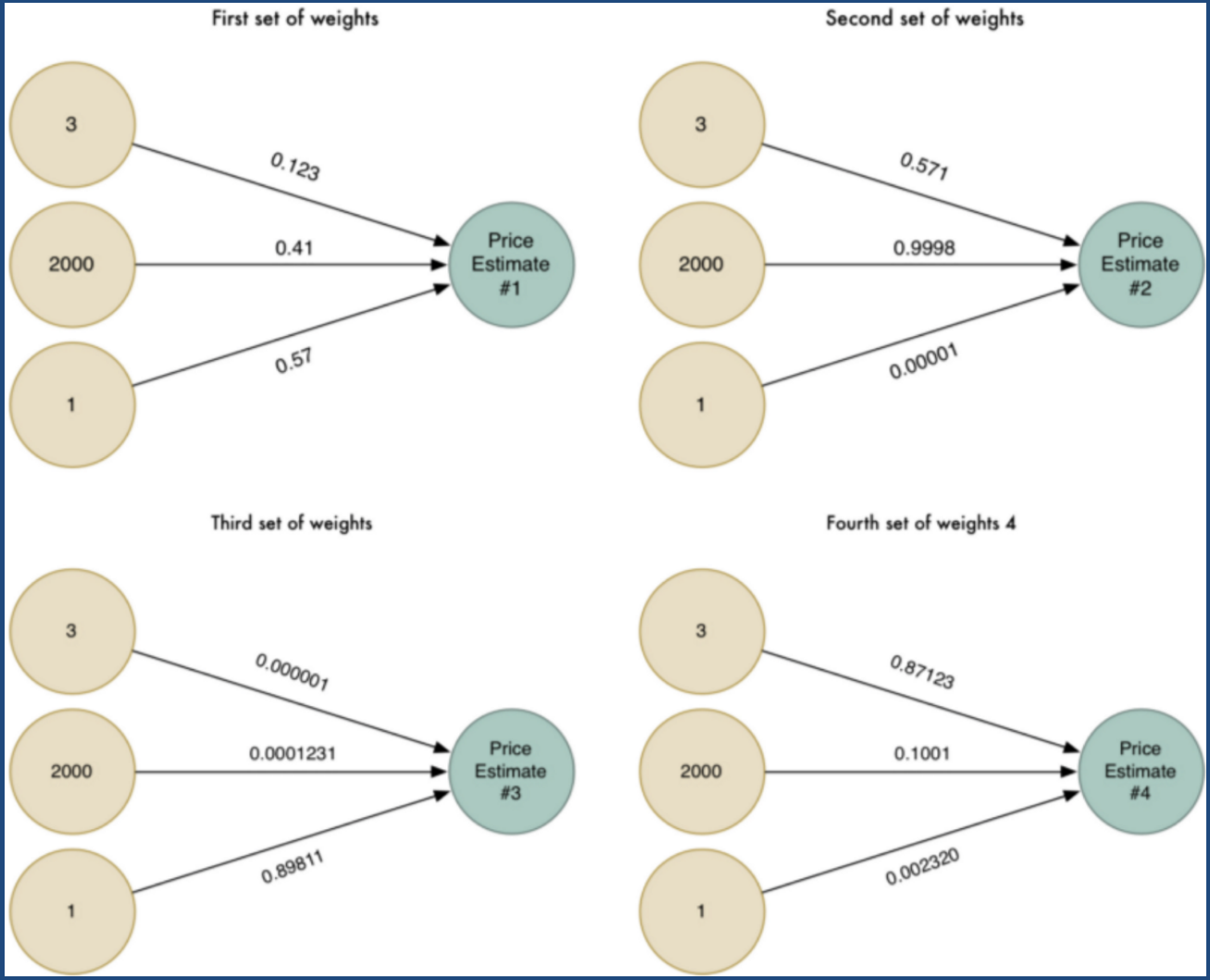
House Pricing

- Ma rendiamo la situazione più complicata, ad esempio possiamo pensare che il quartiere influisca sul prezzo finale solo per case grandi e non per quelle piccole e che ci siano quartieri dove esistono sia case costose che meno costose.

House Pricing

- Per risolvere il problema nuovo possiamo pensare di far girare il nostro programma più volte con valori diversi

House Pricing



House Pricing

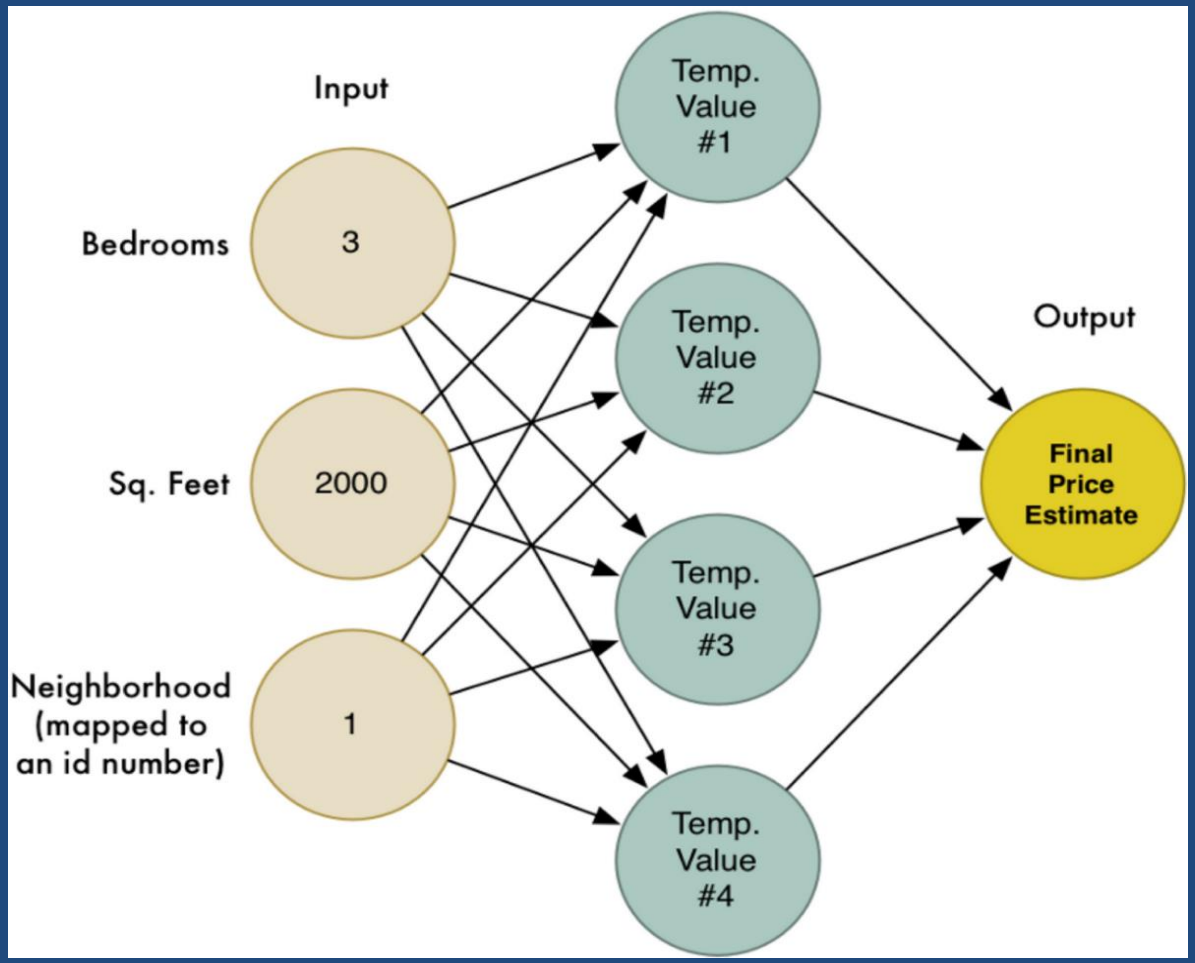
- Ora abbiamo quattro diverse stime. Cerchiamo di combinare questi quattro risultati per ottenere un unico prezzo finale. Inseriamoli di nuovo dentro allo stesso algoritmo (ma usando un'altra combinazione di pesi)!

House Pricing

House Pricing

- La nostra nuova *Super Risposta* combina le stime dei nostri quattro diversi tentativi di risolvere il problema. Grazie a questo, possiamo modellare più casi di quelli che avremmo potuto catturare con un modello semplice.

House Pricing



House Pricing

Questa è una rete neurale!

Ogni nodo sa come prendere un set di valori come input, applicare dei pesi ad essi, e calcolare un valore di output. Collegando insieme molti nodi come questi, possiamo modellare funzioni complesse.

House Pricing

- Abbiamo creato una semplice funzione di stima che prende una serie di input e li moltiplica per vari pesi per ottenere un output. Chiamiamo questa semplice funzione un **neurone**.
- Concatenando molti **neuroni** semplici insieme, siamo in grado di modellare le funzioni che sarebbero troppo complicate per essere gestite da un unico neurone.

Reti Neurali

Sappiamo che gli stessi algoritmi generici di Machine Learning possono essere riutilizzati con dati diversi per risolvere altri problemi.

Quindi cerchiamo di modificare questa stessa rete neurale per il riconoscimento del testo scritto a mano.

Ma per rendere il lavoro molto semplice, cercheremo solo di riconoscere il numero “8”.

Reti Neurali

Il Machine Learning funziona solo quando si dispone di dati — preferibilmente una grande quantità di dati. Quindi abbiamo bisogno di un grande numero di “8” scritti a mano.

Per fortuna, alcuni ricercatori hanno creato il dataset MNIST di numeri scritti a mano, una serie di numeri scritti a mano appunto per questo scopo. MNIST fornisce 60.000 immagini di cifre scritte a mano, ciascuno come immagine 18x18. Qui ci sono alcuni “8” presi dal dataset MNIST:

Reti Neurali

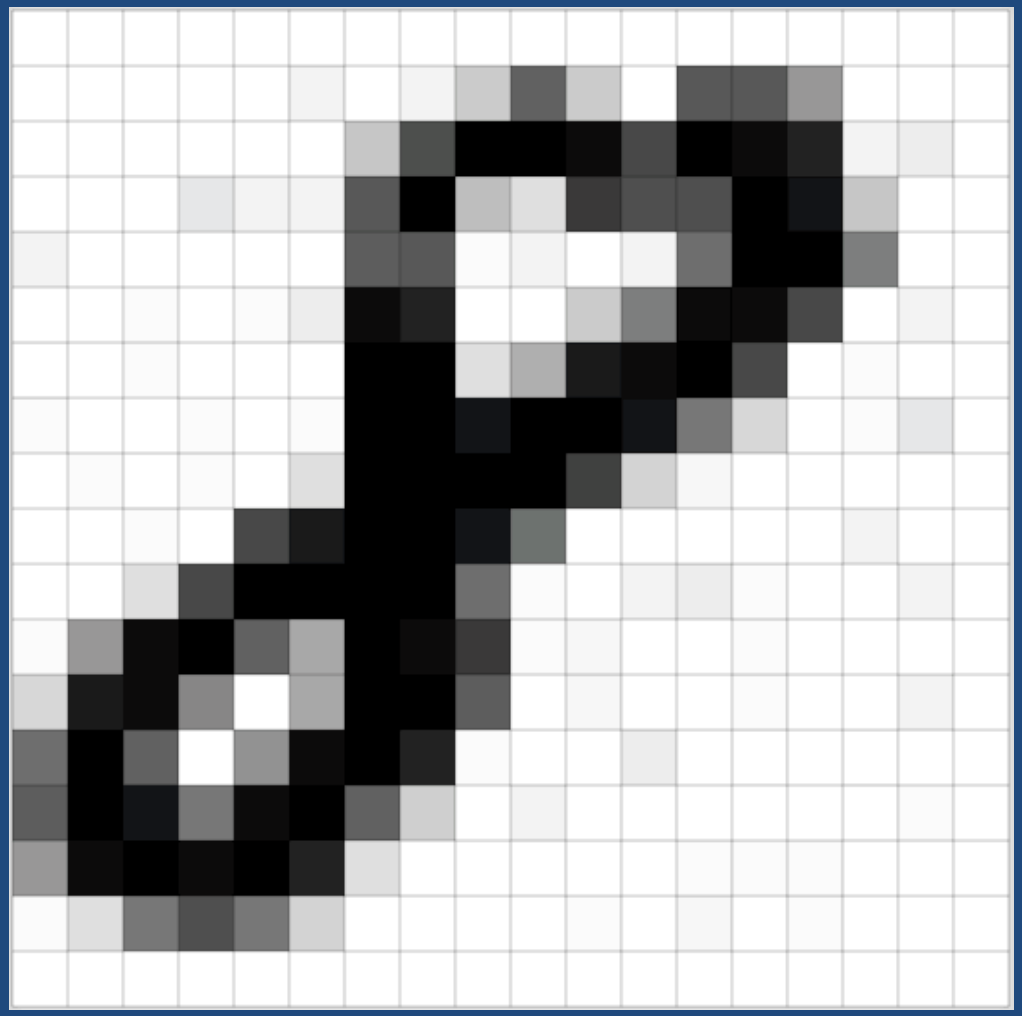
Reti Neurali

Alla fine dei conti, tutto è fatto di numeri

Il Neural Network che abbiamo costruito in precedenzaera in grado di prendere solo tre valori come input (“3” camere da letto, “2000” metri quadrati, etc.). Ma ora vogliamo usare il nostro Neural Network per elaborare delle immagini. Come facciamo ad alimentare la rete neurale usando immagini e non numeri?

La risposta è molto semplice. Una rete neurale prende solo numeri come input. Per un computer, l’immagine non è altro che una griglia di numeri che rappresentano il grado di oscurità di ogni pixel:

Reti Neurali



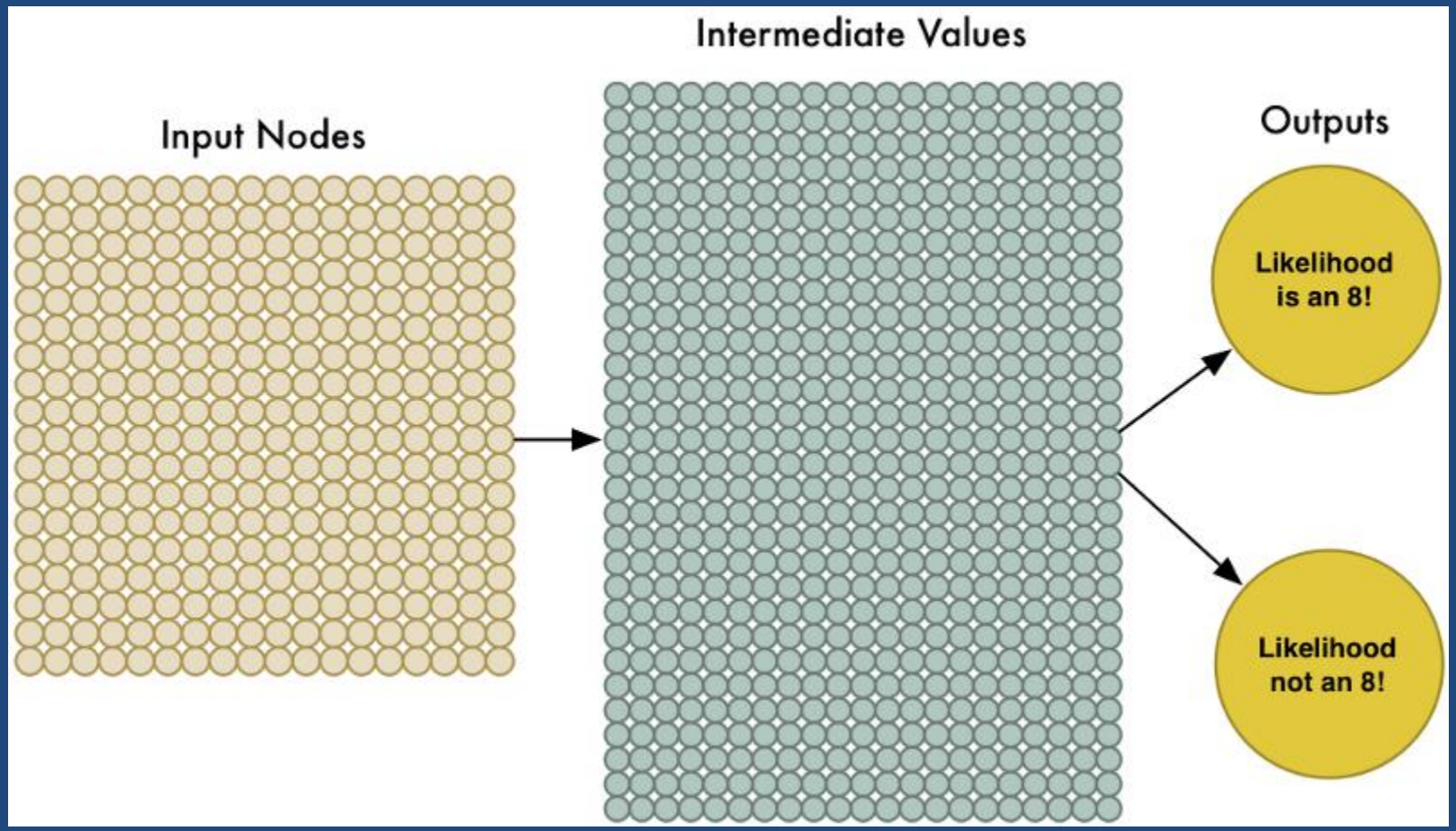
Reti Neurali

Per alimentare un'immagine nella nostra rete neurale, dobbiamo trasformare l'immagine 18x18 pixel in una matrice di 324 numeri:

=

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 12, 0, 11, 39, 137, 37, 0, 152, 147, 84, 0, 0, 0, 0,
0, 1, 0, 0, 0, 41, 160, 250, 255, 235, 162, 255, 238, 206, 11, 13, 0, 0, 0, 0, 16, 9, 9, 150, 251, 45, 21, 184, 159, 154, 2
55, 233, 40, 0, 0, 10, 0, 0, 0, 0, 0, 145, 146, 3, 10, 0, 11, 124, 253, 255, 107, 0, 0, 0, 0, 3, 0, 4, 15, 236, 216, 0, 0,
38, 109, 247, 240, 169, 0, 11, 0, 1, 0, 2, 0, 0, 0, 253, 253, 23, 62, 224, 241, 255, 164, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3, 252
, 250, 228, 255, 255, 234, 112, 28, 0, 2, 17, 0, 0, 2, 1, 4, 0, 21, 255, 253, 251, 255, 172, 31, 8, 0, 1, 0, 0, 0, 0, 0, 4,
0, 163, 225, 251, 255, 229, 120, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 21, 162, 255, 255, 254, 255, 126, 6, 0, 10, 14, 6, 0, 0, 9
, 0, 3, 79, 242, 255, 141, 66, 255, 245, 189, 7, 8, 0, 0, 5, 0, 0, 0, 0, 26, 221, 237, 98, 0, 67, 251, 255, 144, 0, 8, 0, 0
, 7, 0, 0, 11, 0, 125, 255, 141, 0, 87, 244, 255, 208, 3, 0, 0, 13, 0, 1, 0, 1, 0, 0, 145, 248, 228, 116, 235, 255, 141, 34
, 0, 11, 0, 1, 0, 0, 0, 1, 3, 0, 85, 237, 253, 246, 255, 210, 21, 1, 0, 1, 0, 0, 6, 2, 4, 0, 0, 0, 6, 23, 112, 157, 114, 32
, 0, 0, 0, 0, 2, 0, 8, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Reti Neurali



Reti Neurali

La rete neurale è più grande, ma il concetto è lo stesso. Adesso va addestrata con tanti dati, quelli presi dal database che dicevamo prima, facendogli sapere quali sono «8» e quali no.

Reti Neurali

le cose in realtà non sono così semplici!

In primo luogo, la buona notizia è che il nostro algoritmo di riconoscimento di “8” funziona bene solo con immagini semplici in cui la lettera si trova perfettamente nel centro dell’immagine:

Reti Neurali

Ma ora la vera brutta notizia:

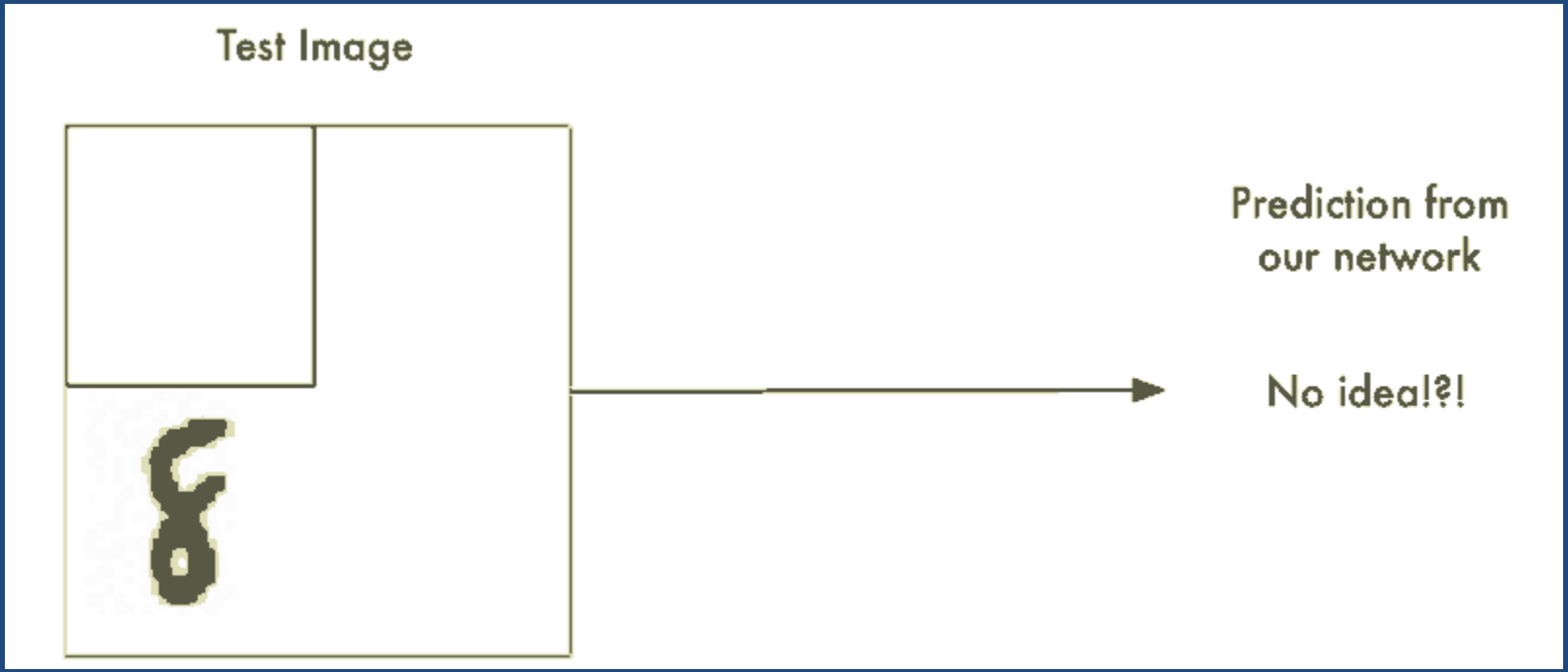
Il nostro algoritmo di riconoscimento di "8" *smette completamente di funzionare* quando la lettera non è più al centro dell'immagine. Basta il minimo cambiamento di posizione per rovinare tutto:

Reti Neurali

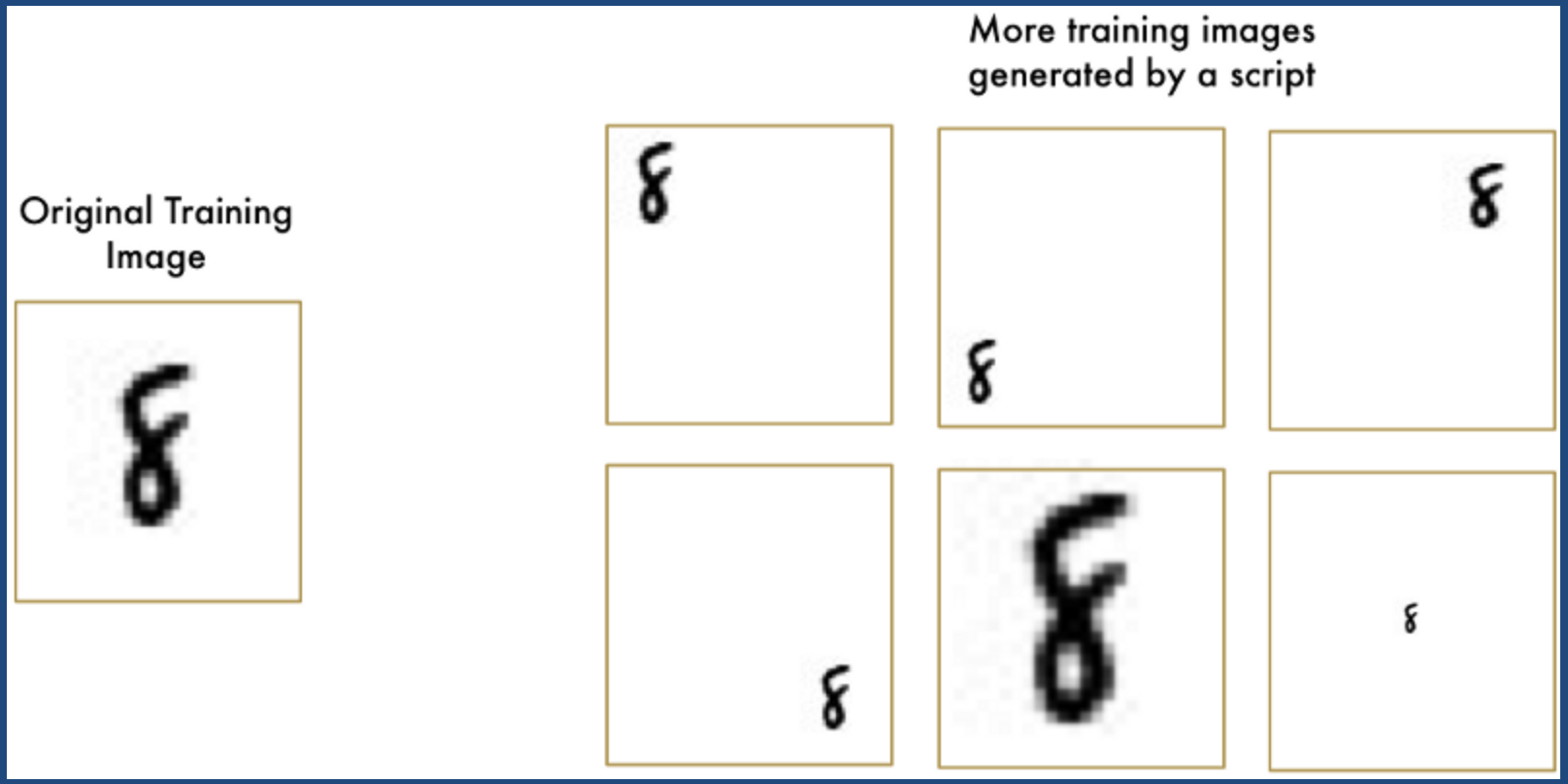
Questo avviene perché la Rete Neurale apprende solo la versione perfettamente centrata degli “8”, ma non ha assolutamente idea di cosa sia un “8” fuori centro. Il Neural Network conosce una sola versione e un solo pattern.

Questo non è molto utile. I problemi nel mondo reale non sono mai così semplici e puliti. Quindi abbiamo bisogno di trovare un modo per far funzionare il Neural Network nei casi in cui gli “8” non siano perfettamente centrati.

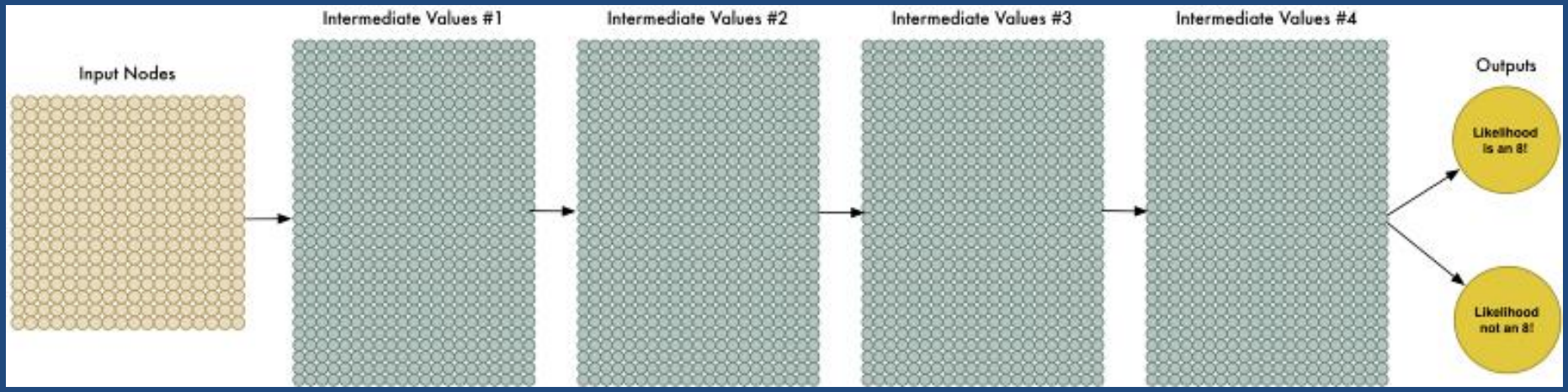
Reti Neurali



Reti Neurali



Deep Neural Network



Reti Neurali

- Non ha senso allenare un Neural Network a riconoscere in modo separato un “8” nella parte superiore di un immagine da un “8” sul fondo della stessa immagine.
- Ci vorrebbe un modo per rendere la rete neurale abbastanza intelligente da sapere che un “8” è la stessa cosa a prescindere dalla parte dell’immagine in cui si trova. Per fortuna … c’è!

Reti Neurali

- **La soluzione si chiama Convolutione**
- Per tutti gli esseri umani è intuitivo sapere che le immagini abbiano una *gerarchia* o *struttura concettuale*.

Reti Neurali

Reti Neurali

Un essere umano è in grado di riconosce immediatamente la gerarchia in questa immagine:

- Il terreno è coperto da erba e cemento
- C'è un bambino
- Il bambino è seduto su un cavallo gonfiabile
- Il cavallo gonfiabile è in cima al prato

Ancora più importante, noi riconosciamo l'idea di un *bambino* a prescindere da quale sia la superficie che il bambino sta calpestando. Non abbiamo imparare nuovamente l'idea di *bambino* per ogni possibile superficie sulla quale il bambino potrebbe apparire.

Reti Neurali

Ma in questo momento, il nostro Neural Network non è in grado di farlo in quanto considera che un “8” in una parte diversa dell’immagine sia una cosa completamente diversa. Non capisce che lo spostamento di un oggetto all’interno della foto non lo rende qualcosa di diverso. Questo significa che deve re-imparare l’identità di ogni oggetto in tutte le posizioni. Un bel problema!

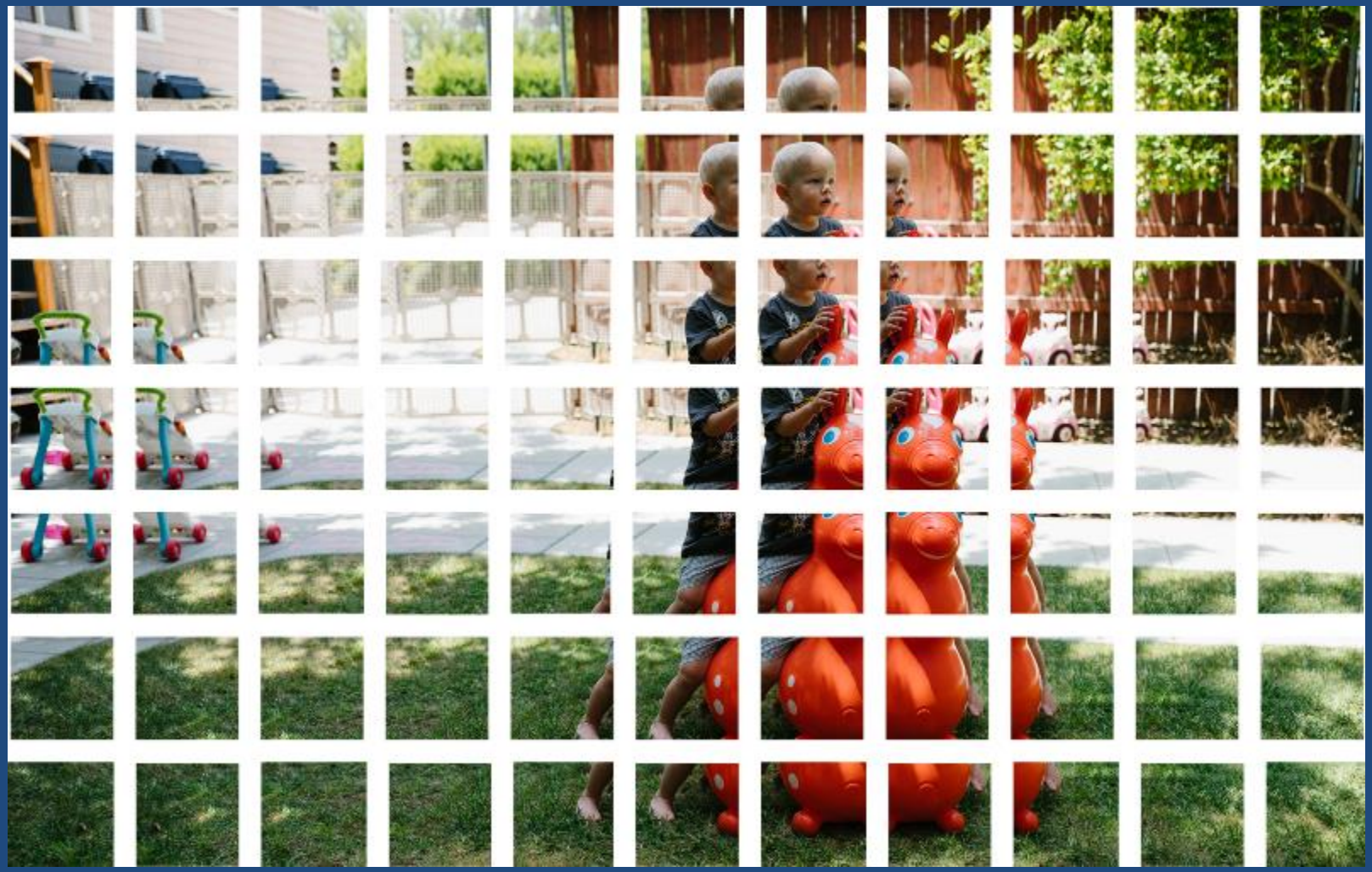
Reti Neurali

Dobbiamo fare in modo che il nostro Neural Network sia in grado di interpretare la **Translation Invariance** — Invarianza di Traslazione — ovvero che un “8” rimane un “8” a prescindere dalla parte della foto in cui si trovi.

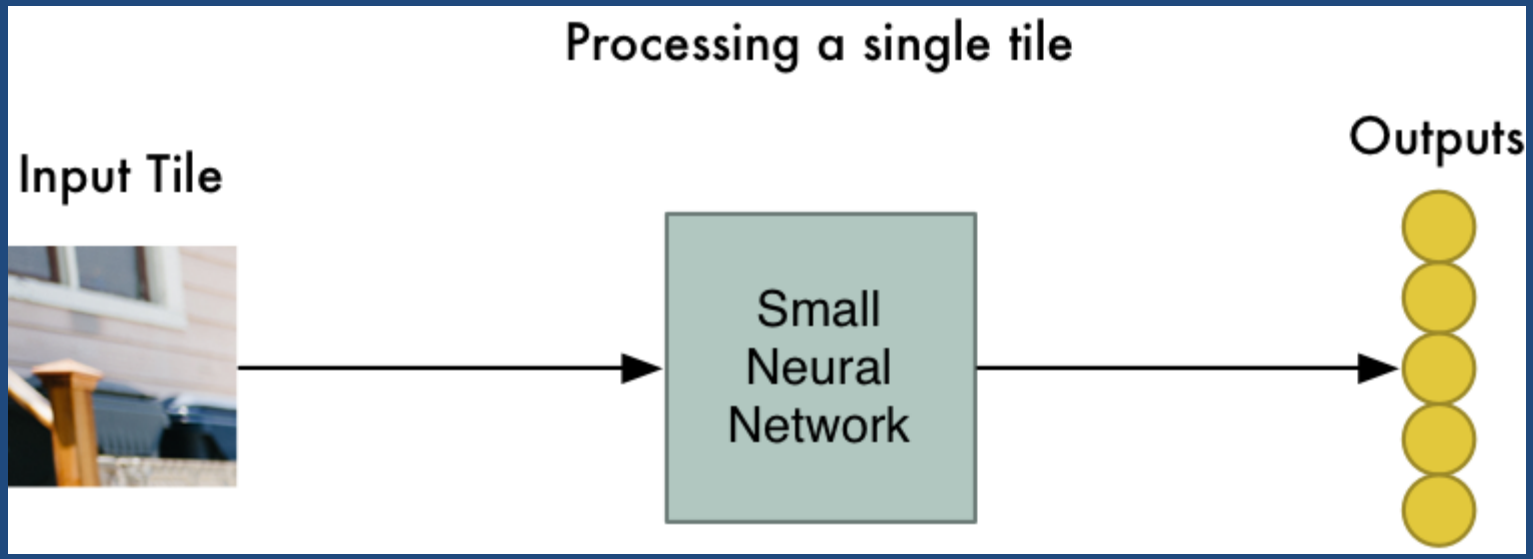
Lo faremo utilizzando un processo chiamato **Convolution** — Convoluzione.

L’idea di Convolution si ispira in parte alla scienza informatica e in parte alla biologia.

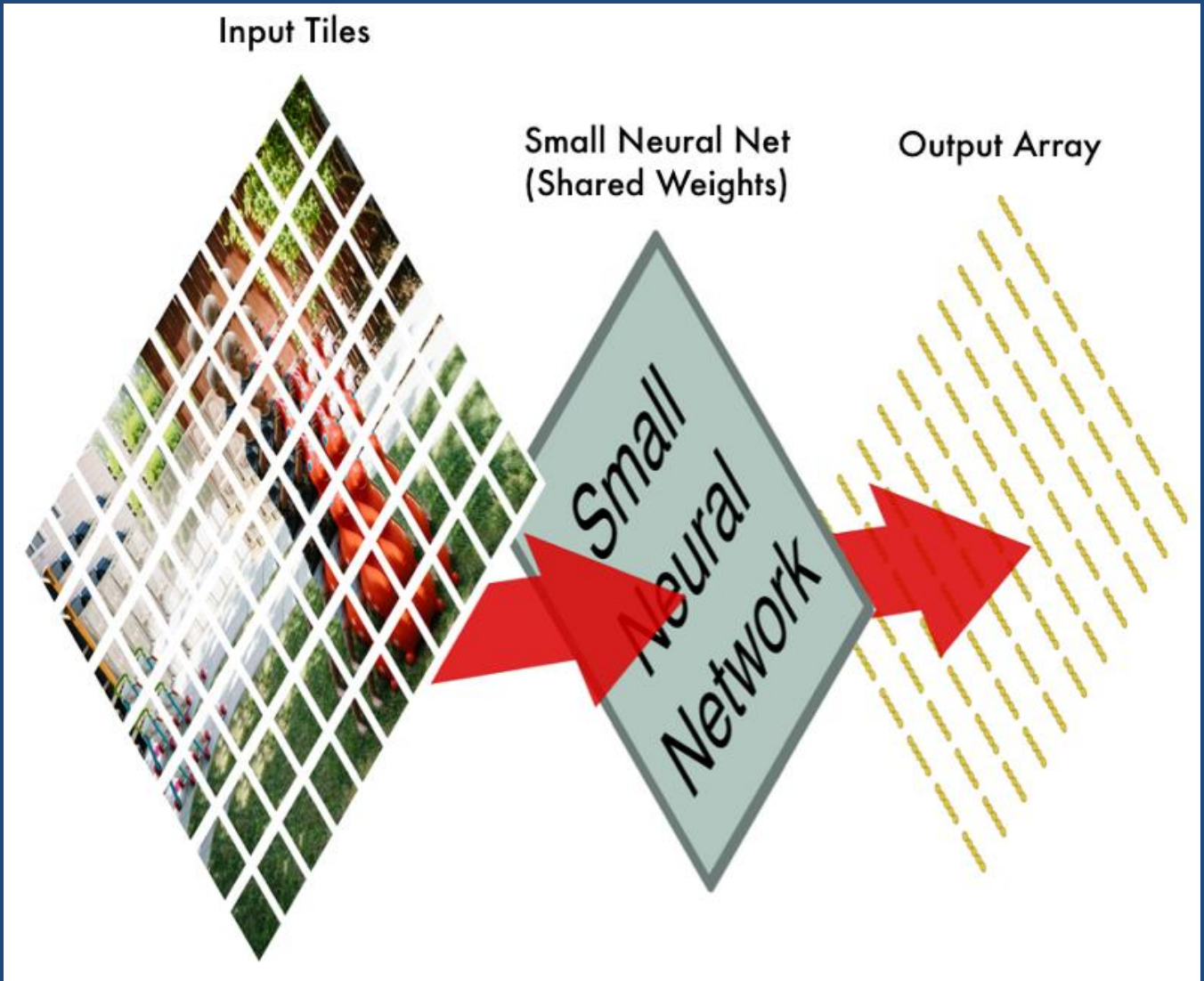
Convoluzione



Convoluzione



Convoluzione

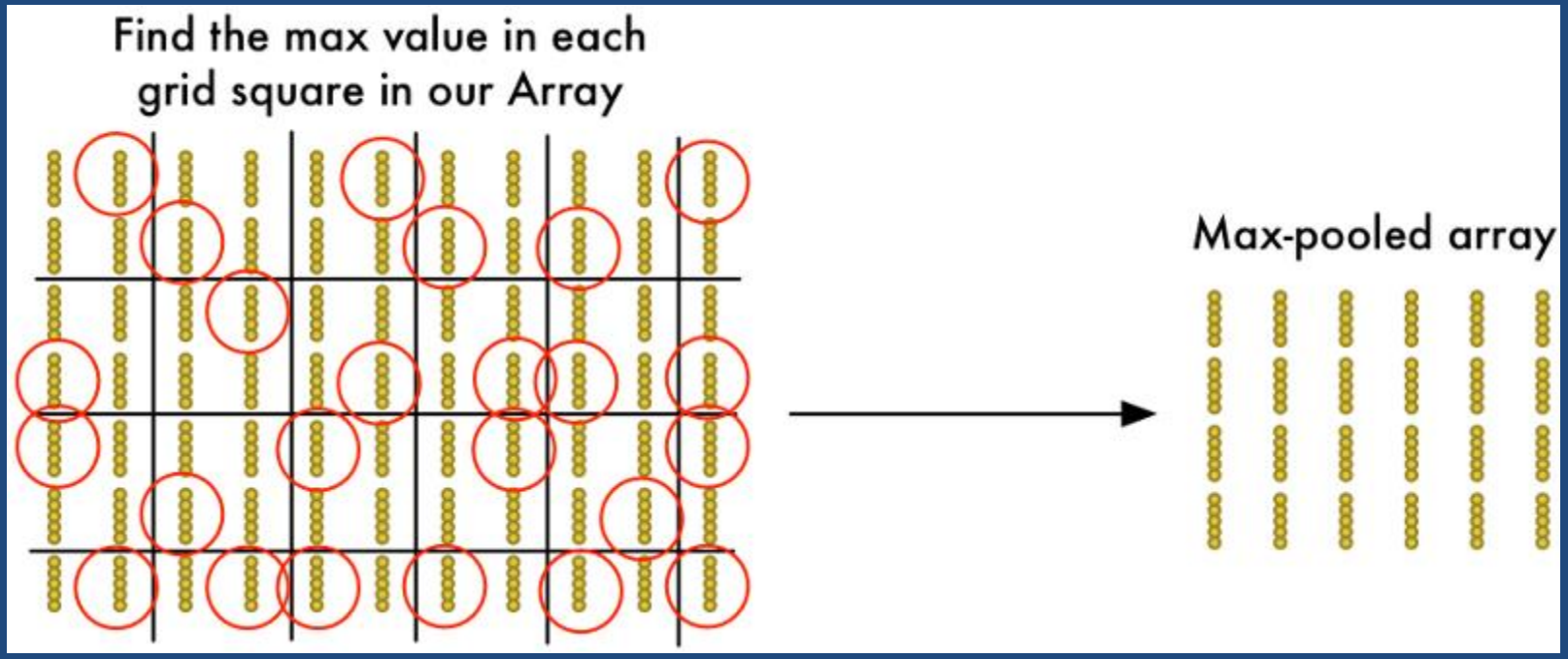


Convoluzione

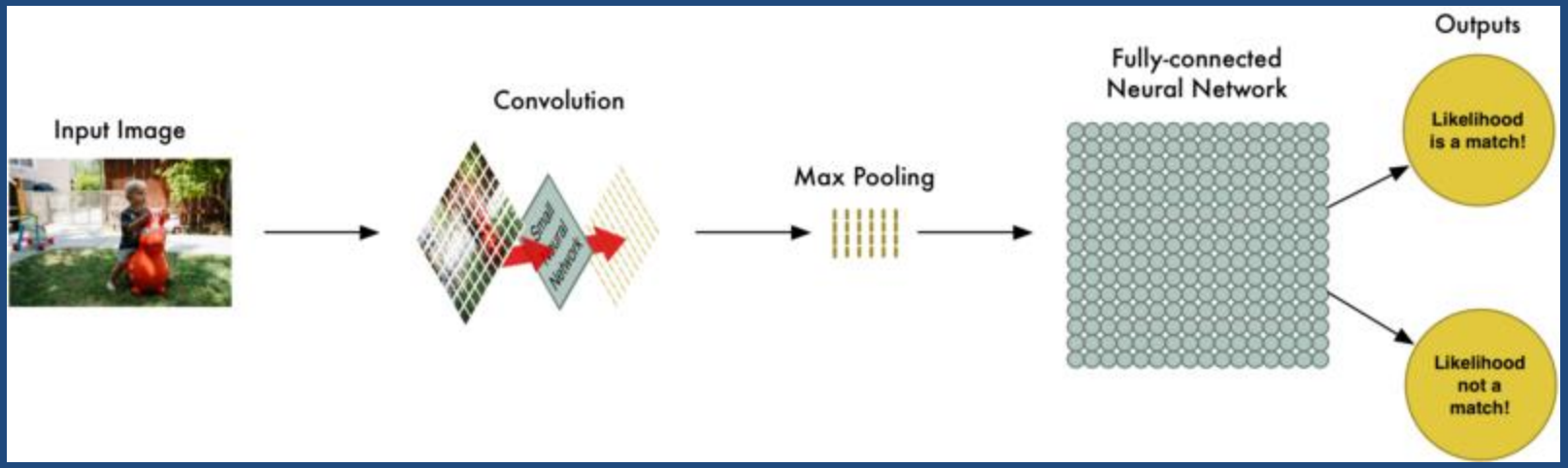
Original Input Image

Array resulting from convolution in Step 3

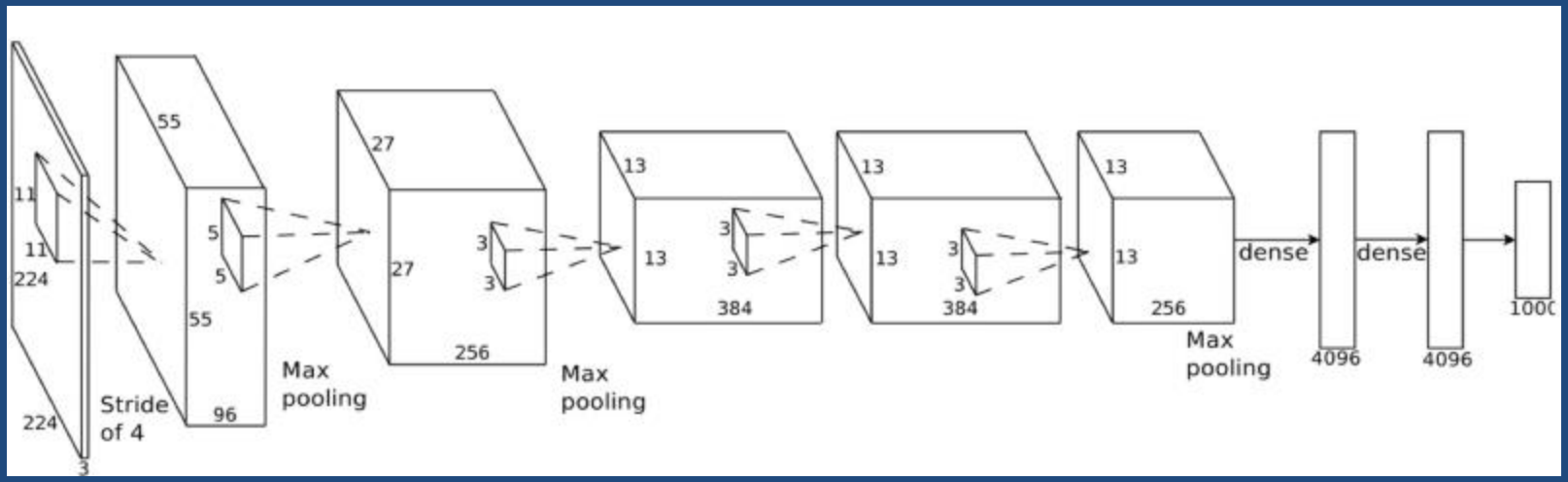
Convoluzione



Convoluzione



Convoluzione



Google dipinge come VanGogh

Google dipinge come VanGogh

Google dipinge come VanGogh

Data Science Words

Unsupervised Learning - Clustering

- Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns.
- By contrast with SUPERVISED LEARNING or REINFORCEMENT LEARNING, there are no explicit target outputs or environmental evaluations associated with each input; rather the unsupervised learner brings to bear prior biases as to what aspects of the structure of the input should be captured in the output.

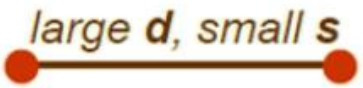
Unsupervised Learning - Clustering

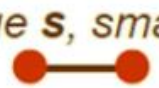
- The organization of unlabeled data into similarity groups called clusters.
- A cluster is a collection of data items which are “similar” between them, and “dissimilar” to data items in other clusters

Unsupervised Learning - Clustering

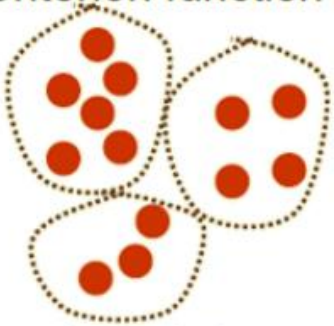
- What do we need for clustering?

1. Proximity measure, *either*
   - similarity measure $s(x_i,x_k)$: large if x_i,x_k are similar
   - dissimilarity(or distance) measure $d(x_i,x_k)$: small if x_i,x_k are similar





2. Criterion function to evaluate a clustering



good clustering

bad clustering

3. Algorithm to compute clustering
   - For example, by optimizing the criterion function

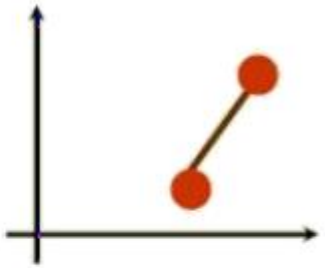
Unsupervised Learning - Clustering

- Distance (dissimilarity) measures

- Euclidean distance

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} \left(x_i^{(k)} - x_j^{(k)}\right)^2}$$

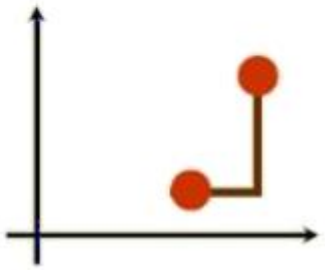
  - translation invariant



- Manhattan (city block) distance

$$d(x_i, x_j) = \sum_{k=1}^{d} \left|x_i^{(k)} - x_j^{(k)}\right|$$

  - approximation to Euclidean distance, cheaper to compute

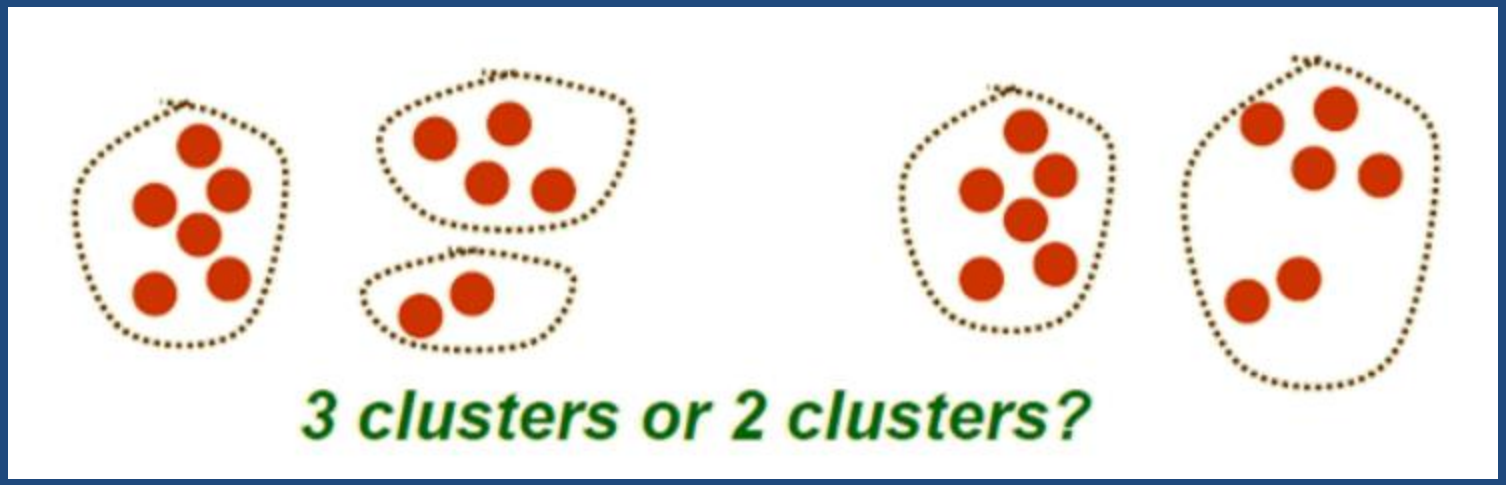


Unsupervised Learning - Clustering

- Cluster evaluation (a hard problem)
  - **Intra-cluster cohesion** (compactness):
    - Cohesion measures how near the data points in a cluster are to the cluster centroid.
    - Sum of squared error (SSE) is a commonly used measure.
  - **Inter-cluster separation** (isolation):
    - Separation means that different cluster centroids should be far away from one another.
- In most applications, expert judgments are still the key

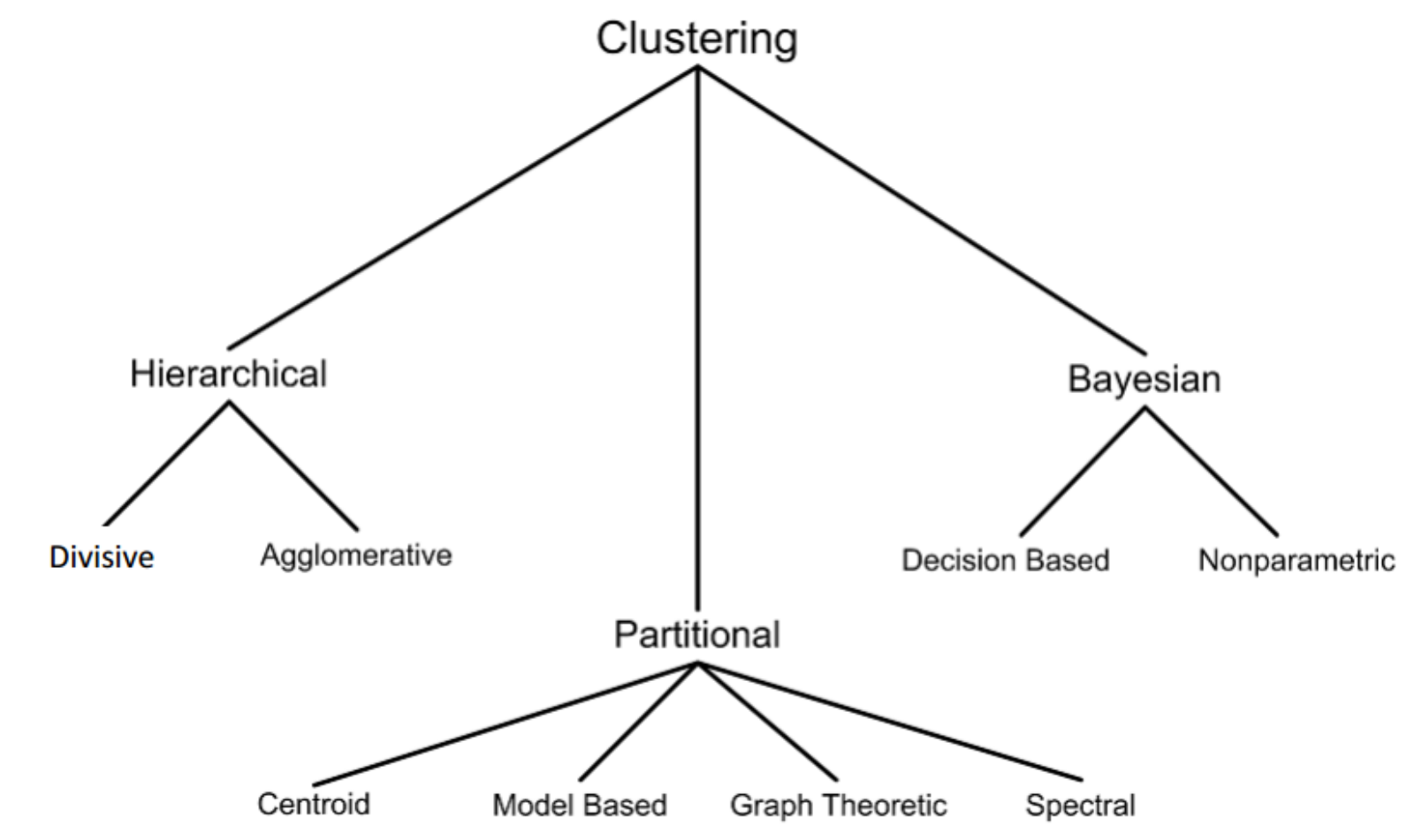
Unsupervised Learning - Clustering

- How many clusters?



- Possible approaches:
  - Fix the number of clusters to K
  - Find the best clustering according to the criterion function

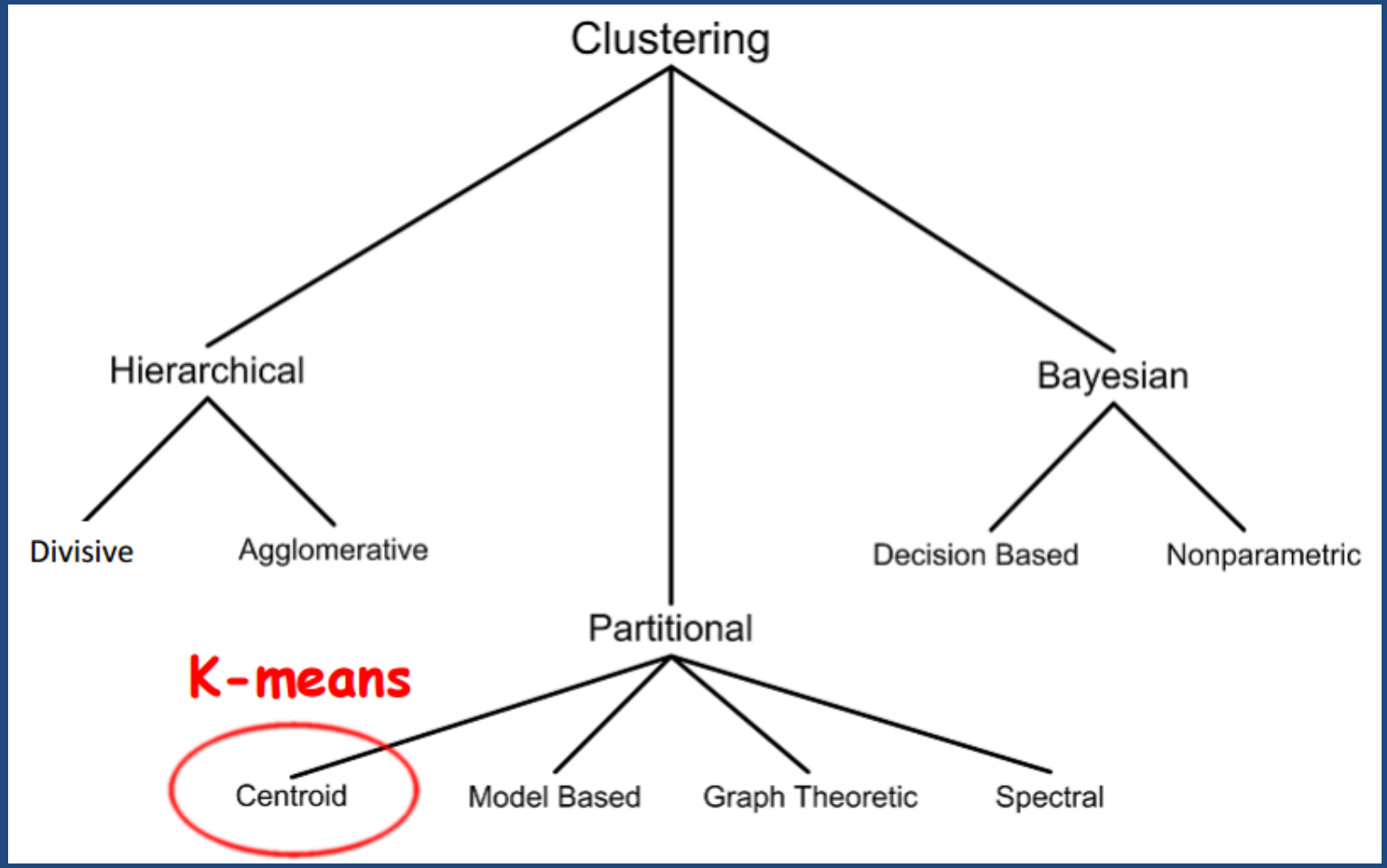
Clustering Techiques



Clustering Techiques

- Hierarchical algorithms find successive clusters using previously established clusters . These algorithms can be either agglomerative ("botton-up") or divisive ("top-down"):
  - Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters;
  - Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.
- Partitional algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.
- Bayesian algorithms try to generate a posteriori distribution over the collection of all partitions of the data.

Clustering Techiques



k-means

- The k-means algorithm is a simple iterative method to partition a given dataset into a user specified number of clusters, *k.*
- Let the set of data points D be $\{x_1 , x_2 , \ldots, x_n\}$, where $x_i = (x_{i1} , x_{i2} , \ldots, x_{ir})$ is a vector in a space of **r** dimensions.
- The k-means algorithm partitions the given data into k clusters:
  - Each cluster has a cluster center, called centroid.
  - k is specified by the user

k-means Algorithm

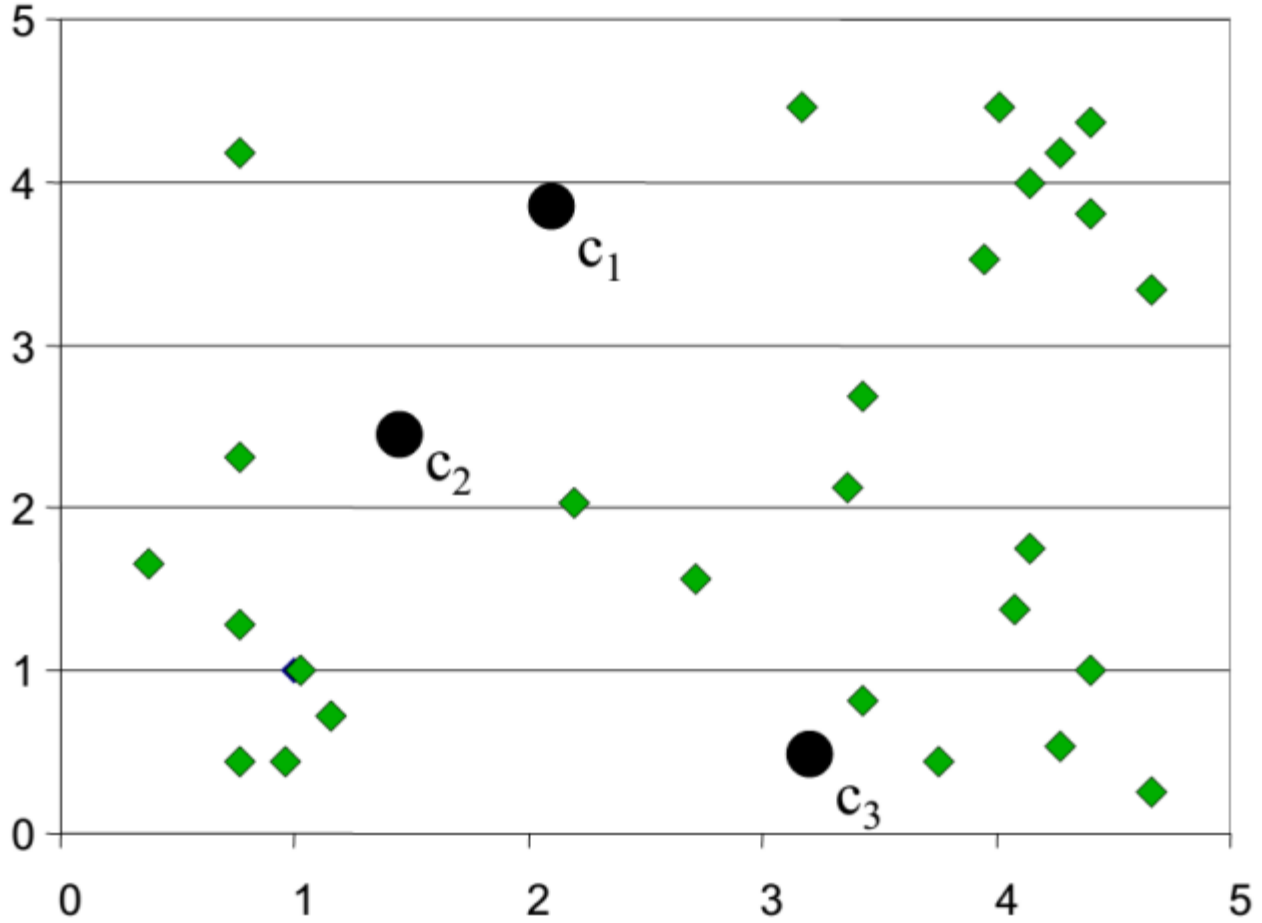
- Given k, the k-means algorithm works as follows:
  1. Choose k (random) data points (seeds) to be the initial centroids, cluster centers
  2. Assign each data point to the closest centroid
  3. Re-compute the centroids using the current cluster memberships
  4. If a convergence criterion is not met, repeat steps 2 and 3

k-means Algorithm

- K-means convergence (stopping) criterion:
  - no (or minimum) re-assignments of data points to different clusters,
  - no (or minimum) change of centroids

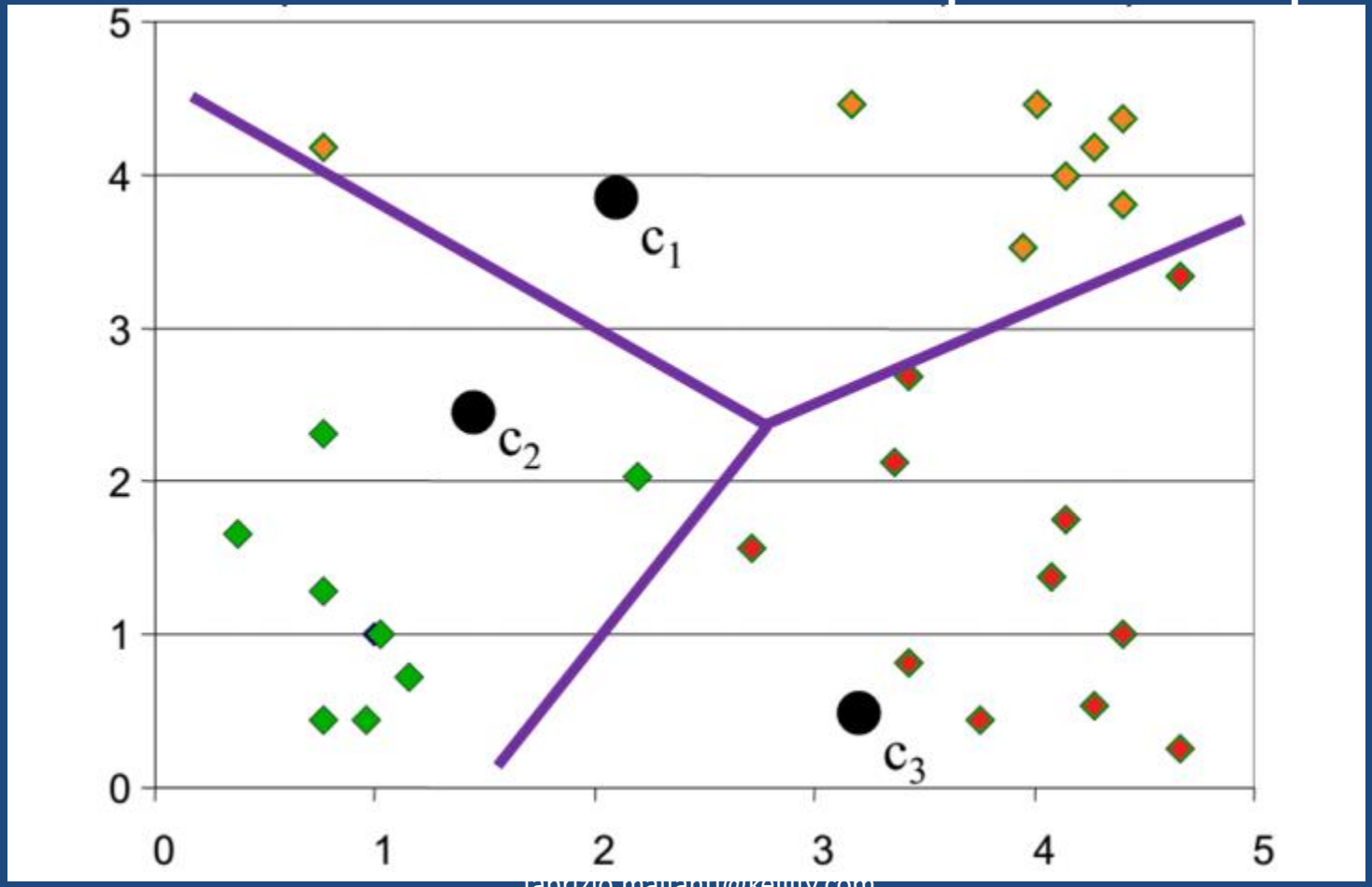
K-means example

- Randomly initialize the cluster centers (k=3)



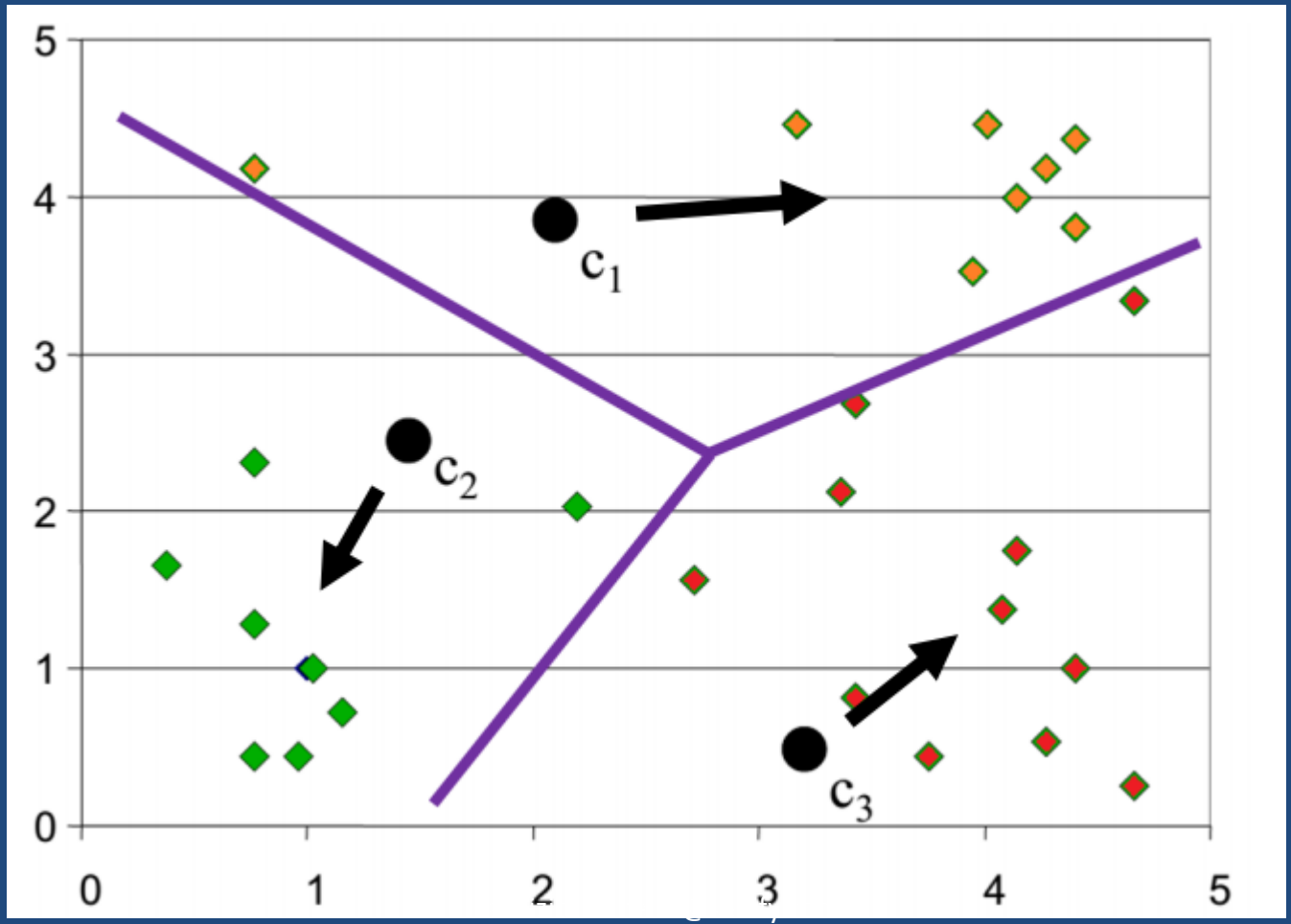
K-means example

- Determine cluster membership for each point



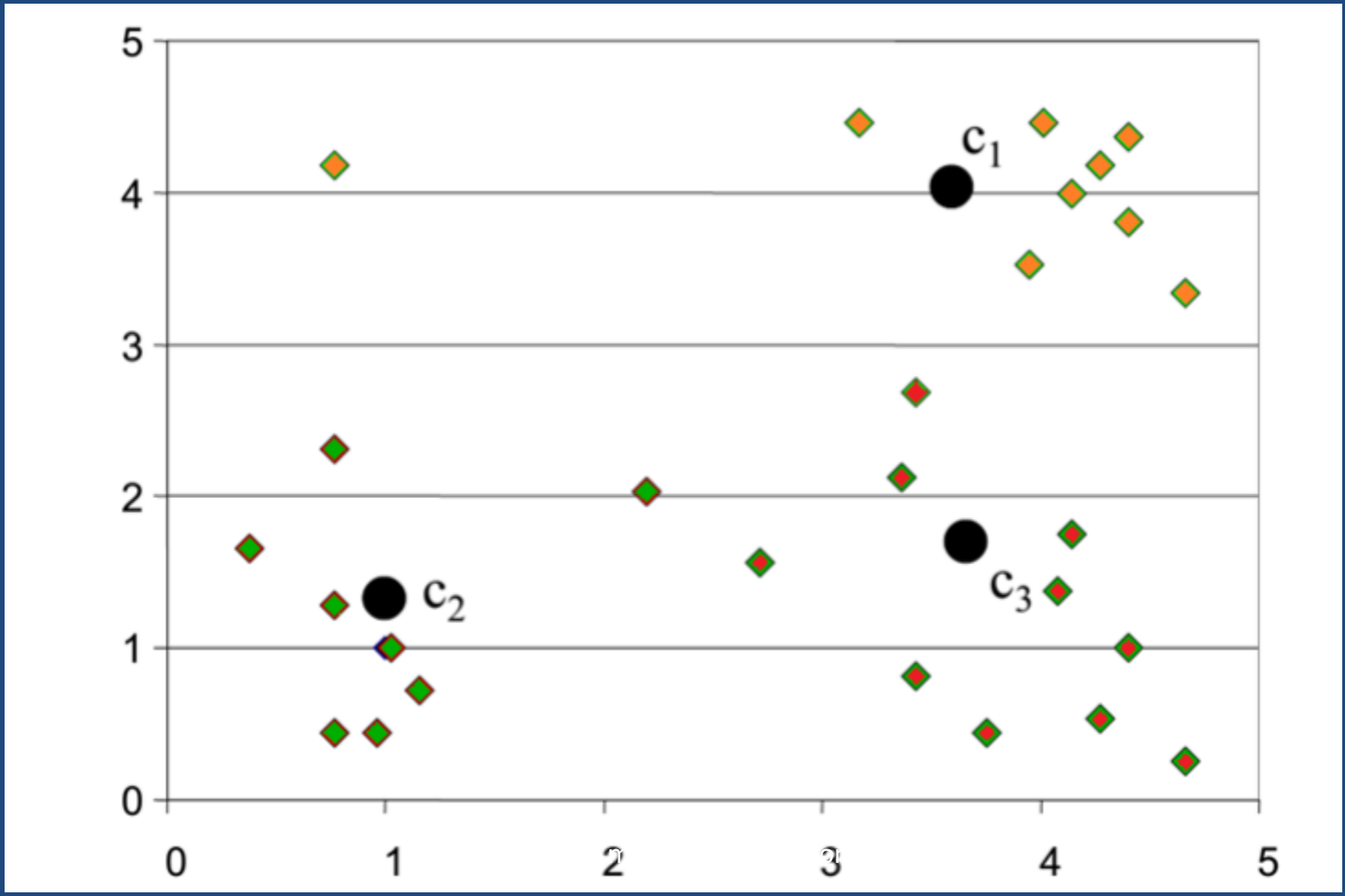
K-means example

- Re-estimates cluster centers



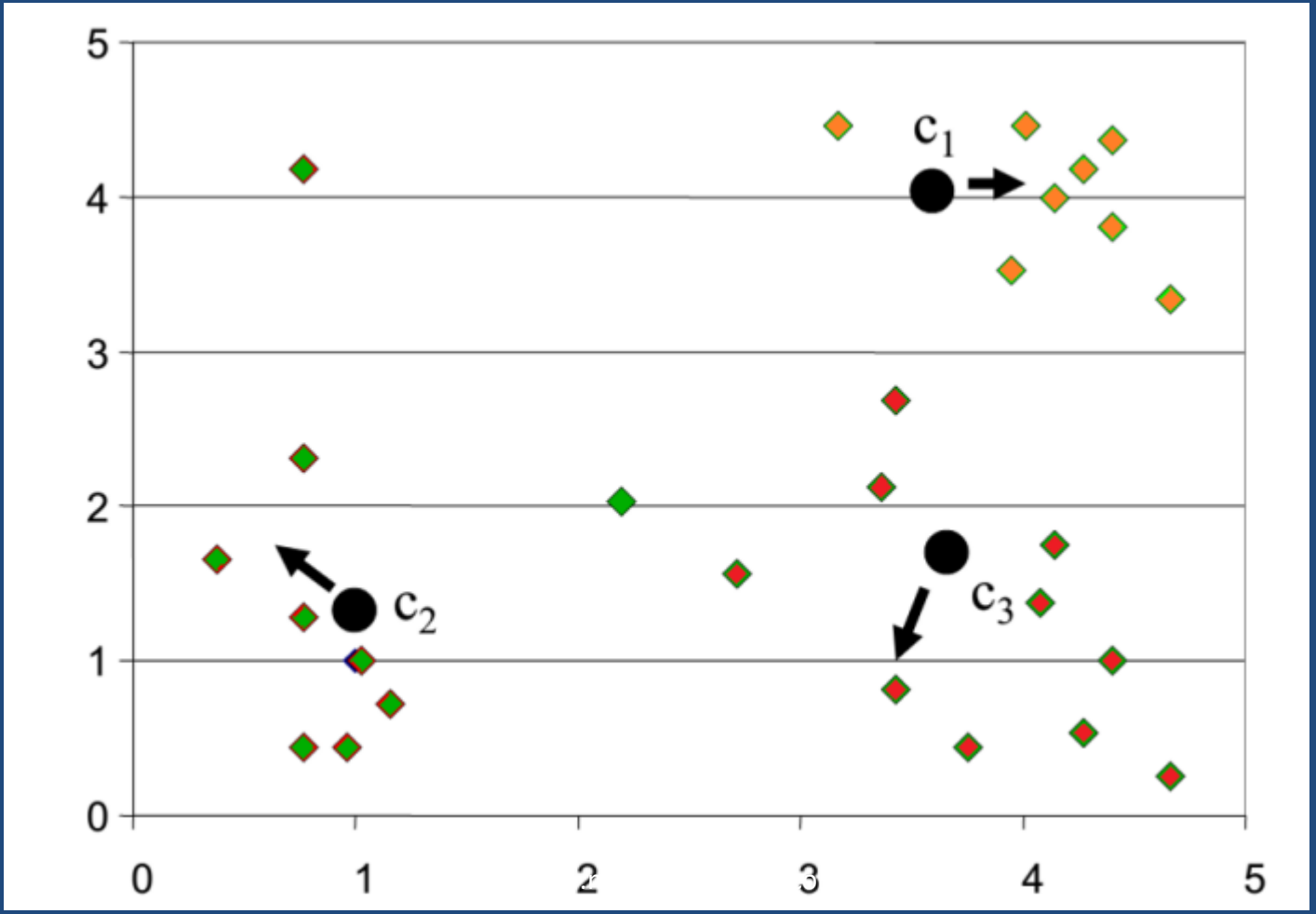
K-means example

- Result of first iteration



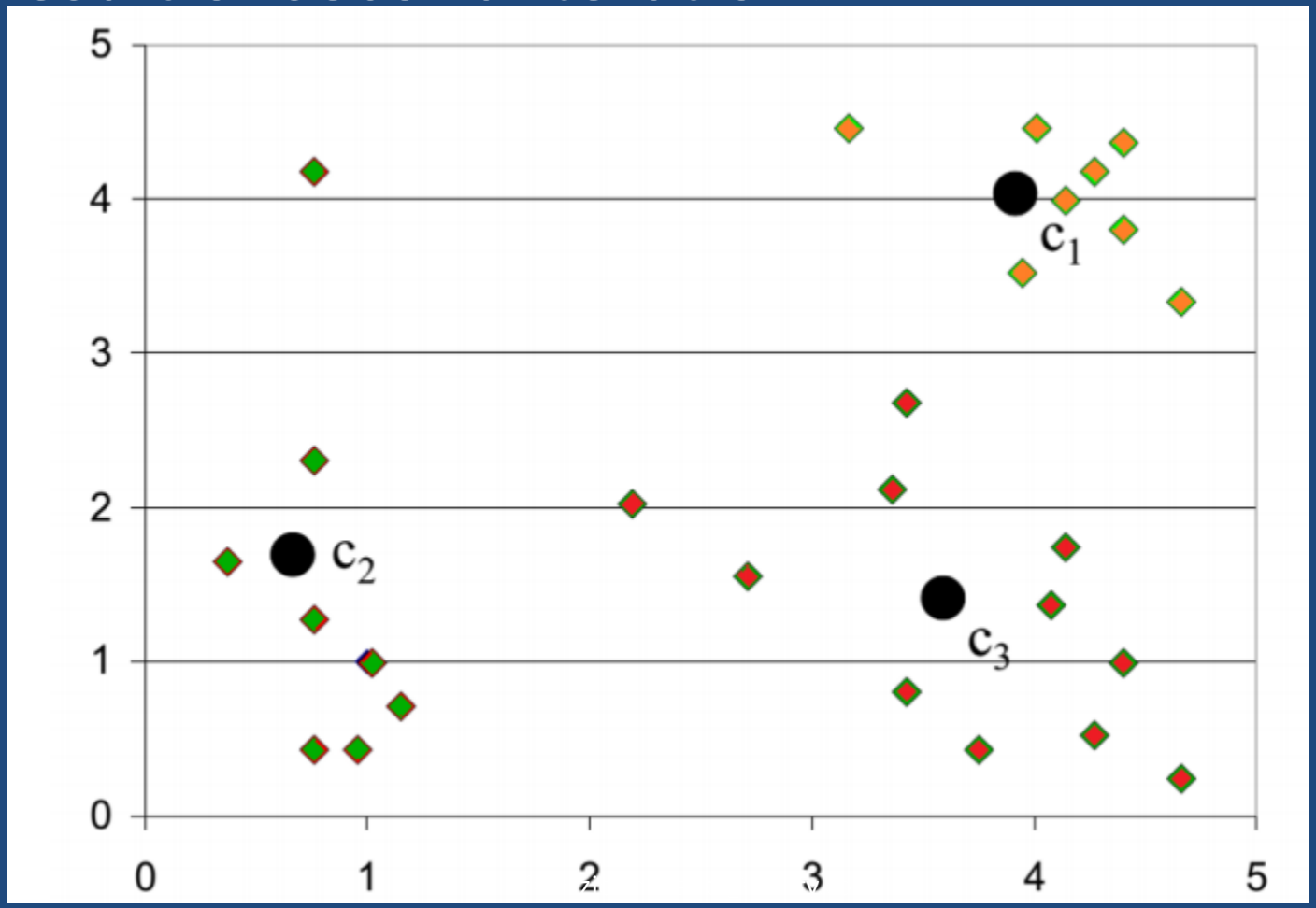
K-means example

- Second iteration



K-means example

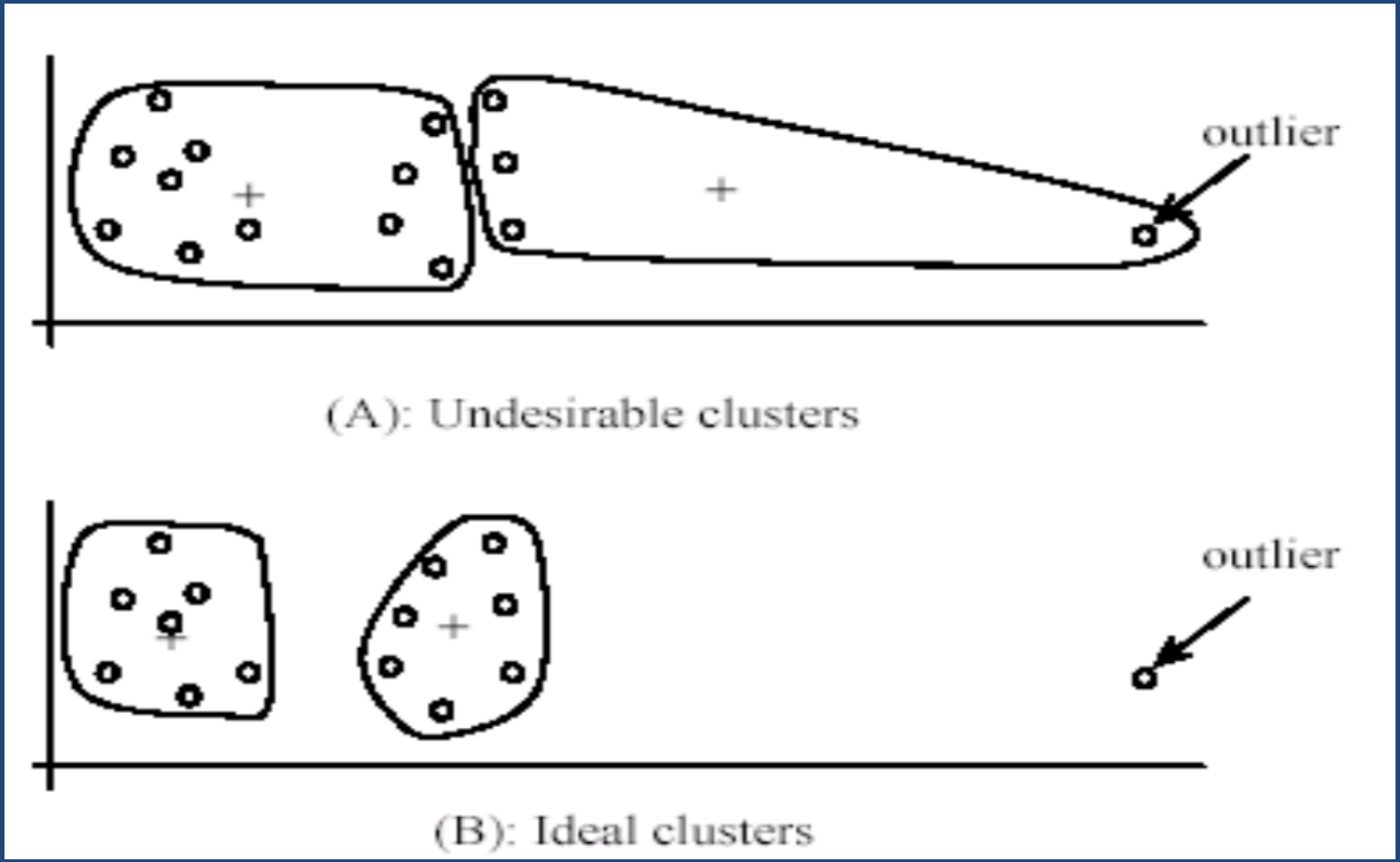
- Result of second iteration:



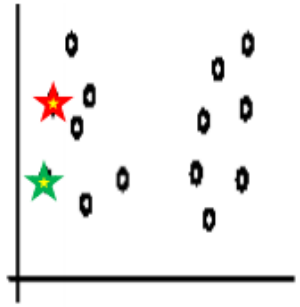
Weaknesses of K-means

- The algorithm is only applicable if the mean is defined. (For categorical data, k-mode - the centroid is represented by most frequent values.)
- The user needs to specify k.
- The algorithm is sensitive to outliers and to the randomly generated initial centroids
- The k-means algorithm is not suitable for discovering clusters that are not convex

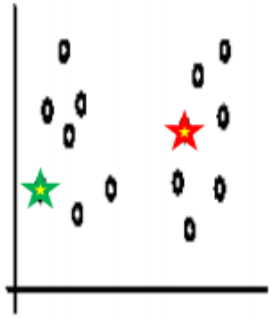
Outliers



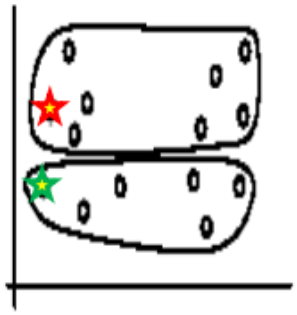
Sensitivity to initial seeds



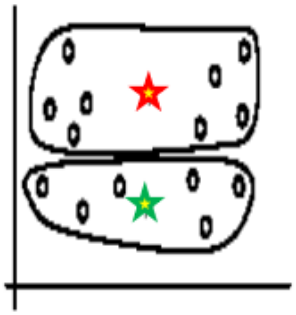
Random selection of seeds (centroids)



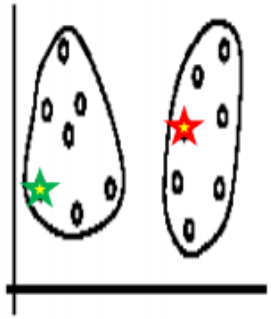
Random selection of seeds (centroids)



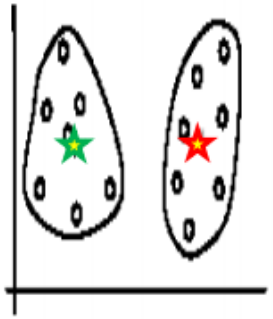
Iteration 1



Iteration 2

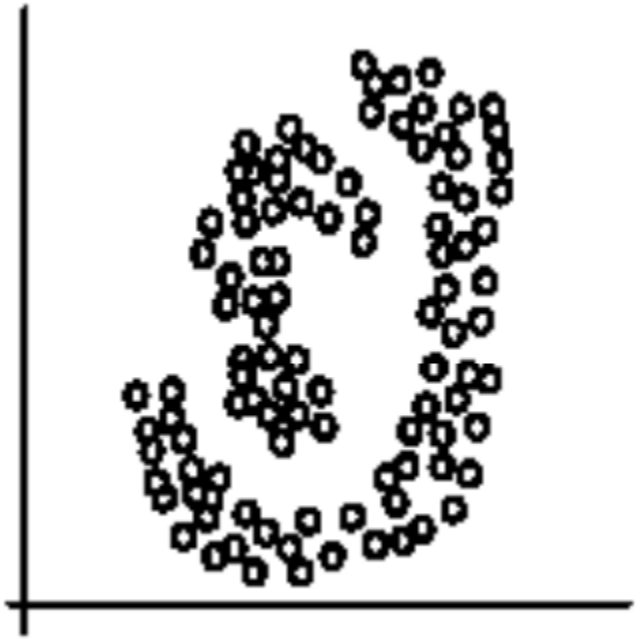


Iteration 1

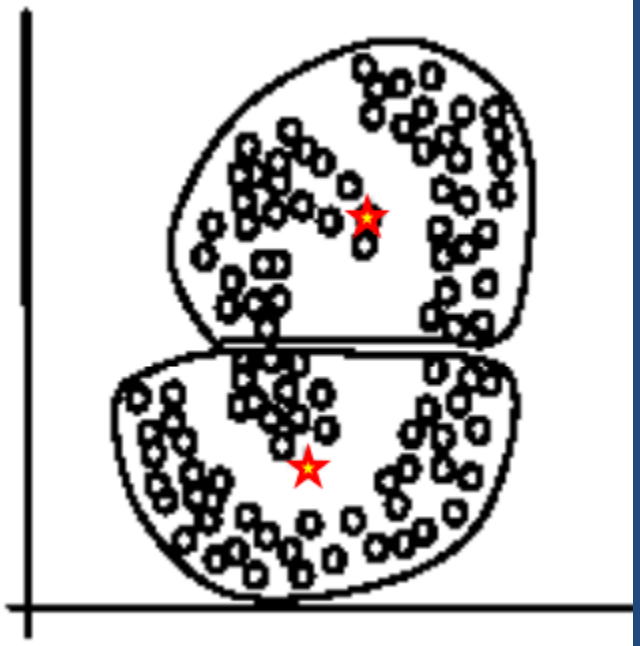


Iteration 2

Special data structures



(A): Two natural clusters

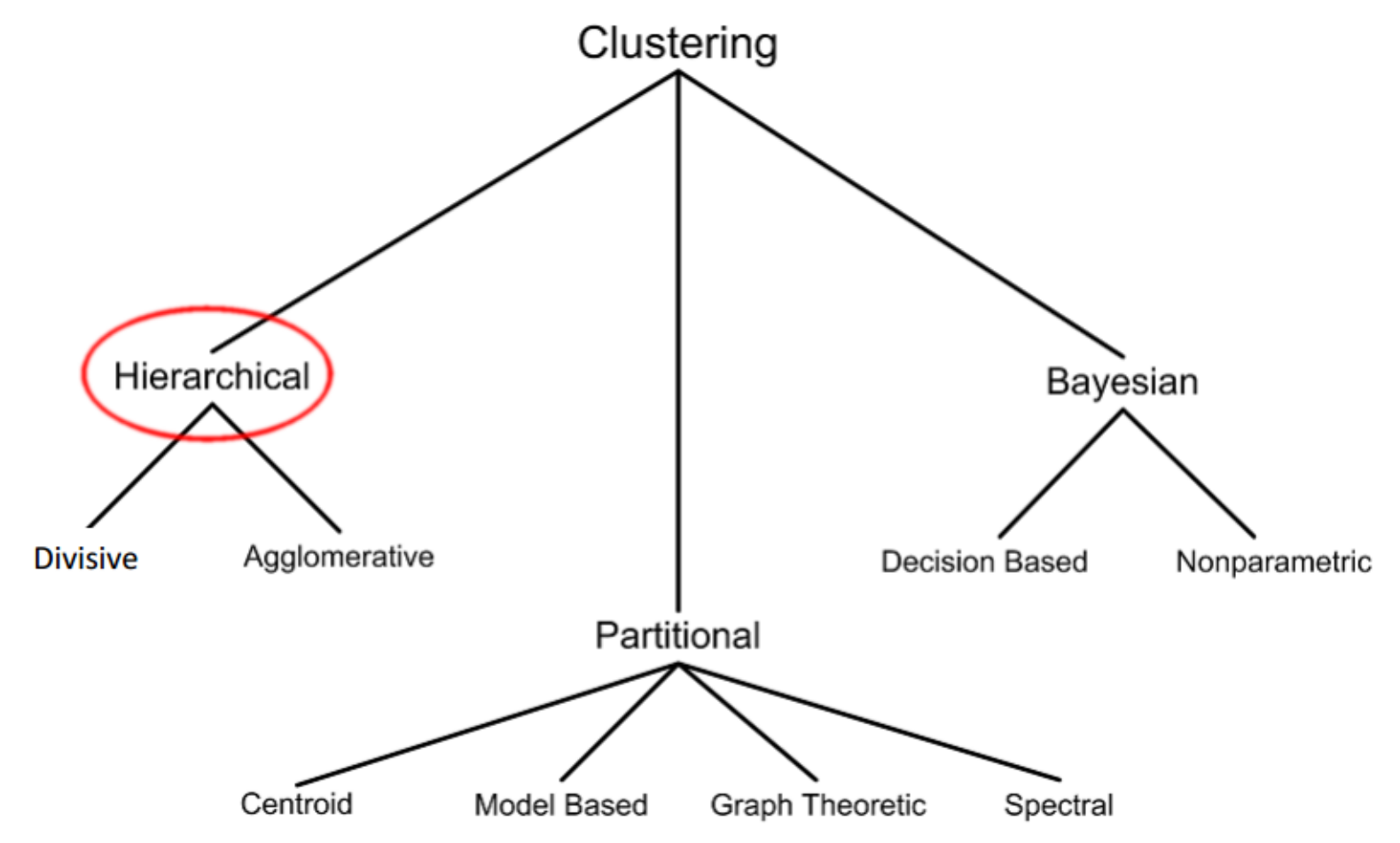


(B): *k*-means clusters

K-means summary

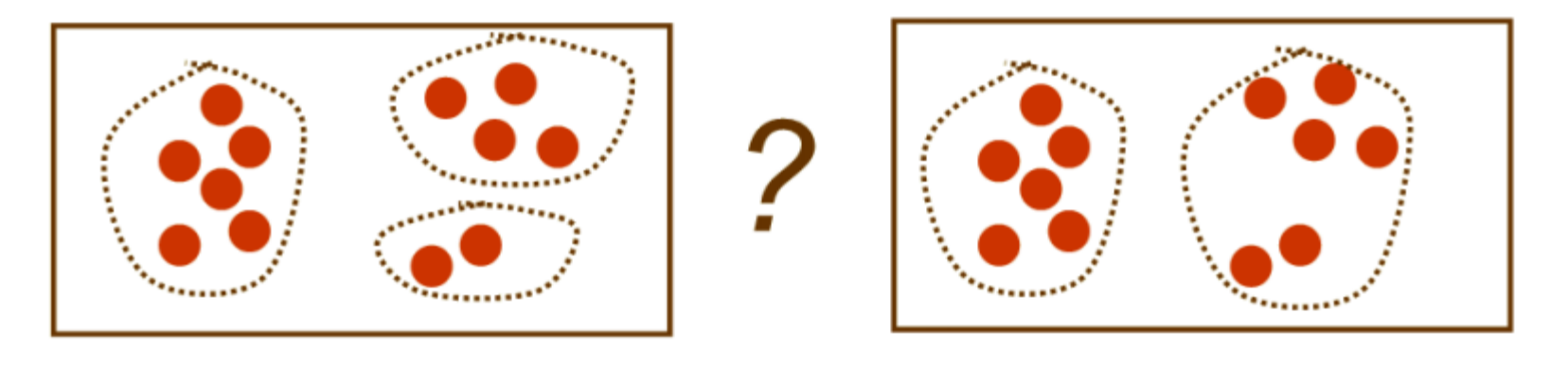
- Despite weaknesses, k-means is still the most popular algorithm due to its simplicity and efficiency
- No clear evidence that any other clustering algorithm performs better in general
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

Clustering techniques

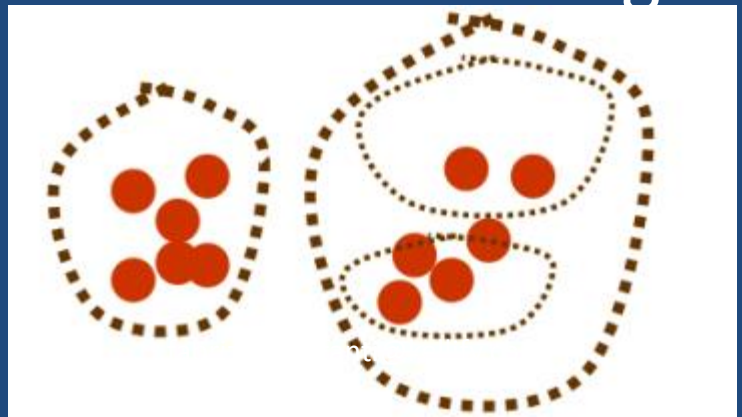


Hierarchical clustering

- Up to now we considered only flat clustering

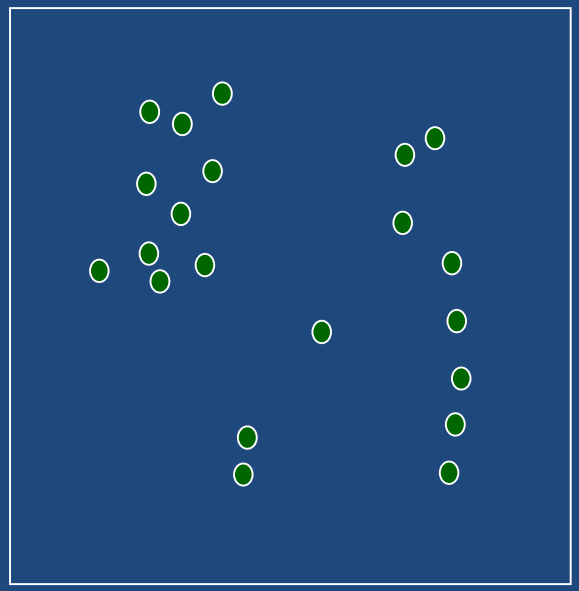


- For some data Hierarchical clustering is more appropriate than flat clustering



Hierarchical Agglomerative Clustering

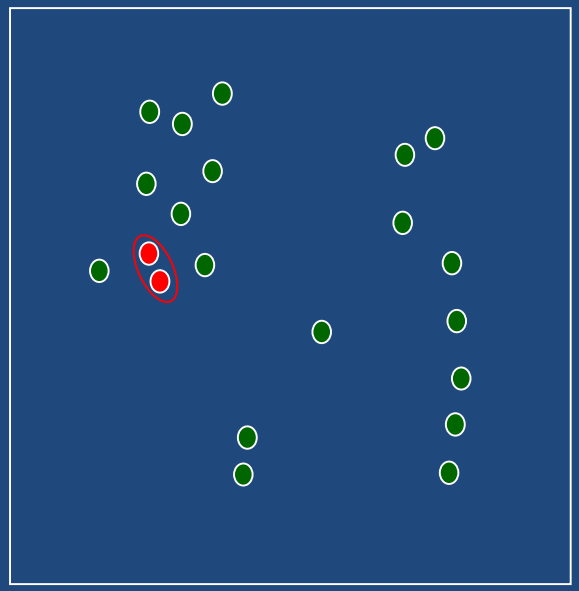
Initially, every datum is a cluster



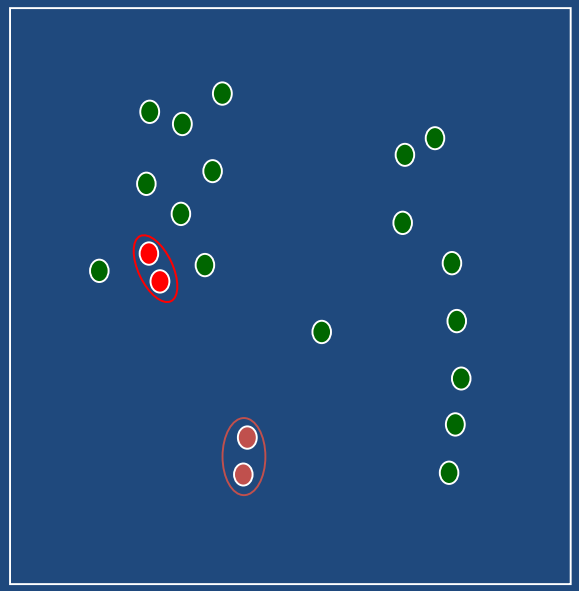
- Another simple clustering algorithm

- Define a distance between clusters
- Initialize: every example is a cluster
- Iterate:
  - Compute distances between all clusters (store for efficiency)
  - Merge two closest clusters
- Save both clustering and *sequence* of cluster operations
- "Dendrogram"

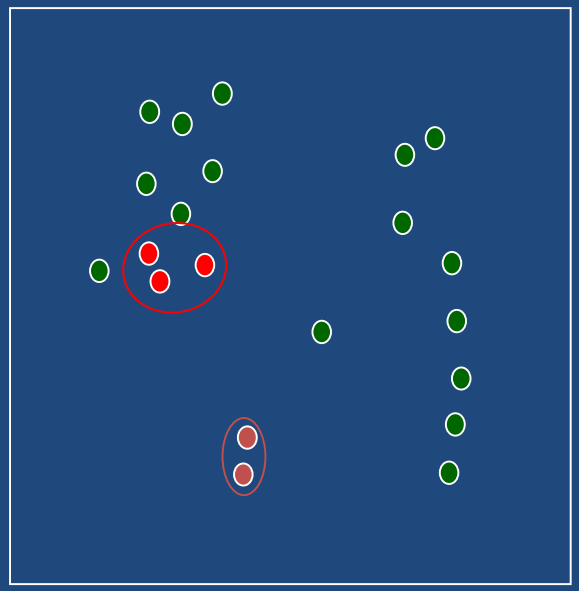
Iteration 1

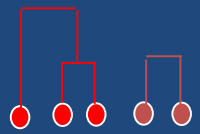


Iteration 2



Iteration 3

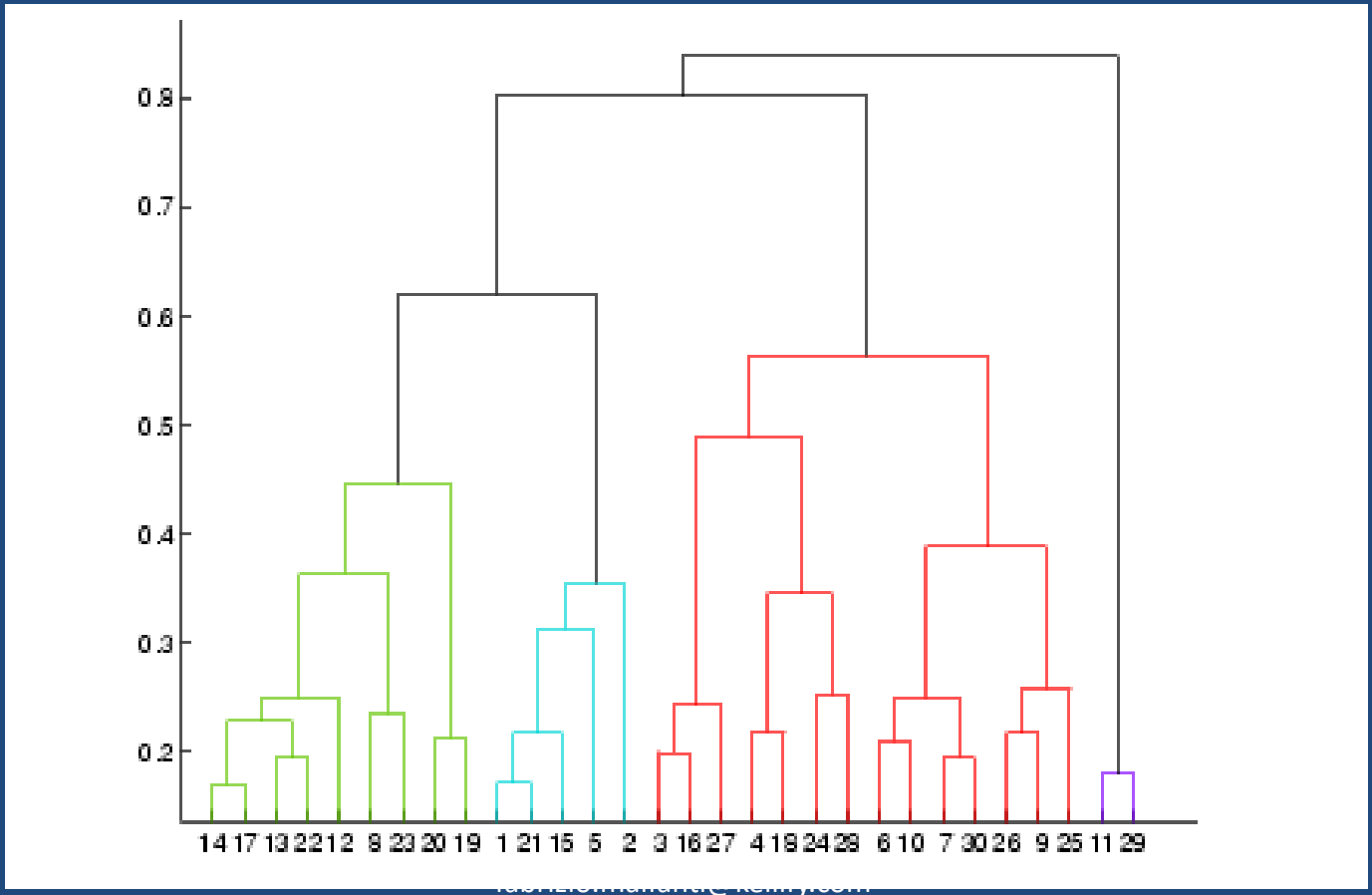




- Builds up a sequence of clusters (“hierarchical”)

A Dendrogram

- preferred way to represent Hierarchical Clusters

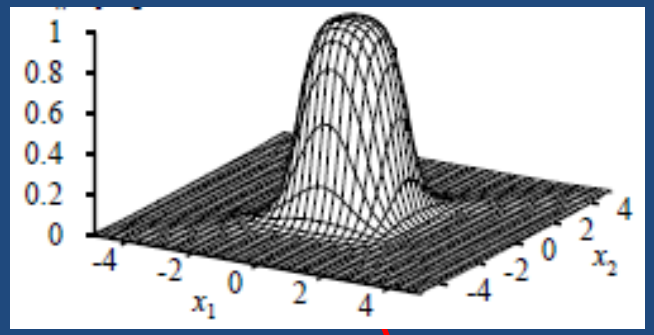


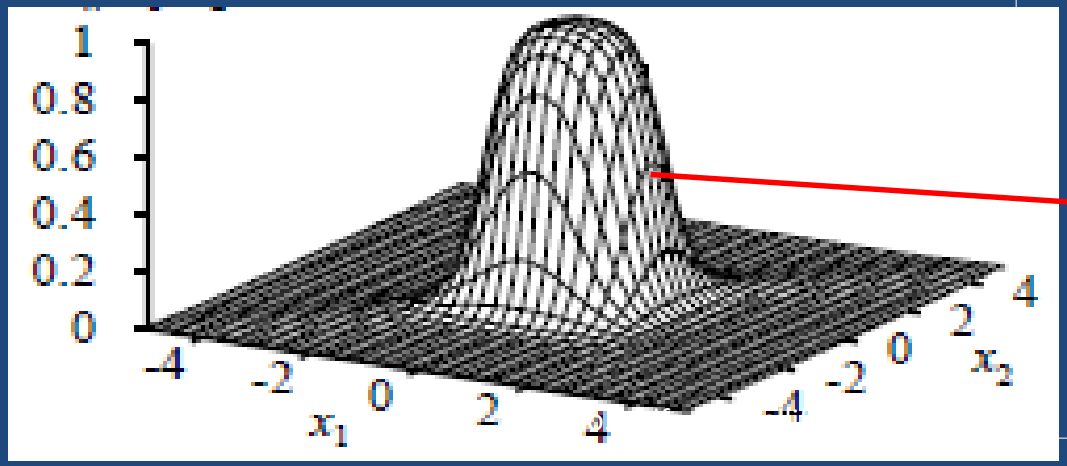
Mixtures of Gaussians and EM

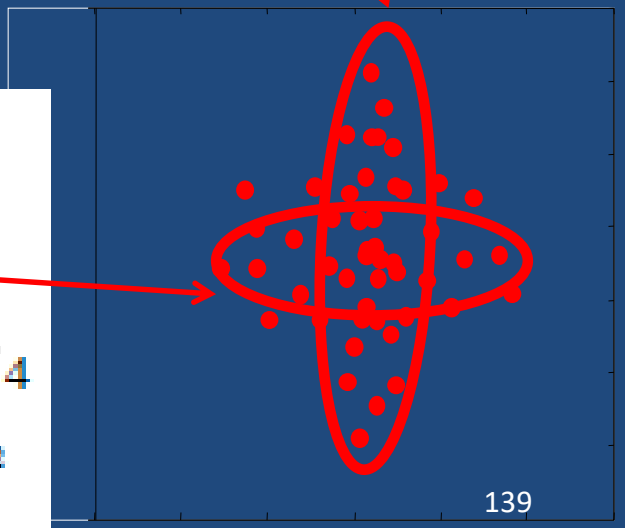
- Finite mixture distributions provide a flexible and mathematical-based approach to the modeling and clustering of data observed on random phenomena.
- We focus here on the use of normal (Gaussian) mixture models, which can be used to cluster continuous data and to estimate the underlying density function.
- These mixture models can be fitted by maximum likelihood via the EM (Expectation–Maximization) algorithm.

Mixtures of Gaussians and EM

Clusters modeled as multivariate Gaussians
EM algorithm: assign data to
cluster with some *probability*







EM Algorithm: E-step

- Start with parameters describing each cluster
- Mean μ_c, Covariance Σ_c, "size" π_c

- E-step ("Expectation")
  - For each datum (example) x_i,
  - Compute "r_{ic}", the probability that it belongs to cluster c
    - Compute its probability under model c
    - Normalize to sum to one (over clusters c)

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

  - If x_i is very likely under the c^{th} Gaussian, it gets high weight
  - Denominator just makes r's sum to one

EM Algorithm: M-step

- Start with assignment probabilities r_{ic}
- Update parameters: mean μ_c, Covariance Σ_c, "size" π_c
- M-step ("Maximization")
  - For each cluster (Gaussian) x_c,
  - Update its parameters using the (weighted) data points

$$N_c = \sum_i r_{ic}$$

Total responsibility allocated to cluster c

$$\pi_c = \frac{N_c}{N}$$

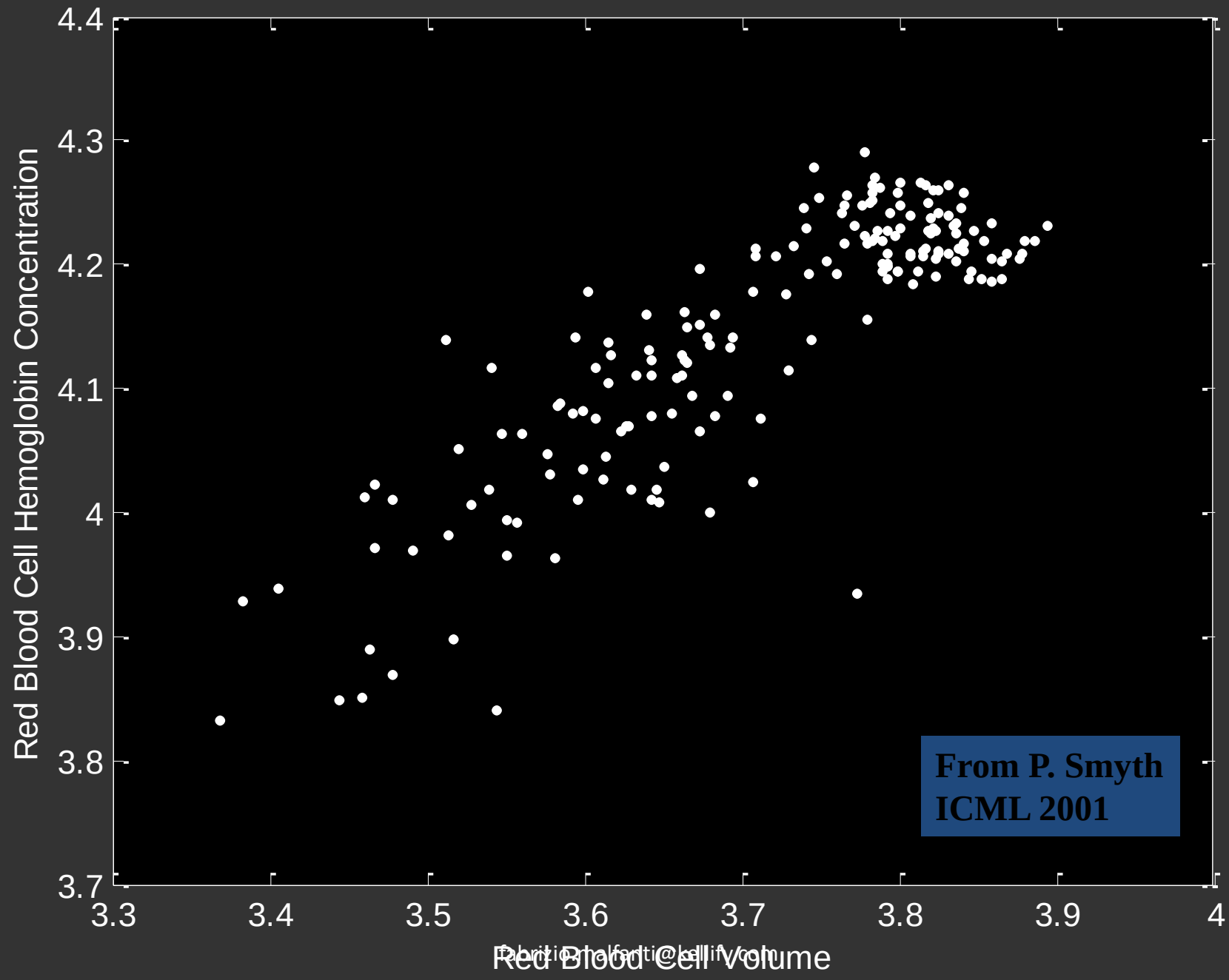
Fraction of total assigned to cluster c

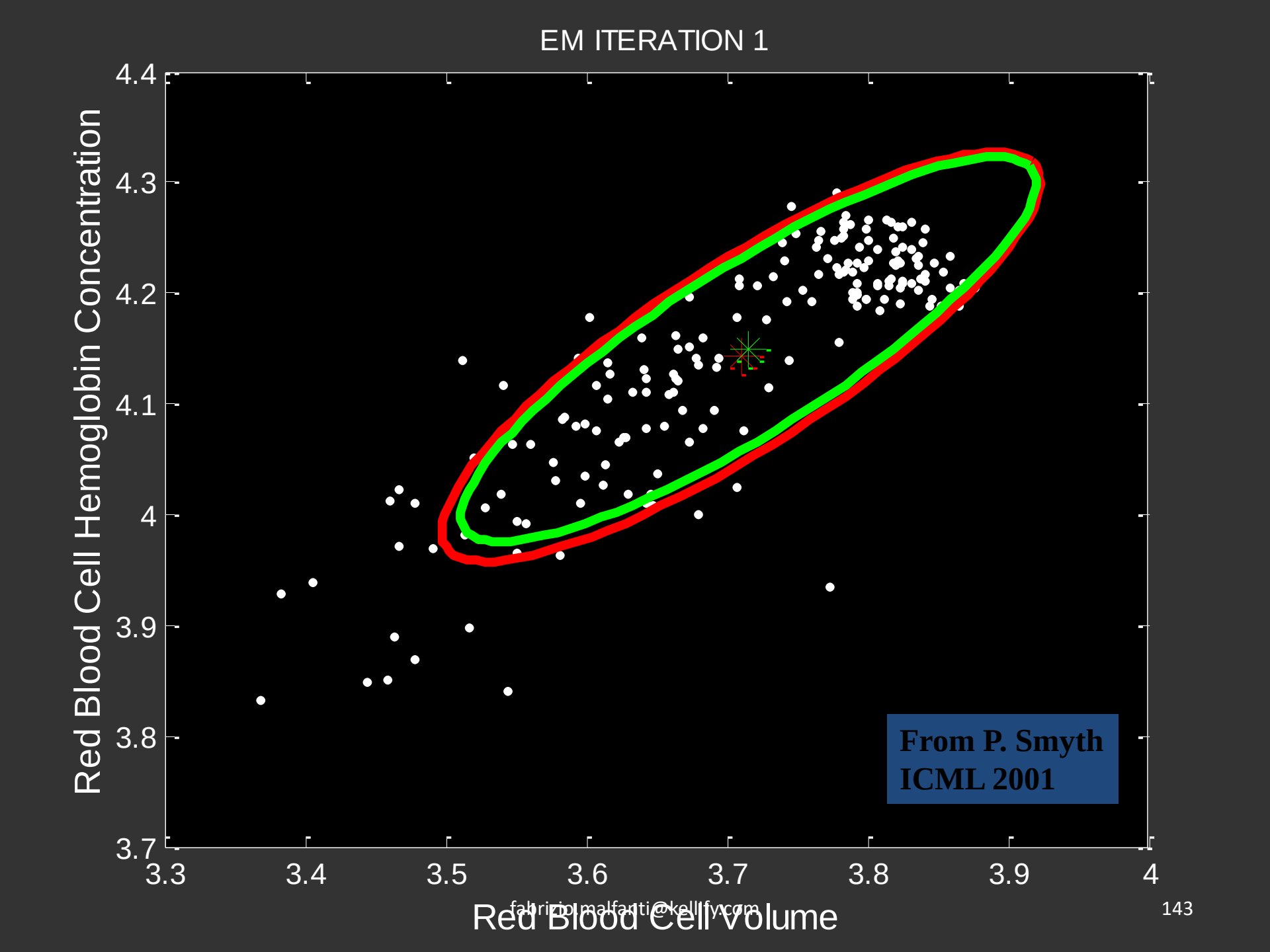
$$\mu_c = \frac{1}{N_c} \sum_i r_{ic} x_i$$

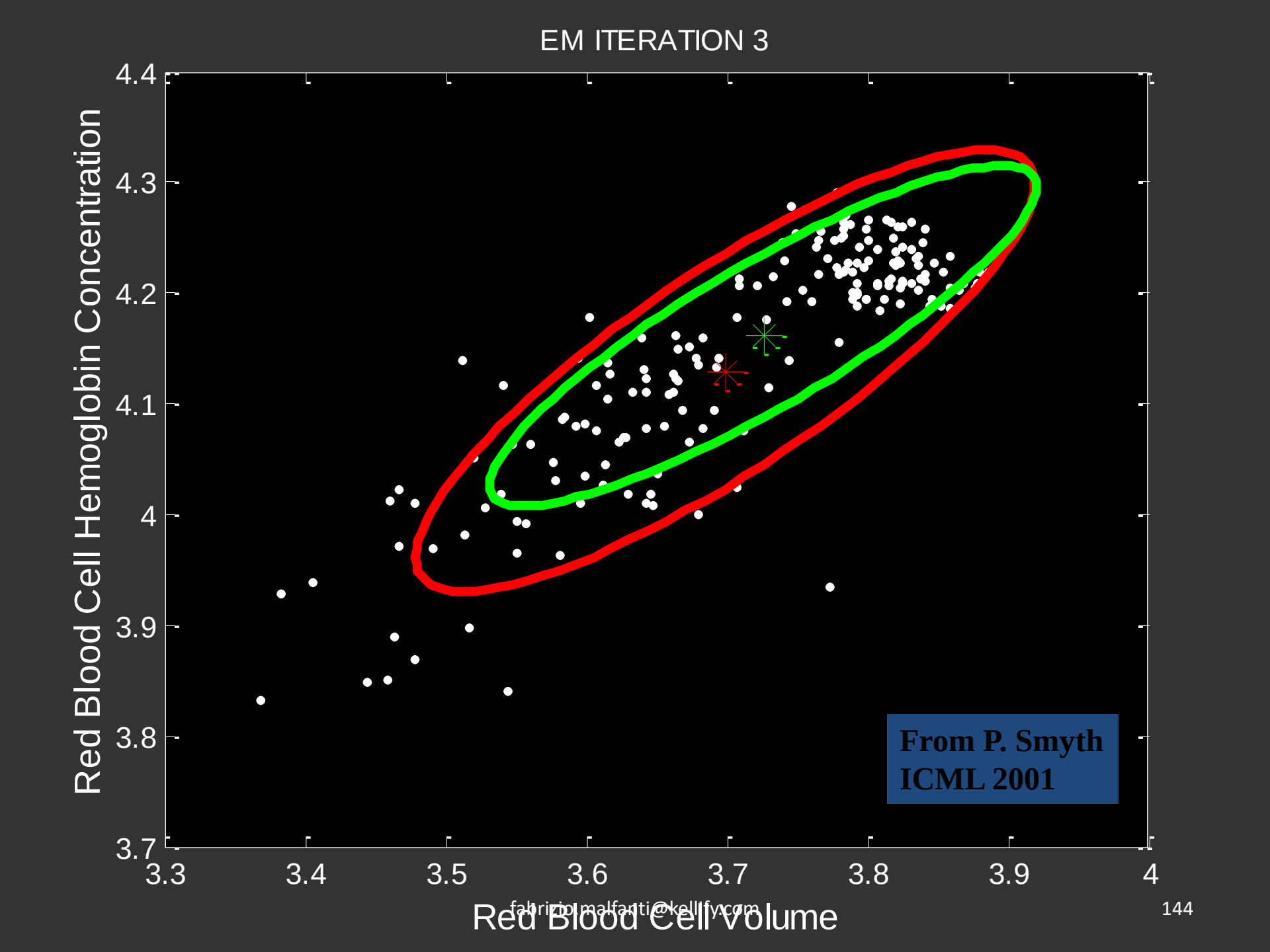
Weighted mean of assigned data

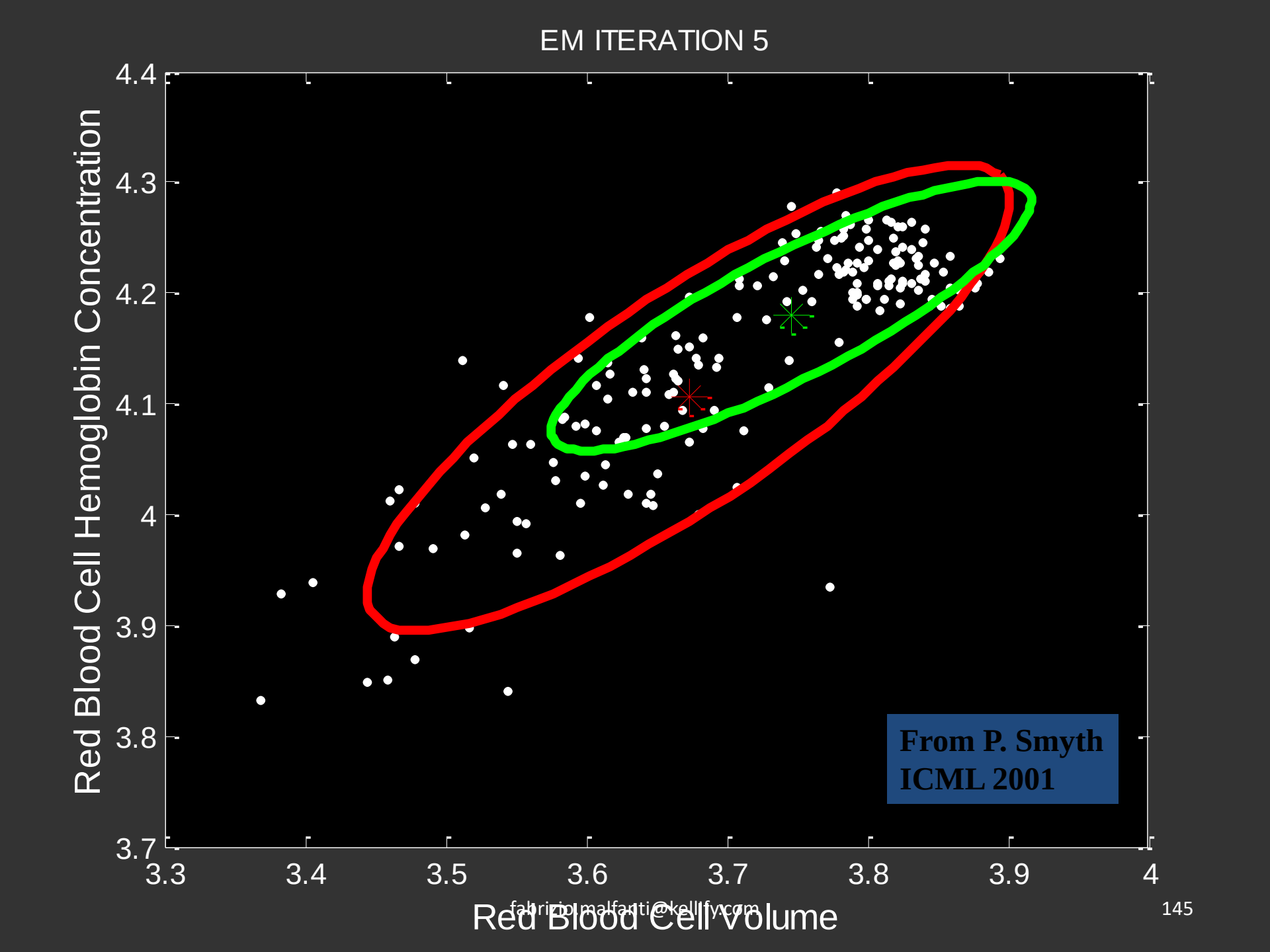
$$\Sigma_c = \frac{1}{N_c} \sum_i r_{ic} (x_i - \mu_c)^T (x_i - \mu_c)$$

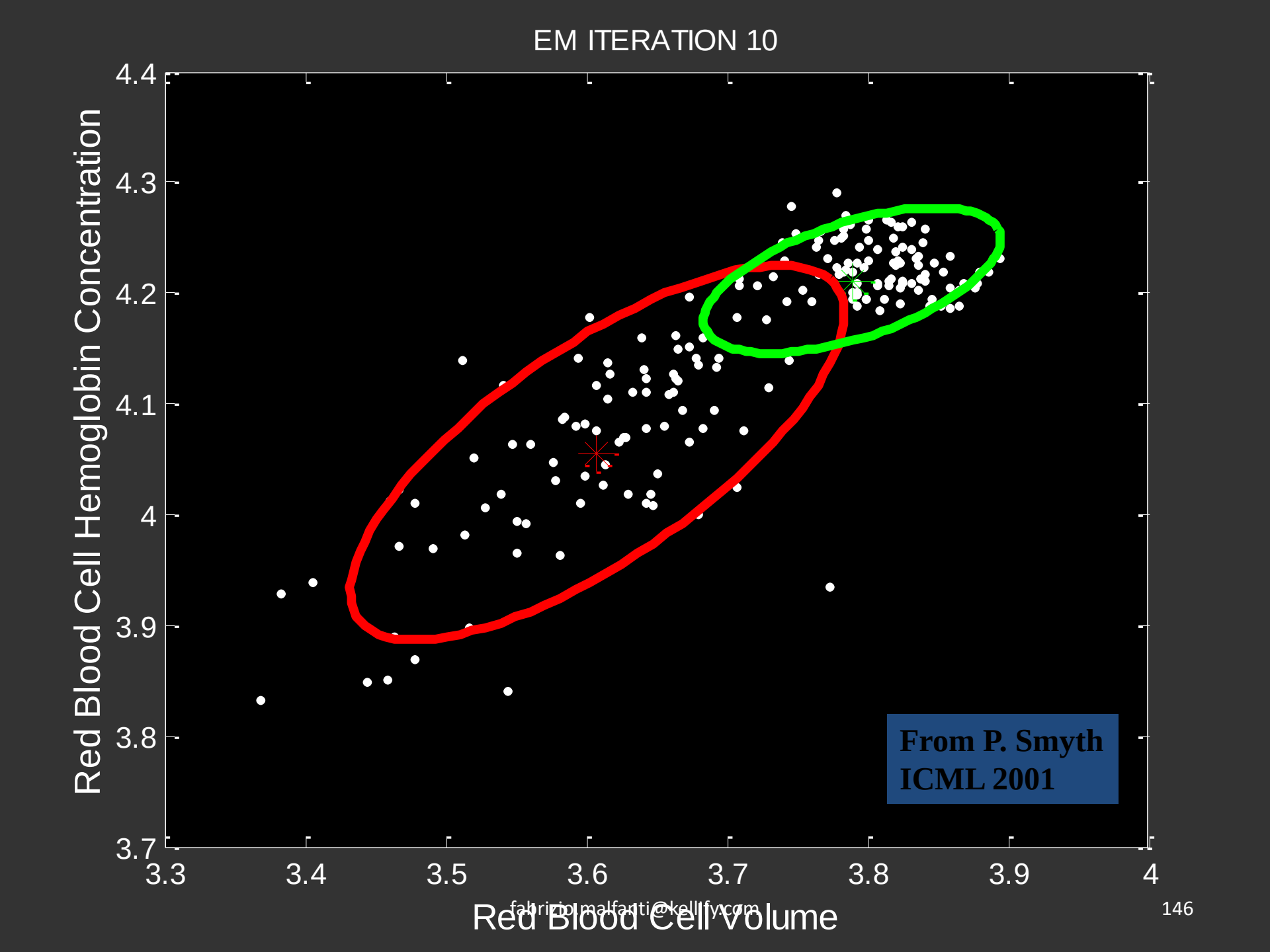
Weighted covariance of assigned data (use new weighted means here)

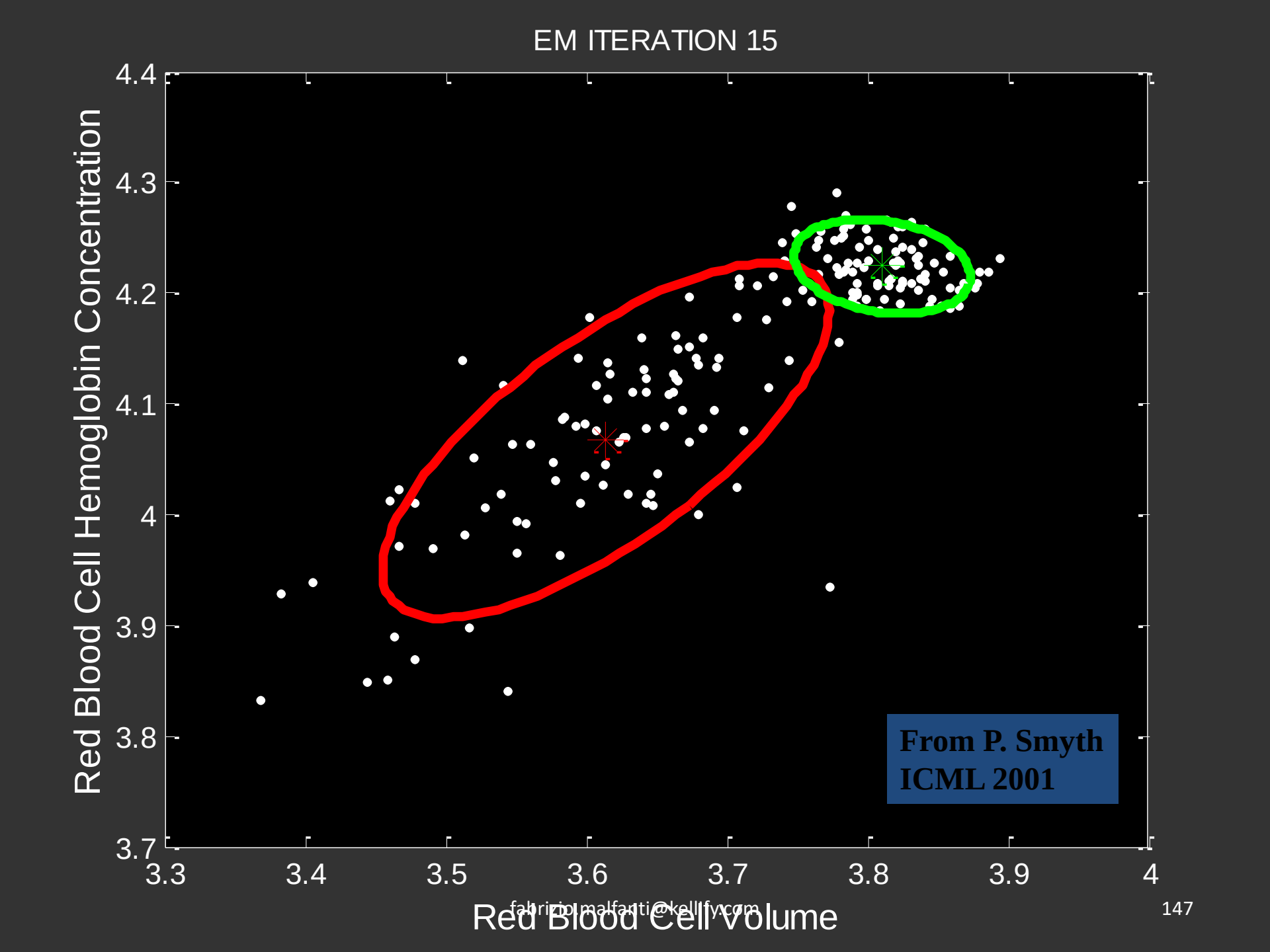

ANEMIA PATIENTS AND CONTROLS
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

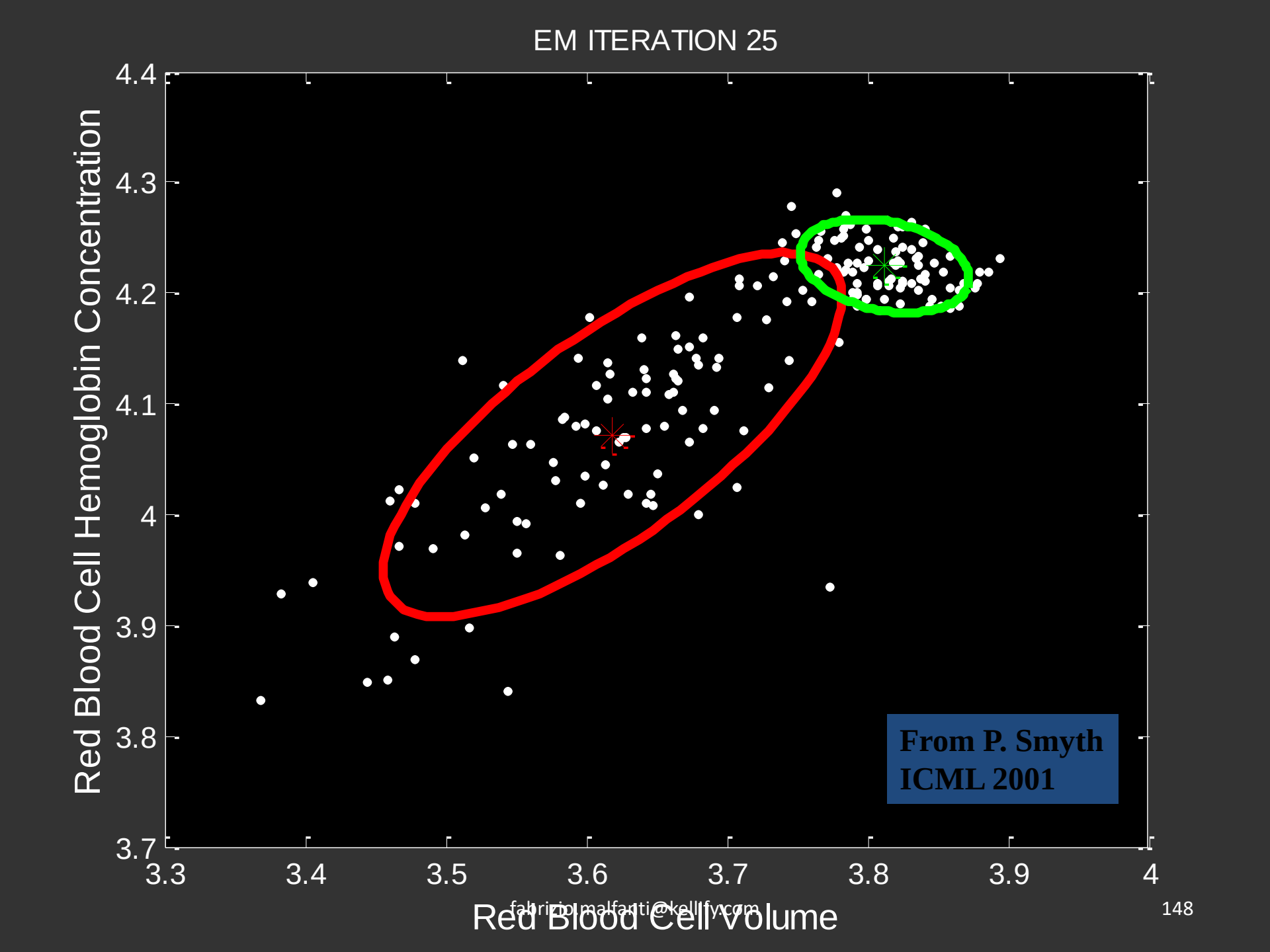

EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

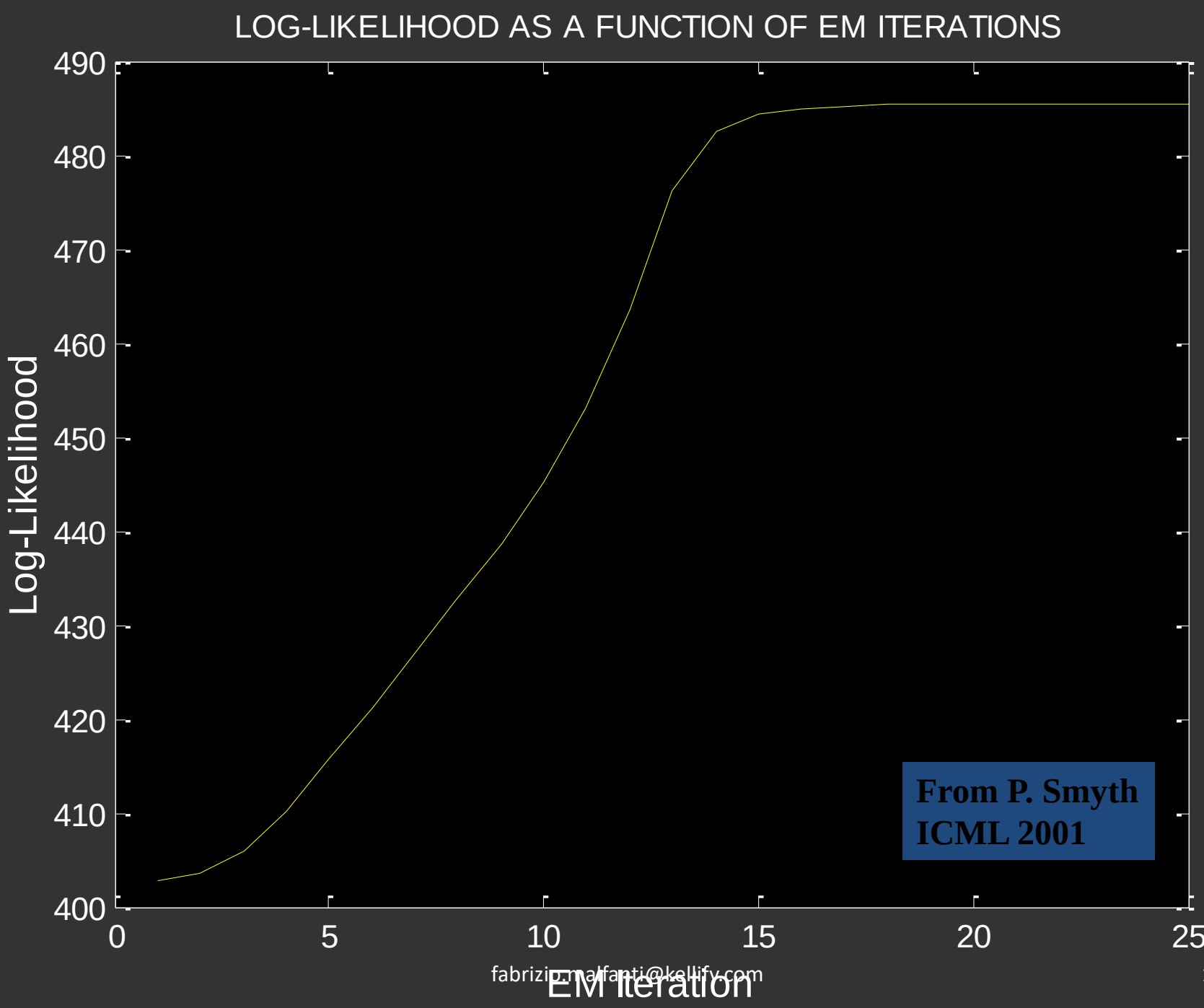

EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

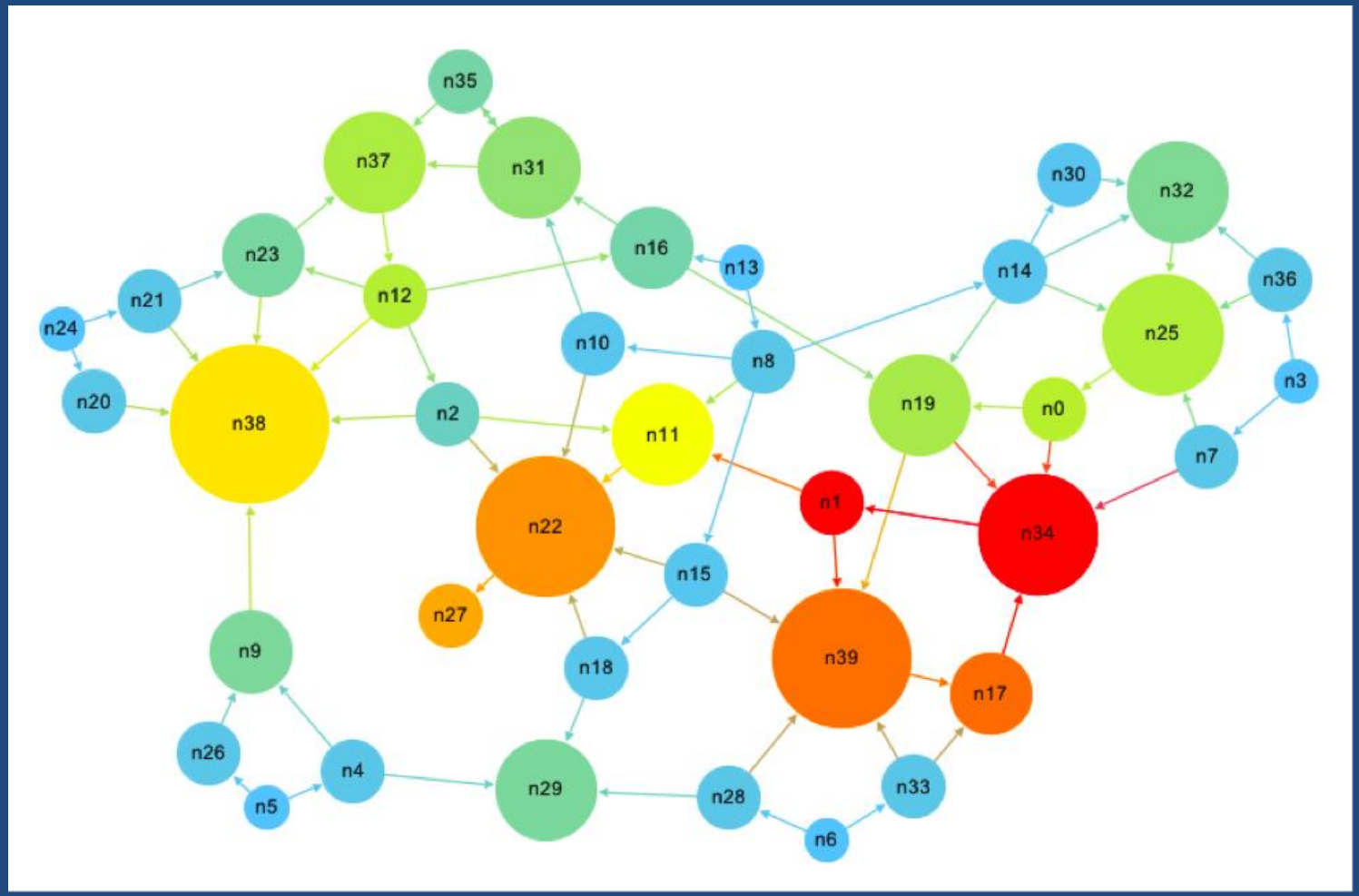

EM ITERATION 10
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 25
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS
Log-Likelihood
EM Iteration
490
480
470
460
450
440
430
420
410
400
0
5
10
15
20
25
From P. Smyth
ICML 2001

PageRank


n35
n37
n31
n30
n32
n23
n16
n13
n14
n36
n21
n12
n25
n24
n10
n8
n3
n20
n38
n2
n11
n19
n0
n7
n1
n34
n22
n15
n27
n9
n18
n39
n17
n26
n4
n29
n28
n33
n5
n6

Application of PageRank

- Ranking Tweets in Twitter
- Recommendation Systems
- Suggesting friends over Social Networks
- Predicting Ecological collapse, tracks species extinctions

Reinforced Learning

- Reinforcement learning is the problem of getting an agent to act in the world so as to maximize its rewards.
- For example, consider teaching a dog a new trick: you cannot tell it what to do, but you can reward/punish it if it does the right/wrong thing.
- It has to figure out what it did that made it get the reward/punishment, which is known as the credit assignment problem.
- We can use a similar method to train computers to do many tasks, such as playing backgammon or chess or scheduling jobs or more complex tasks.

Reinforced Learning

- Reinforcement learning is learning what to do--how to map situations to actions--so as to maximize a numerical reward signal.
- Reinforcement learning is defined not by characterizing learning methods, but by characterizing a learning *problem*.
- Any method that is well suited to solving that problem, we consider to be a reinforcement learning method.