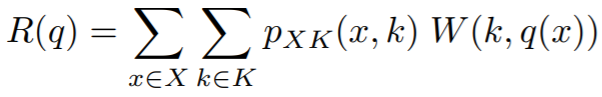
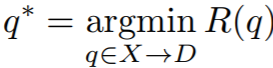
# ***I13 – Bayesovské rozpoznávání, odhady parametrů metodou maximální věrohodnosti. Učení bez učitele (shlukování, k-means). Regrese. Metoda expectation-maximization. (RPZ)***

# **Bayesovské rozpoznávání**

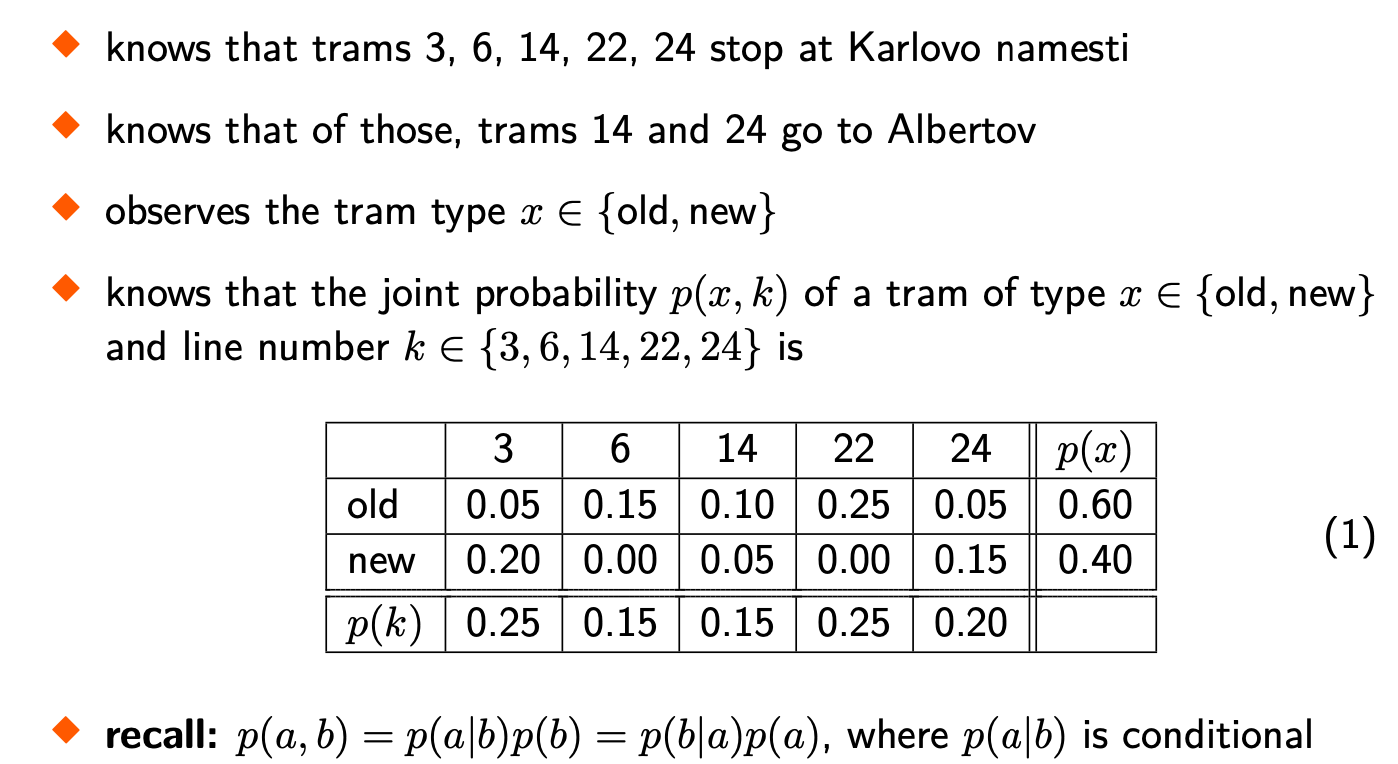
Založeno na minimalizaci Bayesovského risku *R*(*q*).

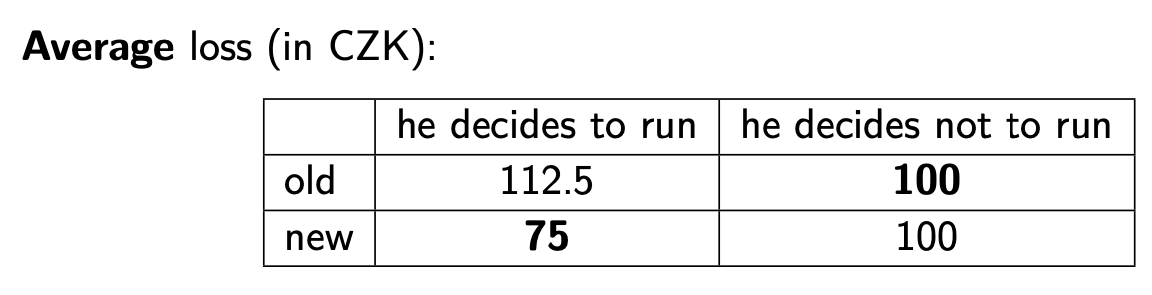
**Bayesovský risk: **

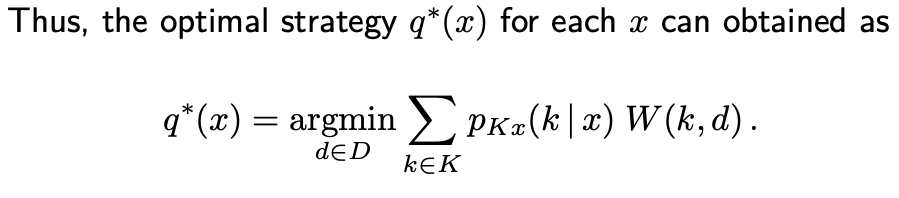
a **Bayesovská strategie** pak je:****

množiny: X … pozorování, D … rozhodnutí, K … třídy (skryté stavy)

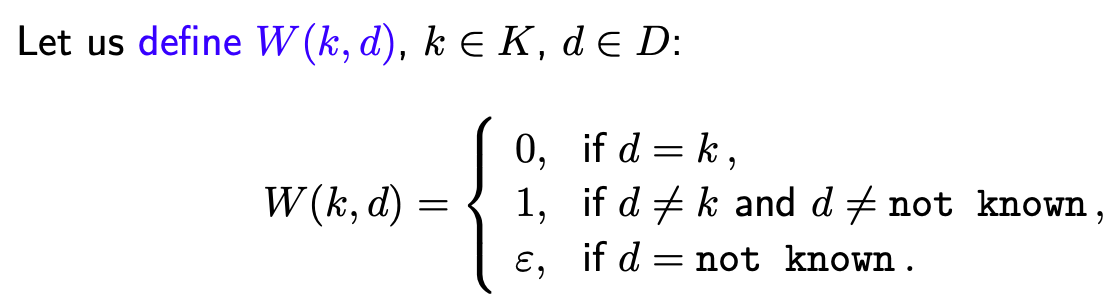
| Shrnutí: Bayesovské rozpoznávání hledá takovou strategii, která minimalizuje ztrátu vyjádřenou Bayesovským riskem. Při 0-1 ztrátové funkci přiřazuje pozorování do stavu s největší aposteriorní pravděpodobností. Nevýhody: Musí existovat smysluplná ztrátová funkce pro daný příklad (jaká je cena za lidský život, apod.), musí být známé apriorní pravděpodobnosti. |
| --- |

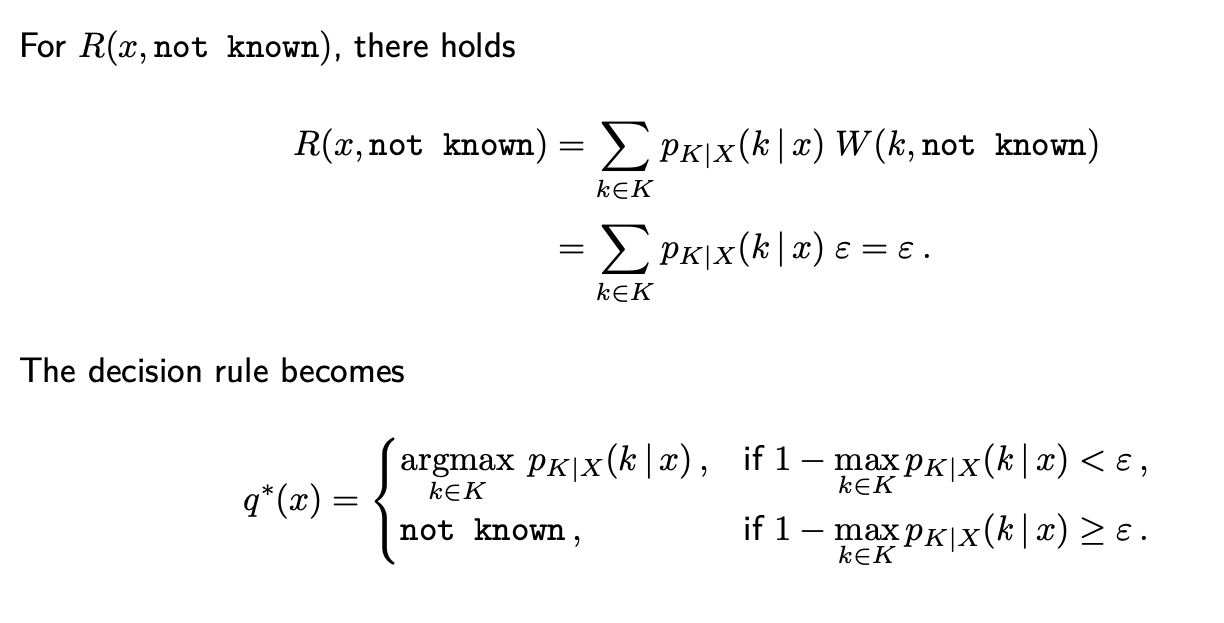


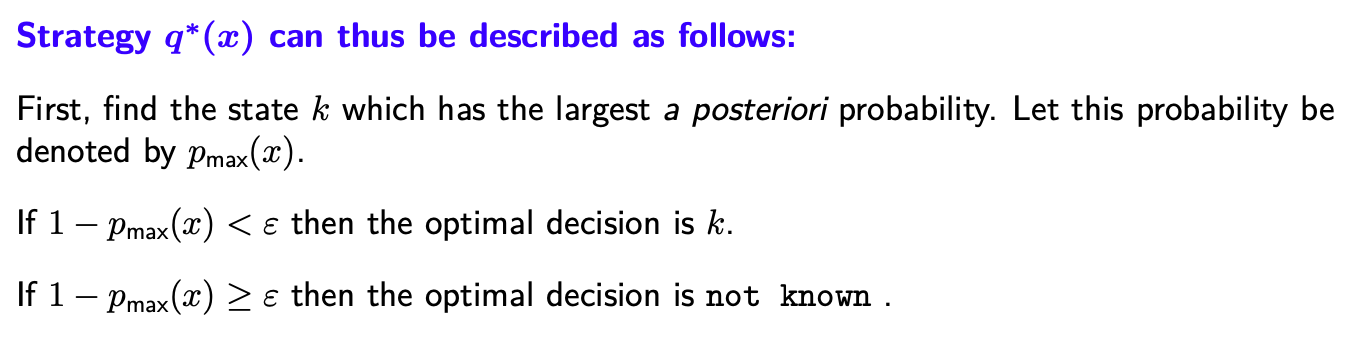




Reject option:



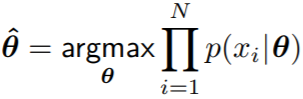




# **Odhady parametrů metodou maximální věrohodnosti**

Vycházíme z předpokladu, že známe “formu” rozložení pravděpodobnosti (např. gaussián, apod.),  
 odhadujeme parametry *θ* této “formy” (počet parametrů je nízký)

**Metoda maximální věrohodnosti (Maximum likelihood estimation = MLE):**

Postup matematicky:

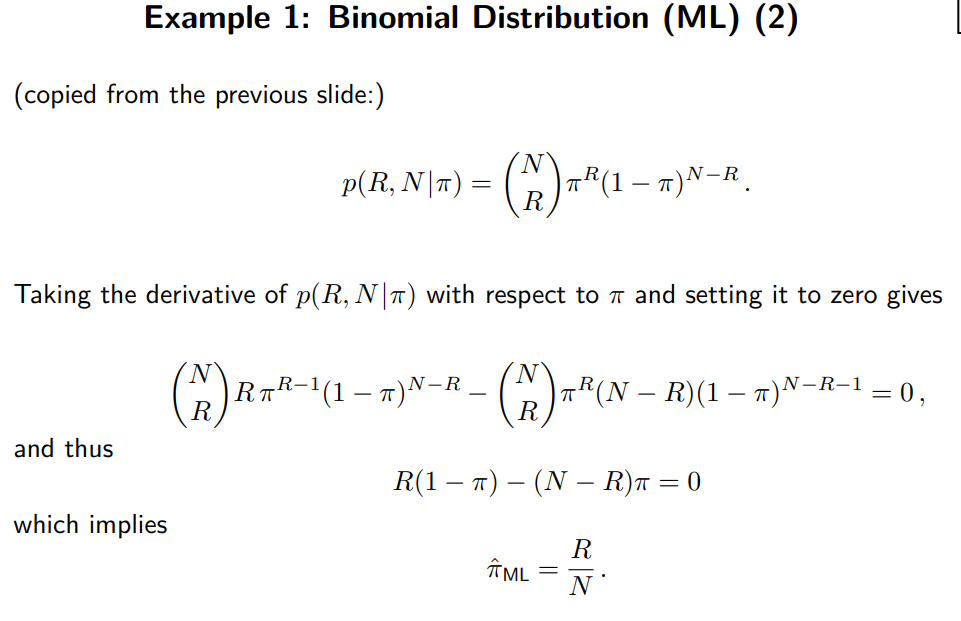
1. zlogaritmuji (likelihood → log-likelihood): z produktu je suma

*díky tomuto obvykle následně derivuji součet a ne součin – jednodušší*

1. zderivuji a položím rovno nule

Asymptoticky přechází ke správným parametrům.

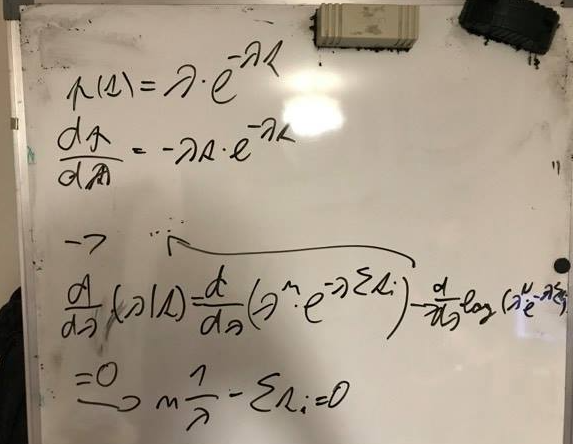
Odhad parametrů se dělá pro každou třídu zvlášť, postupně.



N = počet pokusů, R = kolikrát chci aby padla 1 na kostce třeba (2x .. )

# 

Exponenciální rozdělení:



# **Učení bez učitele (shlukování, k-means)**

Učení bez učitele (unsupervised learning) - pozorování nejsou olabelovaná

*Semi-supervised - používá se k učení jak olabelovaná, tak neolabelovaná pozorování*

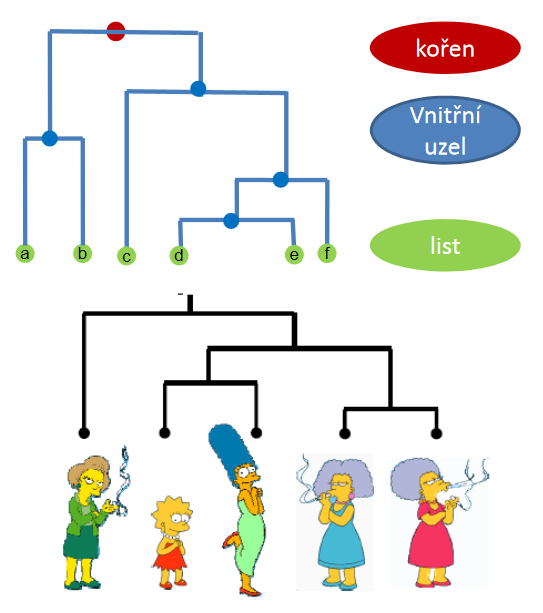
**Shlukování**

Shlukování se používá ke klasifikaci objektů do tříd. Dělí se na hierarchické (vytváří se systém podmnožin, z nichž některé mají další vnitřní členění, tedy podmnožiny nižší hierarchické úrovně) a nehierarchické (množiny jsou disjunktní). Pro příslušnost k daným shlukům je důležitá zvolená metrika, tj. způsob měření vzdálenosti mezi objekty, případně shluky.

Hierarchické metody shlukování zahrnují dva přístupy:

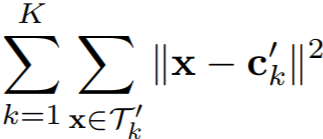
1. divizní - vychází se z celku, který se člení na menší části
2. aglomerativní - vychází se z jedinců, spojují se do větších celků

Dendrogram - graficky znázorňuje míru podobnosti v hierarchicky shlukované množině, např.:



Hierarchické shlukování představuje např. metoda K-nejbližších sousedů (K-NN, ta je ale s učitelem), nehierarchická metoda je např. shlukování pomocí algoritmu K-means.

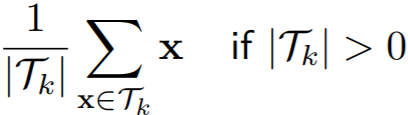
**K-Means**

Data *xi* chceme roztřídit do *k* tříd, hledáme *k* center shluků  
{ck\*} = argmin(ck)  (minimalizujeme vzdálenosti od center)

Algoritmus:

0. inicializace k centroidů (náhodně/k-means++ (viz. níže))

1. přiřazení bodů k nejbližšímu centroidu: 

2. spočítání nového těžiště shluku: ck = , pokud je Tk = ∅: reinit ck

3. pokud Tk = Tk+1 (stejné clustery jako předchozí iterace), tak konec, jinak goto 1

Lze zobecnit pro jakoukoliv metriku *d* (tedy ne jen normu *l*-2 = Eukleidovskou vzdálenost)

Můžeme uvíznout v lokálním extrému (lze řešit více běhy a výběrem nejlepšího řešení)

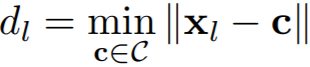
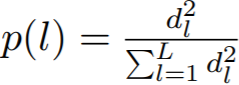
Algoritmus končí v konečném počtu iterací (je konečný počet stavů = NK)

Složitost: I(počet iter)·N(počet dat)·K(počet center)·M(dim dat)

V každé iteraci porovnávám každý bod s každým shlukem → N·K

K++

Inicializace pomocí K++ means zvyšuje pravděpodobnost nalezení globálního minima

Algoritmus: c1 vybrán úplně náhodně, , , dokud nemám K center

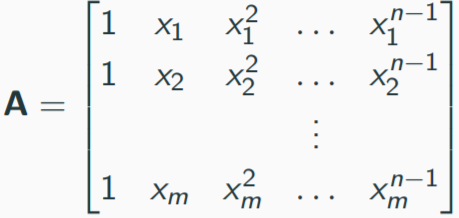
# **Regrese**

**Lineární regrese** je matematická metoda používaná pro proložení souboru bodů v grafu přímkou. O bodech reprezentujících měřená data se předpokládá, že jejich x-ové souřadnice jsou přesné, zatímco ypsilonové souřadnice mohou být zatíženy náhodnou chybou, přičemž předpokládáme, že závislost y na x lze graficky vyjádřit přímkou.

Lineární regrese představuje aproximaci daných hodnot přímkou metodou nejmenších čtverců. Pokud tuto přímku vyjádříme rovnicí y = b1 + b2x, jedná se o nalezení optimálních hodnot koeficientů b1 a b2.

V obecnějším případě může lineární regrese znamenat aproximaci daných hodnot (xi, yi) takovou funkcí y=f(x,b1,b2).

V OPT, přeurčená soustava vedla na regresi, takto: Měli jsme data (měření, známá čísla), z těch bylo nejprve potřeba sestavit matici A. Jednotlivá měření korespondují s řádky matice A. Maticí A budu násobit vektor neznámých parametrů, které se snažím odhadovat. Na pravé straně rovnice je vektor výsledků měření *y*. Matlab vyřeší soustavu pomocí příkazu: θ = A\y (y je vektor “výsledků” měření)

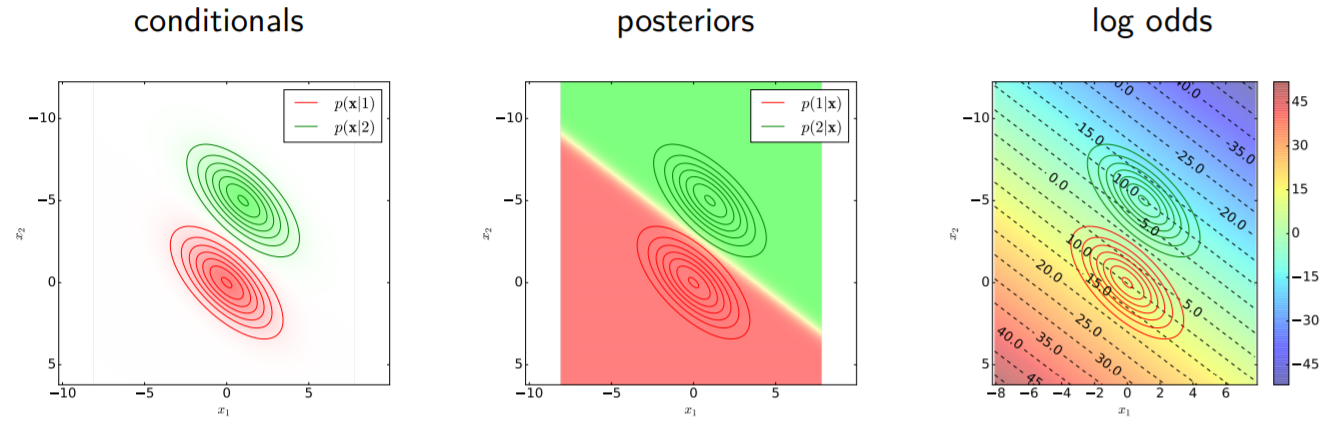


Interpolace polynomem:  
 

**Logistická regrese –** *Z RPZ:*

Lineární klasifikátor, log odds:

  
 a obrácené znaménko pro *q*(*x*) = 2



Pokud je poměr lineární, přecházíme na problém hledání parametrů dělicí nadroviny: w=[w0,w], x = [1,x]

wTx > 0 → q(x)=1

Prior probabilities způsobí jen posun (jsou obsaženy v členu w0)

Log odds:  → p(1|x) = p(2|x) exp(w·x), p(2|x) = p(1|x) exp(−w·x)

→ 1 = p(1|x) + p(2|x) = p(1|x) (1 + exp(−w·x)) = p(2|x) (1 + exp(w·x))

k : {-1,1} →  (sigmoida)

Algoritmus:

0. upravit trénovací množinu: {(xi, yi)}, x z RD, y = +-1, na {yi\*(xi,1)}. Init w z RD+1

1. minimalizujeme funkci E(w) = -l(w) = Sumi ln(1 + e-wx)

Vypočítáme gradient ∇wE = - Sumi x\*e-wx/(1 + e-wx)

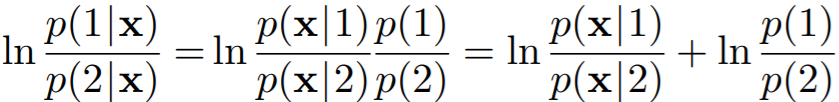
+

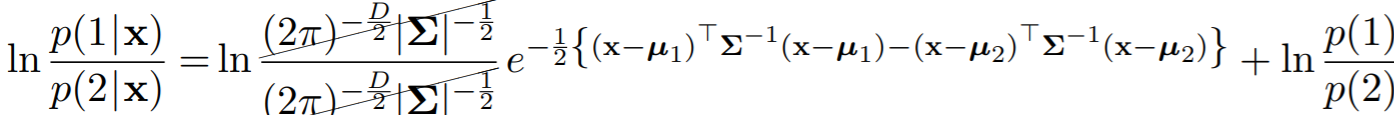
Updatujeme w:= w - μ\*∇wE (jdeme proti směru grad, protože minimalizujeme E)

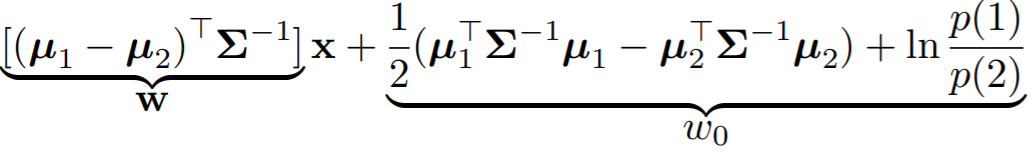
Pokud E(wnové) < E(wstaré): step = 2\*step, jinak: step = step/2 a neupdatuj w,g,E

2. goto 1

*Diskriminační funkce pro 2 třídy se stejnou* ***kovarianční maticí***



* 
  + *ln(ex)=x ; maticově roznásobit ; xTΣ-1x se požerou ; xTΣ-1μ = μTΣ-1x (protože Σ je symetrická: Σ = ΣT)*

→ 

→ **w**·**x** + w0 , kde **w** ∈ RD, w0 ∈ R

Závěr: Pro 2 třídy s norm. rozdělením se stejnými kovariančními maticemi: log odds je lineární funkce

*Kovarianční matice: → vl. čísla (a vektory): vidím osy elipsy Gausse, velké vl. č. = velká var v daném směru*

# **Metoda expectation-maximization**

EM je iterační metoda pro hledání MLE nebo MAP odhadu parametrů. (MLE = maximálně věrohodný odhad, MAP = maximální aposteriorní pravděpodobnost).  
Při EM iteracích se pravidelně střídají kroky výpočtu střední hodnoty (očekávání, E) s kroky maximalizace (M). V kroku E se vytváří očekávaná logaritmická věrohodnostní funkce na základě aktuálního odhadu parametrů. V kroku M se počítají parametry maximalizující očekávanou logaritmickou věrohodnostní funkci nalezenou v kroku E.

Odhady parametrů se používají pro určení rozdělení skrytých proměnných (latent variables).

EM se často používá pro shlukovou analýzu ve strojovém učení a počítačovém vidění.

Použití:

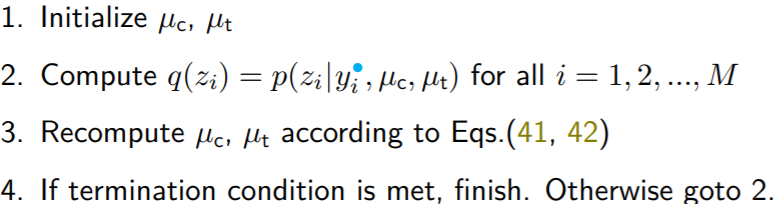
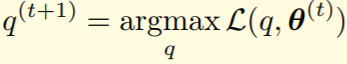
* Missing data - měření jsou nekompletní (pro některé *x* chybí nějaká složka)
* Latent variables - měření kompletní, ale bylo by mnohem lepší kdyby byla ještě nějaká informace, např. mix více rozdělení typicky

Jako k-means (nemáme informaci o třídě).

Častá aplikace: odhad mixu Gaussů

*Dále, viz. I12 – EM Algorithm.*

Algoritmus (S2: “E-step”, S3: “M-step”)

 (*μc, μt* viz. níže (2) )  
, 

**E-step** počítá pravděpodobnost, že daný bod je generovaný danými komponenty (jeho aposteriorní pravděpodobnost, že patří k dané třídě)

**M-step** počítá neznámé parametry analyticky (odhad parametrů pomocí MLE/MAP)

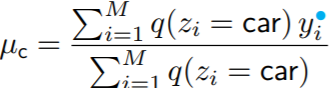
je užitečný a efektivní dále pro exponenciální rozdělení

chytře maximalizuje likelihood tím, že tlačí lower bound nahoru

iterativní metoda ⇒ nemusí skončit v globálním maximu

opatrně inicializovat (stejně jako u K-means, NNs, …)

*Příklad car/truck:*

* *0. init μc, μt*
* *1. *
* *2. μc, μt: *
* *ukončit pokud je termination condition splněna*
* *k-means se liší pouze v bodě 1 - tam se přiřazuje pouze 1 nebo 0*