P7

score metric for (x, y) word pair: $\underline{\underline{S_{(a,b)}(x,y)}}$ baseline $= \begin{cases} \cos(\vec{a}-\vec{b}, \vec{x}-\vec{y}) & \text{if } \|\vec{x}-\vec{y}\| \le \delta \\ 0 \end{cases}$

seed direction (she, he)

word $\vec{w} \in \mathbb{R}^d$, $\|\vec{w}\| = 1$

δ: 1th experiment; similarity th

Analogy

Direct gender bias

Definition: $\left[\text{Direct-bias}_c = \frac{1}{|N|} \sum_{w \in N} |\cos(\vec{w}, g)|^c\right.$

N: gender neutral word set

c: strictness

?? confused here. projection

g: [gender subspace top principle component]

indirect Bias Definition: $\beta(w, v) = \left(w \cdot v - \frac{w_\perp \cdot v_\perp}{\|w_\perp\|_2 \|v_\perp\|_2} \right) / w \cdot v$

Described in next page

$w_\perp = w - w_g \;;\; v_\perp = v - v_g$ $\quad \boxed{w_g = (w \cdot g)\, g}$

$\|$ if 0 means no projection.

$\Downarrow$

$\beta(w, v) = 0$

if, $w_g = w \Rightarrow w_\perp = 0$

then $\Rightarrow \beta(w, v) = 1$ ✓

B subspace $\{b_1, \ldots, b_k\} \in \mathbb{R}^d$ (unit vector, orthogonal to each other) $\| k = 1 \Rightarrow$ vector.

Debiasing Algorithm: Projection Direction $\bar{v}_B = \sum_{j=1}^{k} (v \cdot b_j)\, b_j$ (original vector)

& value $= \underline{v^T B}$ $\;$ // orthogonal project: $\underline{\underline{v - v_B}}$

step 1: Identify gender subspace: Defining sets?? $D_1 \ldots D_n \subset W$ (total words)

mean of $D_i \Rightarrow \mu_i := \sum_{w \in D_i} \vec{w} / |D_i|$. $\quad \{\vec{w} \in \mathbb{R}^d\}$ word vector

$k \geq 1$

Let Bias subspace

$C := \sum_{i=1}^{n} \sum_{w \in D_i} (\vec{w} - \mu_i)^T (\vec{w} - \mu_i) / |D_i|$ $\quad$ [k row of SVD is Bias subspace B]

P7

new embedding ↘ orthogonal projection

Hard de-biasing: $\vec{w} := (\vec{w} - \vec{w}_B)/\|\vec{w} - \vec{w}_B\|$ // re-embedding definition.

got B matrix earlier ✓ projection

how ?? → step

; word to Neutralize $N \subseteq W$

family/Equality set $\mathcal{E} = \{E_1, E_2, \dots, E_m\}$ // ?? what we want equidist.

$E_i \subseteq W$

↳ finally all words ~~will~~ have similar component in gender neutral Direction)

$\mu := \sum_{w \in E} w/|E|$

projection to B

$v := \mu - \mu_B$ // orthogonal

this term varies only for words in E set

For $\forall$ $w \in E$; $\vec{w} := \vec{v} + \sqrt{1 - \|v\|^2} \left[\dfrac{\vec{w}_B - \mu_B}{\|\vec{w}_B - \mu_B\|} \right]$ ↘ Added for the Bias component differences.

output subspace B, new embedding $\{\vec{w} \in \mathbb{R}^d\}_{w \in W}$

$(w_B, \; w_{\perp B} = w - w_B)$

Soft bias Correction: $W \in \mathbb{R}^{d \times |vocab|}$ new ✓

$T \rightarrow$ transformation $d \times d$

$$\min_T \|(TW)^T(TW) - W^T W\|_F^2 + \lambda \|(TN)^T(TB)\|_F^2$$

matrix size Vocab×vocab

optimization problem

[matrix of the Neural embedding words]

Measurement of indirect bias: Between two gender neutral words (w, v)

(measurement)

indirect bias $\beta(\vec{w}, \vec{v}) = \dfrac{\vec{w}^T \cdot \vec{v} - \dfrac{w_{\perp B}^T \, v_{\perp B}}{|w_{\perp B}^T| \, |v_{\perp B}|}}{w^T \cdot v}$ ⇒ [match in gender Independent direction]

overall match.

P2

simplified version of pagerank:

$$\boxed{R(u)} = c \sum_{v \in B_u} \frac{\boxed{R(v)}}{(N_v)}$$

$(N_v) \rightarrow$ Number of links from u

Back link of u (towards u)

$\rightarrow$ Normalization.

$\rightarrow$ Recursive equation.

A matrix $\rightarrow$ page $\times$ page big matrix.

$A_{u,v} = \frac{1}{N_u}$ if edge exists.

0 if no edge between u, v

eigen value

$R = c\, A R$

$\rightarrow$ A vector! (bad notation)

$\rightarrow$ eigen vector of A

$\rightarrow$ Eigen vector of A $\rightarrow$ symmetric matrix.

$\downarrow$

All evectors are orthogonal.

$\rightarrow$ Dominant eigenvector

$\rightarrow$ Power iteration.

Ranksource modification.

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + c\, \underbrace{E(u)}_{\text{source of rank vector.}}$$

L1 Norm, $\|R'\|_1 = 1$, c is maximized.

if $E(u) \geq 0$ the c is reduced.

Decay factor.

$\boxed{\text{All 1's matrix}}$

$$\Rightarrow R' = c(AR' + E) = c\,\underbrace{(A + E \times \mathbf{1})}_{\text{Eigen value of this one.}} R' \quad /\!/ \text{ since } \|R'\|_1 = 1$$

P9

$R_0 \leftarrow S$ // Random ints.

loop: (Dominant) finding Eig vector for A

→ $R_{i+1} \leftarrow AR_i$ // power iteration (PE)

→ $d \leftarrow \|R_i\|_1 - \|R_{i+1}\|_1$ // constrainted.

→ $R_{i+1} \leftarrow R_{i+1} + dE$ // little move from PE

increases convergence,

maintain $\|R\|_1$

// Normalize.

→ $\delta \leftarrow \|R_{i+1} - R_i\|_1$

while $\delta > e$ // convergence

P12: Recommendation problem formulation.

$C \to$ set of users (user space, $\to$ name, age, demograph, ..)

$S \to$ possible items. (name, title, producers, etc)

u, utility function, usefulness between (user, item)

$u : C \times S \to R$ (rating value $\to$ utility)

so objective, $\forall c \in C$, $\boxed{s_c' = \arg\max_{s \in S} u(c, s)}$

Content based filtering methods:

focus on $\boxed{u(c, s_i)}$ · user already has rated.

$\hookrightarrow$ similar to previous s_i will be recommended

Term frequence $TF_{ij} = \dfrac{f_{i,j}}{\max_z f_{z,j}}$; $f_{ij} \cong$ no time k_i word appear in document, d_j

inverse document freq $IDF_i = \log \dfrac{N}{n_i}$; $N \to$ total documents.

$n_i \to$ k_i appeared in how many documents.

for weight of keyword k_i, in document d_j

$$w_{ij} = TF_{ij} \times IDF_i$$

for content of document d_j for all the key words k.

$$\text{Content}(d_j) = (w_{1j}, w_{2j}, \dots, w_{kj})$$

P12

content based profile (c) = $\{w_{c1}, w_{c2} \ldots w_{ck}\}$ → for keyword (k) in system. → for user $c \in C$

The utility function $u(c,s)$ = score(content based prof (c), content(s))

$$\Rightarrow \boxed{u(c,s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\bar{w}_c \cdot \bar{w}_s}{\|\bar{w}_c\|_2 \times \|\bar{w}_s\|_2}}$$

Collaborative method:

$u(c,s) \leftarrow u(c_j, s)$; $c_j \in C$ & $c_j \approx c$

(similar user group)

(peer)

(i) memory based / Heuristic.

rating, $r_{c,s}$ = $\boxed{\underset{c' \in C}{\text{aggregate}}}$ $r_{c',s}$ // impute unknown value.

not given but estimated ; given ratings

⇓ Agg. function can be ?

$$r_{c,s} = \begin{cases} \text{(a)}\ \frac{1}{N} \sum_{c' \in C} r_{c',s} \\ \text{(b)}\ k \sum_{c' \in C} sim(c,c') \times r_{c',s} \\ \text{(c)}\ \bar{r}_c + k \sum_{c' \in C} sim(c,c') \times (r_{c',s} - \bar{r}_{c'}) \end{cases}$$

more like collaboration term.

where, $k = \dfrac{1}{\sum_{c' \in C} |sim(c,c')|}$ (Normalizing constant)

$\bar{r}_c = \left(\dfrac{1}{|S_c|}\right) \sum_{s \in S_c} r_{c,s}$, where $S_c = \{s \in S \mid r_{c,s} \neq \phi\}$

(Average rating)

P 12

Pearson coefficient based similarity

$$sim(x,y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{x,y}} (r_{x,s} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2}}$$

// Iterate over (s) item

Alternatively, Cosine based similarity.

for each item (s)

(ii) model based Algorithm:

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times Pr(r_{c,s} = i \mid r_{c,s'}, s' \in S_c)$$

P 13 Selected dimensionality f:

$q_i \in \mathbb{R}^f$ // item

$p_u \in \mathbb{R}^f$ // user.

interaction between user u & item i

the approx. rating $\hat{r}_{ui} = q_i^T p_u$ —— (1)

How to get it?

SVD? but empty elements??

Imputation → bad idea.

So, optimization problem:

$$\min_{p, q} \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u) + \lambda (\|q_i\|^2 + \|p_u\|^2) \quad \text{—— (11)}$$

set of given training set (previously observed)
(Explicitly feedback points)

Learning methods

(i) SGD

$e_{ui} := r_{ui} - q_i^T p_u$

$q_i \leftarrow q_i + \gamma (e_{ui} \cdot p_u - \lambda q_i)$ // Gradient descent

(−) is changed by internal calculation

$p_u \leftarrow p_u + \gamma (e_{ui} \cdot q_i - \lambda p_u)$ + fast

(ii) Alternate // ALS

To solve nonconvexity → fix one, and solve for the other.

P 13

Existance of product/user bias.

modify eq (i) by $\boxed{b_{ui} = \mu + b_i + b_u}$

So, $r_{ui} = \underbrace{\mu}_{\text{overall reating}} + b_i + b_u + q_i^T p_u$ — (iv)

Now the optimization problem changes to,

$$\min_{p,q,b} \sum_{(u,i)\in K} \left(r_{ui} - \mu - b_u - b_i - q_i^T p_u\right)^2 + \lambda\left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2\right) \quad \text{— (v)}$$

// may bias the model.

Additional input source: cold start overcome.

$N(u) \to$ implicit preference on items by the users.

$\boxed{x_i \in \mathbb{R}^f}$ // item association.

$\sum_{i\in N(u)} x_i$ // sum of implicit preference

Normalization $\Rightarrow |N(u)|^{-0.5} \sum_{i\in N(u)} x_i^{0.5}$ // empirical.

user Attributes $\underset{\text{set}}{A(u)} \Rightarrow \underset{\text{elements.}}{\boxed{y_a \in \mathbb{R}^f}}$ Associated factors to the Attributes.

Now overall: $r_{ui} = \mu + b_i + b_u + q_i^T \left[p_u + \underbrace{|N(u)|^{-0.5} \sum_{i\in N(u)} x_i + \sum_{a\in A(u)} y_a}_{\text{two extra terms.}}\right]$ — (6)

P13

Temporal dynamics:

including time $\hat{r}_{ui}(t) = \underbrace{\mu}_{\text{static}} + \underbrace{b_i(t)}_{\text{item bias}} + \underbrace{b_u(t)}_{\text{user bias}} + \underbrace{q_i^T}_{\text{static}} \, p_u(t)$

Dynamic

human behaviour dynamics.

input with confidence level:

$$\min_{p,q,b} \sum_{(i,u)\in K} \underbrace{c_{ui}}_{\text{modified confidence term}} \left(r_{ui} - \mu - b_u - b_i - p_u^T q_i\right)^2 + \lambda\left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2\right)$$

⇒ modified confidence term.

(Variational Lower bound)

$z \rightarrow x$ — $p(x)$ → Prob dist. over variable

Latent, observed. — $p(x)$ → Pdf of Distribution x

Posterior $$p(z|x) = \frac{p(x|z)\,p(z)}{\int_z p(x|z)\,p(z)}$$

Derivation 1: $\log p(x) = \log \int_z p(x,z)$ // (-) of information $p(x)$ value

(-) or 0 max. $= \log \int_z p(x,z) \frac{q(z)}{q(z)} dz$

$\geq E_q\left[\log \frac{p(x,z)}{q(z)}\right]$ // Jensen's inequality

Reorganizing

$L = E_q[\log p(x|z)] - KL[q(z)||p(z)]$ ELBO, $L = E_q[\log p(x,z)] + H_q(z)$ // entropy def.

|Negative| > |+ve| // so overall (-)

|Always else $p(x) = 1$|: no info.

interprete: (-) of information ≥ ELBO // reverse it (interesting)

No more info that -|ELBO| in $p(x)$

⇒ more info than |ELBO|

Derivation 2: $KL(q(z)||p(z|x)) = \int q(z) \log \frac{q(z)}{p(z|x)} dz$

(Backward KL)??

$= -L + \log p(x)$ // easy $p(z|x) = \frac{p(x,z)}{p(x)}$

fixed

$\therefore \boxed{\log p(x)} = L + \boxed{KL(q(z)||p(z|x))} \geq 0$

// missed margin of $q(z)$ to estimate $p(z|x)$ (failed) if large gap L & $p(x)$

if 0 then $L = \log p(x)$

interpretation:

By making elbo highest means $q(z) \approx p(z|x)$

successful posterior estimation.

P15

$$P(w_i) = \sum_{j=1}^{T} P(w_i \mid z_i)\, P(z_i = j)$$ // model itself

word

topic to word prob

topic prob.

No. Document → D
No. of topic → T

(Distribution over Dist.)

(multinomial probs are also from Dist → Dirichlet (conjugate prior))

D documents, T topics & W unique words.

$P(w \mid z=j) = \phi_w^{(j)}$ // multinomial distribution (T of them) (topic to word) (event count)

(k face dice n time, event count) what events? how many time?

(W words total W counts)

$P(z=j) = \theta_j^{(d)}$ // D multinomial (Document to topic) (D)

(T topics total D times)

Now the objective $\boxed{\text{Maximize } P(w \mid \theta, \phi)}$

Modified objective for dirichlet distribution.

LDA

max it $P(w \mid \theta, \phi) = \int P(w \mid \phi, \theta)\, P(\theta \mid \alpha)\, d\alpha$ // But intractable??

with dirichlet parameter (α) (as conjugate prior for multinomial)

determined by Variation bayes / Expectation propagation.

The complete model (Gibbs sampling usage opportunity)

$$\begin{cases} w_i \mid z_i, \phi^{z_i} & \sim \text{Discrete}(\phi^{z_i}) \\ \phi & \sim \text{Dir}(\beta) \\ z_i \mid \theta^{d_i} & \sim \text{Discrete}(\theta^{d_i}) \\ \theta & \sim \text{Dir}(\alpha) \end{cases}$$

w_i ... from earlier

ϕ ... // New [conjugate prior]

z_i ... Earlier

θ ... // New [conjugate prior]

α, β hyperparameter.

P15

By integrating w.r.t. θ, ϕ, $\Rightarrow$ $P(w, z)$

No document here

word w $\leftarrow$ j topic (Assign)

no. time

$$P(w,z) = P(w|z)\, P(z)$$

where,

$$P(w|z) = \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^T \prod_{j=1}^{T} \frac{\prod_w \Gamma(n_j^{(w)}+\beta)}{\Gamma(n_j^{(\cdot)}+W\beta)}$$

//see wiki multinomial Approx. $\rightarrow$ (2)

j topics

Gamma function.

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\, dx$$

similar

similarly,

$$P(z) = \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^D \prod_{d=1}^{D} \frac{\prod_j \Gamma(n_j^{(d)}+\alpha)}{\Gamma(n_\cdot^{(d)}+T\alpha)}$$

no of time word w from docu d Assign j topic

distinct from early. $\rightarrow$ (3)

$$\Rightarrow P(z|w) = \frac{P(w,z)}{\sum_z P(w,z)}$$

// Again Intractable T^n terms
so require Approximation

Solve it By MCMC. (Gibbs Sampling).
require $P(z_i|\underline{z}_{-i}, \underline{w})$

using the earlier (2) and (3) we get. (by cancellation)

$$P(z_i|\underline{z}_{-i}, \underline{w}) \propto \boxed{\frac{n_{-i,j}^{(w_i)}+\beta}{n_{-i,j}^{(\cdot)}+W\beta}} \cdot \boxed{\frac{n_{-i,j}^{(d)}+\alpha}{n_{-i,j}^{(\cdot)}+T\alpha}}$$

(prob of w_i under topic j) $\leftarrow$

$\rightarrow$ prob of topic 'j' in document 'd_i'

//just need counter

$n_{-i}^{(\cdot)} \rightarrow$ not include current assignment (z_i)

P15: for single sample: (using just count)

$$\hat{\phi}_j^{(w)} = \frac{n_j^{w} + \beta}{n_j^{(\cdot)} + \beta}$$

// for new word w and new topic z.

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_{\cdot}^{(d)} + T\alpha}$$

Solved by Gibbs Sampling

Alternates: Variational bayes (?), Expectation propagation (?)

Total hyperparameter: α, β, T → (Varied Across)

(α, β fixed in experiment)

T → Topic Number ??

↓

This is model selection?

Target: $P(w \mid T)$

w → All words

T → topic Number.

Approximated by $P(w \mid z, T) \Rightarrow P(z \mid w, T)$ posterior

P16 ①

Expected value. training data

$$\tilde{P}(f) = \sum_{x,y} \tilde{P}(x,y) f(x,y) \quad \sum_{x,y} \tilde{P}(x) P(y|x) f(x,y) = P(f)$$

empirical — $\frac{1}{N} \times$ No of time (x,y) appears. (training data)

calculate from data. (model)

$P(x) \sim x$ — empirical of x (training data) — given — indicator function — feature function

we requires $\tilde{P}(f) = P(f)$
train — model

Explicitly, $\sum_{x,y} \tilde{P}(x) P(y|x) f(x,y) = \sum_{x,y} \tilde{P}(x,y) f(x,y)$ // constraint equation
model — training data.

Key goal

$$C \equiv \{ P \in P \mid P(f_i) = \tilde{P}(f_i) \text{ for } i = 1 \dots n \}$$

Fig 1

Entropy: $H(x) = \sum_x P(x) \log \left[\frac{1}{P(x)}\right]$

All the 2 possible
Data prob

Conditional Entropy, $H(P) = -\sum_{x,y} \tilde{P}(x) P(y|x) \log P(y|x) \geq 0$
model

lower bound

sure model

(set of probabilities distribution) $H(P) \leq$ uniform case entropy. $= \log |y|$ = cardinality of y

Maximum Entropy: $P_* \in C$ satisfy $\tilde{P}(f) = P(f)$
train model
upper bound.

$$P_* = \underset{P \in C}{\arg\max}\, H(P) \quad \text{// uniform case}$$

⇓

well defined & unique.

⇒ To select a model from a set of prob. distributions, that has maximum entropy.

Parametric Form:

find $P_* = \underset{P \in C}{\arg\max}\, H(P)$ // primal optimization.

Lagrangian $\Lambda(P, \lambda) = H(P) + \sum_i \lambda_i (P(f_i) - \tilde{P}(f_i))$
(model) training (ground truth)

Now, $P_\lambda = \underset{P \in P}{\arg\max}\, \Lambda(P, \lambda)$

$\Psi = \Lambda(P_\lambda, \lambda)$ // max value. // Dual function.

P 15 II

model

Solving, $\underbrace{P_\lambda(y|x)}_{\text{model}} = \frac{1}{Z_\lambda(x)} \exp\left(\sum_i \lambda_i f_i(x,y)\right)$ unknown (solved later) λ^*

normalization w.r.t. y

model

$$\Psi(\lambda) = -\sum_x \tilde{p}(x) \log Z_\lambda(x) + \sum_i \lambda_i \tilde{p}(f_i)$$

??

Dual optimization find $\lambda^* = \arg\max_\lambda \Psi(\lambda)$ / maximize

$P_{\lambda^*} \rightarrow$ parametric form

Relation to maximum likelihood : Training data.

$$L_{\tilde{p}}(P) = \log \prod_{x,y} \underbrace{P(y|x)}_{\text{mode}}^{\tilde{p}(x,y)} = \sum_{x,y} \tilde{p}(x,y) \log p(y|x)$$

By definition, $\Rightarrow \Psi(\lambda) = L_{\tilde{p}}(P_\lambda)$ // See earlier section.

$P^* \in C$ with maximized entropy is parametric model from $P_\lambda(y|x)$ family that maximize the likelihood of training sample $\tilde{p}$

Table 1 Summary

Compute the Params : (i) $f_i(x,y) \geq 0$

Algo: Input iterative Scaling.

input: $f_1 \cdots f_n \rightarrow$ empirical $\tilde{p}(x,y)$

output λ_i^*, P_λ

1. $\lambda_i = 0 \ (i = 1 \cdots n)$
2. $i \{1 \cdots n\}$

See the algorithm

P19 $x_i \in \mathbb{R}^n$; $i = 1 \ldots l$ observations. → or. $y_i \in \{1, -1\}$

$f(\underline{x}, \alpha) \rightarrow$ Approximation

Expectation of test error.

parameter

$$\boxed{R(\alpha) = \int \frac{1}{2} \underbrace{|y - f(\underline{x}, \alpha)|}\, dP(\underline{x}, y)}$$ // true

$$R_{emp}(\alpha) = \frac{1}{2l} \sum_{i=1}^{l} |y_i - f(\underline{x}_i, \alpha)|$$ // Approximation.

Connection between them

$l \downarrow h \uparrow$ vc confidence. $\eta \downarrow$

Vapnik 1995

$$R(\alpha) \leq \underbrace{R_{emp}(\alpha) + \sqrt{\left(\frac{h\uparrow(\log(2l/h) + 1) - \log(\eta\uparrow/4)}{l\uparrow}\right)}}$$

Here, $0 \leq \eta \leq 1$; with prob $(1-\eta)$ holds.

set to minimize this bound

$l \rightarrow$ Example

$h \rightarrow$ vc dimension.

$\eta \uparrow$
$h \downarrow$ we want
$l \uparrow$

[vc confidence lower ⇒ the better] Approximation (may overfit)

The vc dimension: increases function capacity ↑ confidence boundary (higher is Bad)

infinite vc dimension $f(x, \alpha) = \theta(\sin(\alpha x))$, $x, \alpha \in \mathbb{R}$

$x_i = 10^{-i}$

y_i ... Assign anything.

$$\alpha = \pi\left(1 + \sum_{i=1}^{l} \frac{(1 - y_i) 10^i}{2}\right)$$

true
Approximate.

vc dimension = ∞

shattering depends on choice of points.

chose points that can be shattered.

Seperable Case

⑪ projection to HP = $\boxed{-b}$ bias

p19 satisfying hyperplane (HP): $\underline{w} \cdot x + b = 0$

& Linear

$\Downarrow$ normal to HP.

(All points projected as $-b$ amount) (nice)

Seperable case
$$\begin{cases} x_i \cdot w + b \geq 1 \; ; & \text{overshoot in projection } y_i = +1 \\ x_i \cdot w + b \leq 1 \; ; & y_i = -1 \end{cases}$$

$\Rightarrow$ $y_i (x_i \cdot \underline{w} + b) - 1 \geq 0$ //

introducing lagrangian $\alpha_i, i = 1 \ldots l$; multiply & add all. $\alpha_i \geq 0$

$$\boxed{L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot \underline{w} + b) + \sum_{i=1}^{l} \alpha_i}$$

primal problem.

$$\begin{cases} w = \sum_i \alpha_i y_i x_i & \text{// solution.} \\ \sum \alpha_i y_i = 0 \end{cases}$$

Now the Dual problem.

use $K(x_i, x_j)$ in nonlinear case

putting values $\Rightarrow$

$$\boxed{L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\underline{x_i} \cdot \underline{x_j})}$$

// Dual formulation

kernel idea.

KKT condition:

$$\frac{\partial}{\partial w_v} L_p = w_v - \sum_i \alpha_i y_i x_{iv} = 0 \quad ; v = 1 \ldots d$$

$$\frac{\partial}{\partial b} L_p = \cancel{y_i (x_i \cdot w + b) - 1 \geq} - \sum_i \alpha_i y_i = 0$$

$$y_i (\underline{w} \cdot \underline{x_i} + b) - 1 \geq 0 \qquad i = 1 \ldots l$$

$$\alpha_i \geq 0 \qquad \forall i$$

$$\alpha_i (y_i (\underline{w} \, x_i + b) - 1) = 0 \qquad \forall i$$

(if seperable)

P19 Non linear SVM:

$\phi : R^d \to \mathcal{H}$ Higher dimension Projection.

$$K(\underline{x}_i, \underline{x}_j) = \phi(\underline{x}_i) \cdot \phi(\underline{x}_j)$$

need $\underline{w}$ of $|\mathcal{H}|$ dimensional ??

How to use the kernel ??

→ we just Need dot product.

using the traindata
No of support vector

test phase → $f(x) = \sum_{i=1}^{N_s} \alpha_i y_i \phi(s_i) \cdot \phi(x) = \sum_{i=1}^{N_s} \alpha_i y_i \underbrace{K(s_i, x)}_{\text{only need this.}} + b$

Mercer's Conditions:- $(\mathcal{H}, \phi)$ $(d \to \mathcal{H})$

$K(\underline{x}, \underline{y}) = \sum_i \phi(\underline{x})_i \phi(\underline{y})_i$ mapping exists

if $\forall g(x)$ that satisfy.

$\int g(x)^2 dx$ is finite.

then

$\int K(x,y)\, g(x)\, g(y)\, dx\, dy \geq 0 \Rightarrow$ Positive Semidefinite PSD

Open Question: How to formulate ϕ ?

since vc dimension is $|\mathcal{H}|+1$ // in this case

so $|\mathcal{H}| \uparrow$ bad generalize.

Radial basis kernel:

$K(\underline{x}, \underline{y}) = e^{-\|\underline{x}-\underline{y}\|^2 / 2\sigma^2}$ // may be Infinite VC Dimension.

↳ two layer sigmoid NN.

1. Layer I N_s weights each d dimensional
2. Layer II N_s weights (α_i)

Finally Sigmoid.

Generalize
1 vs 1 classifier

F19

Nonseparable Case:

may cause error

$$\underline{x}_i \cdot \underline{w} + b \geq +1 - \varepsilon_i \quad ; y_i = +1$$

$$\underline{x}_i \cdot \underline{w} + b \leq -1 + \varepsilon_i \quad ; y_i = -1$$

$$\varepsilon_i \geq 0 \quad \forall i$$

→ if any $\varepsilon_i > 1$ error occurs.

$\sum_i \varepsilon_i$ = upper bound of traing error.

Dual problem:) $L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \underline{x}_i \cdot \underline{x}_j$

Subject to: $\boxed{0 \leq \alpha_i \leq C}$ → user parameter. (higher penalty to error)

$$\sum \alpha_i y_i = 0$$

Solution is $w = \sum_{i=1}^{N_S} \alpha_i y_i \underline{x}_i$

P20

skip gram model maximize $\frac{1}{T}\sum_{t=1}^{T} \sum_{\substack{-c \le j \le c \\ j \ne 0}} \log p(w_{t+j} \mid w_t)$

sequence of words $\{w_1, w_2, \dots, w_T\}$

So, maximizing $p(w_{t+j} \mid w_t)$ ↑

Defined as $P(w_O \mid w_I) = \dfrac{\exp({v'_{w_O}}^T v_{w_I})}{\sum_{w=1}^{W} \exp({v'_w}^T v_{w_I})}$

c ↑
training time ↑
Accuracy ↑

v'_{w_O} = output vector reps.
v_{w_O} = input vector reps

$W=1$ ↦ All the words ?? [huge computation]

v_{w_O} → vector representation of w_O (via Network).

W → huge size !!

Each word has two reps
⇒ input v_w &
⇒ output v'_w

Hierarchical Softmax: Need $\log_2 W$ nodes.

$$p(w \mid w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = ch(n(w,j))]\!] \, {v'_{n(w,j)}}^T v_{w_I} \right)$$

1 if true, else 0

→ care about input/output representation. [computation ∝ $L(w)$]

Negative Sampling should be high (interesting)

NEG objective ⇒ $\underbrace{\log \sigma({v'_{w_O}}^T v_{w_I})}_{\text{row of matrix}} + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^T v_{w_I}) \right]$

positive

$P_n(w)$ → different choices

negative word how (neg)

⇓ should be low & negative

$\sigma(x) = \dfrac{1}{1+\exp(-x)}$

//modified NCE

P 20 II

Subsampling of freq words.

Discarding prob $P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$ → word frequency.

// Discarding prob.

$f(w_i) \uparrow \quad P(w_i) \uparrow$

$f(w_i) \downarrow \quad P(w_i) \downarrow$

(High freq words discarded more)

Balance between rare & frequent words.

Bigram skip:

(Phrase selection)

$$\text{score}(w_i, w_j) = \frac{\overbrace{\text{count}(w_i, w_j)}^{\text{bigram}} - \delta}{\underbrace{\text{count}(w_i)}_{\text{unigram } w_i} \times \underbrace{\text{count}(w_j)}_{\text{unigram } w_j}}$$

P21

The probabilistic model:

$$P(\{s_t, y_t\}) = P(s_1)\, P(y_1|s_1) \underbrace{\prod_{t=2}^{T} P(s_t|s_{t-1})\, P(y_t|s_t)}_{\text{Separable. Conditional Independence.}}$$

y_t → observed, D

s_t → Hidden state, K

Let, K states.

$P(s_t|s_{t-1}) \Rightarrow K \times K$ matrix (state transition matrix)

$P(y_t|s_t) \Rightarrow K \times D$ observation matrix

→ modeled by GMM/Neural Network.

what if: $s_t = s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(m)}$ } factorial HMM !!

$K^{(m)}$ possible values. ← each

$K^{(m)} = K$ // simplicity

Then, K^M states $\boxed{K^M \times K^M}$ state transition matrix!!

→ impossible to work with

→ Requires constraint on state tx. mat.

factorial HMM → underlying state tx is constraind

$$P(s_t|s_{t-1}) = \prod_{m=1}^{M} \underbrace{P(s_t^{(m)} | s_{t-1}^{(m)})}_{\text{Decoupled.}}$$

$\boxed{s_{t-1}^{m} \rightarrow s_t^{m} \text{ only consideration}}$

p 21

observation:

$D \times D \Rightarrow$ covariance matrix

$$P(Y_t | s_t) = |C|^{1/2} (2\pi)^{-D/2} \exp\left\{ -\frac{1}{2} (Y_t - \mu_t)' C^{-1} (Y_t - \mu_t) \right\}$$

$$\mu_t = \sum_{m=1}^{M} W^{(m)} s_t^{(m)}$$ // for all mth contribution]

$W^{(m)}$: $D \times K$ [columns are probabilistic contribution of each states] $\Rightarrow D \times 1$ finally.

state variable $s_t^{(m)} \Rightarrow K \times 1$ vectors. → only one 1 (one hot encoding)

$\Rightarrow$ Depends on $s_t^{(m)}$ value

Learning & Inference:

Expectation Maximization: Param learning

$$Q(\phi^{new} | \phi) = E\left\{ \log P(\{s_t, Y_t\} | \phi^{new}) \,\middle|\, \phi, \{Y_t\} \right\} \quad (5)$$

ϕ^{new}: new params. ϕ: current params

$$P^{(m)} = P(s_t^{(m)} | s_{t-1}^{(m)})$$

Factorial HMM : $\phi = \{ W^{(m)}, \pi^{(m)}, P^{(m)}, C \}$ // find all of these ?? parameters.

$\pi^{(m)} = P(s_1^{(m)})$

E step:

Compute Q : Expand s by using earlier equations.

→ can be expressed as Expectation of

$$E\{\cdot | \phi, Y_t\} \Rightarrow \underbrace{\langle s_t^{(m)} \rangle}_{\text{state occupation}} \; ; \; \langle s_t^{(m)} \cdot s_t^{(n)} \rangle \; ; \; \underbrace{\langle s_{t-1}^{(m)}, s_t^{(m)} \rangle}_{\text{state transition}}$$

state occupation γ_t — $K \times 1$ vec.

?? two states jointly

state transition Σ_t — $K \times K$ mat

M step: maximize Q using Jensen's Inequality.

solved by: weighted linear regression.

P21

Gibbs Sampling : Inference :

$$s_t^{(m)} \sim P\Big(s_t^{(m)} \Big| \underbrace{\{s_t^n : n \neq m\}}_{\text{Neigh}}, \underbrace{s_{t-1}^{(m)}}_{\text{past}}, \underbrace{s_{t+1}^{(m)}}_{\text{future}}, \underbrace{y_t}_{\text{observation}}\Big)$$

$$\propto \underbrace{\underbrace{P\left(s_t^{(m)} \mid s_{t-1}^{(m)}\right)}_{\text{state transition}} \underbrace{P\left(s_{t+1}^{(m)} \mid s_t^{(m)}\right)}_{\text{state transition}}}_{\text{markovian}} \underbrace{P\left(y_t \mid s_t^{(1)}, \ldots s_t^{(m)}, \ldots s_t^{(M)}\right)}_{\boxed{\text{graphical model design itself}}}$$

Completely factorized Variational Inference:

$$\underbrace{\log P(\{y_t\})}_{\text{I}} = \log \sum_{\{s_t\}} P(\{s_t, y_t\})$$

$$= \log \sum_{\{s_t\}} Q(\{s_t\}) \frac{P(\{s_t, y_t\})}{Q(\{s_t\})}$$

$$\underbrace{\geq \sum_{\{s_t\}} Q(\{s_t\}) \log \left[\frac{P(\{s_t, y_t\})}{Q(\{s_t\})}\right]}_{\text{II}}$$

The difference between II & I is $|\text{I} - \text{II}|$ //simple math.

$$KL(Q \| P) = \sum_{\{s_t\}} Q(\{s_t\}) \log \left[\frac{Q(\{s_t\})}{P(\{s_t\} \mid \{y_t\})}\right]$$

→ change parameter of $Q(\{s_t\})$ to minimize.

$P \geq 1$

(iv)

vector itself.

$$Q(\{S_t | \theta\}) = \underbrace{\prod_{t=1}^{T}}_{\substack{\text{time}\\ \text{step}}} \underbrace{\prod_{m=1}^{M}}_{\substack{\text{possible}\\ \text{steps at each } t.}} Q(\{S_t^{(m)} | \theta_t^{(m)}\})$$

$$\theta_t^{m} = \begin{bmatrix} \theta_{t,1}^{(m)} \\ \theta_{t,2}^{(m)} \\ \vdots \end{bmatrix}$$

vector element

m-th markovian chain state k, at time t

$$Q(S_t^{(m)} | \theta_t^{(m)}) = \underbrace{\prod_{k=1}^{K} \left(\theta_{t,k}^{(m)} \right)^{S_{t,k}^{(m)}}}_{\substack{\text{multiply.}\\ \text{All the params.}}} ; \quad S_{t,k}^{(m)} \in \{0, 1\} \quad S_t^{(m)} = \begin{bmatrix} S_{t,1}^{(m)} \\ S_{t,2}^{(m)} \\ \vdots \end{bmatrix}$$

with $\underbrace{\sum_{k=1}^{K} S_{t,k}^{(m)} = 1}_{\text{only one is } 1 \text{ else } 0}$

$\theta_t^{(m)} \to$ state occupation prob. with multinomial var. $S_t^{(m)}$ under distribution Q

vector softmax element wise.

vector of diagonal elements $W^{(m)'} C^{-1} W^{(m)}$

$$\theta_t^{(m)\,\text{New}} = \phi \left\{ W^{(m)'} C^{-1} \underbrace{\tilde{Y}_t^{(m)}}_{\text{residual error}} - \frac{1}{2} \Delta^{(m)} + (\log P^{(m)})\, \theta_{t-1}^{(m)} + (\log P^{(m)})' \theta_{t+1}^{(m)} \right\}$$

$$\tilde{Y}_t^{(m)} = Y_t - \sum_{l \neq m}^{M} W^{(l)} \theta_t^{(l)}$$

Structured Variational Inference:

$$Q(\{S_t\} | \theta) = \frac{1}{Z_Q} \prod_{m=1}^{M} Q(S_1^{(m)} | \theta) \prod_{t=1}^{T} Q(S_t^{(m)} | S_{t-1}^{(m)}, \theta)$$

$\hookrightarrow$ normalized

$$Q(S_1^{(m)} | \theta) = \prod_{k=1}^{K} \left(h_{1,k}^{(m)} \pi_k^{m} \right)^{S_{1,k}^{(m)}}$$

$$Q(S_t^{(m)} | S_{t-1}^{(m)}, \theta) = \prod_{k=1}^{K} \left(h_{t,k}^{(m)} \sum_{j=1}^{K} P_{k,j}^{(m)} S_{t-1,j}^{(m)} \right)^{S_{t,k}^{(m)}}$$

P21

$$Q\left(S_t^{(m)} \mid S_{t-1}^{(m)}, \theta\right) = \prod_{k=1}^{K} \left(h_{t,k}^{(m)} \prod_{j=1}^{K} \left(P_{kj}^{(m)} \right)^{S_{t-1,j}^{(m)}} \right)^{S_{t,k}^{(m)}}$$

one hot vector.

$$\theta = \{ \pi^{(m)}, P^{(m)}, \underbrace{h_t^{(m)}}_{} \}$$

$K \times 1 \rightarrow$ prob of observation $P(Y_t \mid S_t)$

for each k setting $S_t^{(m)}$

$$Q\left(S_{1,j}^{(m)} = 1 \mid \theta\right) = \underbrace{h_{1,j}^{(m)}}_{} \, P\left(\underbrace{S_{1,j}^{(m)} = 1}_{} \mid \phi\right)$$

$\Rightarrow$ having an observation at $t = 1$, under $S_{1,j}^{(m)} = 1$

has prob of $h_{1,j}^{(m)}$

Can be proved that, $KL(Q \| P)$ is minimized.

$$h_t^{(m)\,new} = \exp\left\{ W^{(m)'} C^{-1} \tilde{Y}_t^{(m)} - \frac{1}{2} \Delta^{(m)} \right\}$$

res $\rightarrow = Y_t - \sum_{l \neq m}^{M} W^{(l)} \langle S_t^{(l)} \rangle$

connected to ELBO bound

$$F(Q, \phi) = \underbrace{E_Q\left\{ \log P(Y, S \mid \phi) \right\} - E_Q \log \{Q(S)\}}_{} \leq \log P(Y_S)$$

(−) value. → reverse it (interesting)

No more info than |ELBO|

P22

Problem formulation:

$\{A_1, A_2 \ldots A_m\}$ → m smart phone.

$A_i = \{A_{i1}, \ldots A_{iL}\}$ // complete trace for A.

$A_{ij} = \{t_{ij}, x_{ij}, y_{ij}\}$ // temporal / spatial information.

Query $Q = \{Q_1 \ldots Q_f\}$; $f \ll L$ time.

Targets k relevant trajectories of Q from A site

· Trajectory comparison function $LCSS(Q, A_i)$ → compare their trajectory.

Longest Common Subsequence (LCSS)

By definition:

$$LCSS_{\delta,\epsilon}(A, B) = \begin{cases} 0, & A \text{ or } B = \emptyset \\ 1 + LCSS_{\delta,\epsilon}(Head(A), Head(B)) & \text{if: } |a_{x:L_1} - b_{x:L_2}| < \epsilon \\ & |a_{y:L_1} - b_{y:L_2}| < \epsilon \\ & |L_1 - L_2| < \delta \\ \max\left(LCSS_{\delta\epsilon}(Head(A), B), LCSS_{\delta\epsilon}(A, Head(B))\right) & ; \text{otherwise} \end{cases}$$

L_2 → time, x → x coordinate; L_2 → time, y → y coordinate

δ → time matching window; ϵ → spatial matching window

Both are application Specific.

[iterative Algorithm]

$$Head(A) = (\underbrace{(a_{x:1}, a_{y:1})}_{(x,y \text{ at time } 1)}, \ldots, (a_{x:L-1}, a_{y:L-1}))$$

Pb2

Bounding Above Loss : Easier Computation.

$$Loss(MBE_Q, A_i) = \sum_{j=1}^{|A_i|} \begin{cases} 1, & \text{if } A_i[j] \text{ within envelop.} \\ 0; & \text{otherwise.} \end{cases}$$

MBE_Q : Minimum Bounding Envelop of Query Q

MBE_Q is the area between high envelop & low envelop. $EnvHigh[i]$ $Envlow[i]$

$$EnvHigh[i] = max(Q[j] + \epsilon); \quad |i-j| \leq \delta$$

$$Envlow[i] = min(Q[j] - \epsilon); \quad |i-j| \leq \delta$$

#P 29 (1)

unique Solution

{ G → recovers the training Data Distribution
  D → $\frac{1}{2}$ everywhere

Adversarial Nets:

gen ~ P_g

Noise vector $P_z(z)$ → map $G(z, \theta_g)$

⇓

Differentiable function (MLP) multi layer perceptrons

$D(x, \theta_d)$ → Differentiable MLP.

↳ Discriminator → x from data / G ??

$$\boxed{\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x \sim p_{data}(x)} \left\{ \log [D(x)] \right\} + \mathbb{E}_{z \sim p_z(z)} \left\{ \log [1 - D(G(z))] \right\}}$$

⇒ iterative Numerical approach:

{ k step of D → to keep near optimal (inner loop)
  1 step of G → changes slowly enough.
[SML/PCD way??]

Theory: Algorithm 1 ⇒ crack of jack.

understanding Sequence: when D is optimal ⇒ ?? (keep it)? near

How/When G is optimal | D is optimal

what happens when D is optimal ??

what happens when both are optimal

P 29 ⑪

for no of train

for _ in range (k)

sample $z^1, \dots z^{(m)} \to P_g(z)$

sample $x^1 \dots x^m \to P_{data}(x)$

update θ_d, D by ascending ~~grad~~ stochastic grad.

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}) + \log\left(1 - D(G(z^{(i)}))\right)\right] \quad // \text{maximize}$$

→ end_for

inner loop

Keep k near optimal

sample m noise $\{z^{(1)} \dots z^{(m)}\}$

Update the gen. θ_g descending gradient

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D(G(z^{(i)}))\right) \quad // \text{ minimize.}$$

end_for.

global optimality: $P_g = P_{data}$.

Prop 1: For G Fixed,

$$\boxed{D^*_G(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}}$$

// optimal Discriminator (maximize)

training criterion:

$$V(G,D) = \int_x P_{data}(x) \log(D(x))\,dx + \int_z P_z(z) \log(1 - D(G(z)))\,dz$$

$$= \int_x P_{data}(x) \log(D(x))\,dx + \int_x P_g(x) \log(1 - D(x))\,dx$$

for any function of $y \to a \log y + b \log(1-y)$ achieves its maximum in $[0,1]$ at $\left(\frac{a}{a+b} = y\right)$

129

$$\underline{C(G)} = \max_D V(G, D)$$

$$= \mathbb{E}_{x \sim p_{data}} \left[\log D^*_G(x) \right] + \mathbb{E}_{z \sim p_z} \left[\log \left(1 - D^*_G(G(z)) \right) \right]$$

$$= \mathbb{E}_{x \sim p_{data}} \left[\log D^*_G(x) \right] + \mathbb{E}_{x \sim p_g} \left[\log \left(1 - D^*_G(x) \right) \right]$$ // from earlier argument ($y = \frac{a}{a+b}$) maximize

$$= \mathbb{E}_{x \sim p_{data}} \left[\log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + \mathbb{E}_{x \sim p_g} \left[\log \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right]$$

Now D is set → [tune $p_g(x)$ / $G(x)$, to minimize this] — 1

The global minima for $C(G)$ is iff $\boxed{p_g(x) = p_{data}(x)}$ // Now D fails to comprehend

in that case,

$$\mathbb{E}_{x \sim p_{data}} \left[\log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + \mathbb{E}_{x \sim p_g(x)} \left[\log \left[\frac{p_g(x)}{p_{data}(x) + p_g(x)} \right] \right] =$$

$$= -\log 2 - \log 2 = -2 \log 2$$

Because G is too good

Not because D is bad

D is still OPTIMAL

Re organizing the equation 1 we get

[p_g may not be optimal]

$$C(G) = -\log 4 + \mathbb{E}_{x \sim p_{data}} \left[\log \frac{p_{data}(x)}{\left(\frac{p_{data}(x) + p_g(x)}{2} \right)} \right] + \mathbb{E}_{x \sim p_g} \left[\log \frac{p_g(x)}{\left(\frac{p_{data}(x) + p_g(x)}{2} \right)} \right]$$

$$= -\log 4 + \underbrace{KL\left(p_{data} \,\Big\|\, \frac{p_{data} + p_g}{2} \right) + KL\left(p_g \,\Big\|\, \frac{p_{data} + p_g}{2} \right)}$$

$$= -\log 4 + 2\, JSD\left(p_{data} \,\|\, p_g \right)$$ // jenson shanon divergence

For optimal G^* ∃ $C(G) = -\log 4$!! // $\boxed{\text{if } p_g \text{ is optimal}}$ → if only ($p_g = p_{data}$)

[minimum for optimal D]

P 29

IV

Convergence of Algo: 1 1st discriminator reaches optimal for G $\boxed{D^*(x) = P_{data} / (P_g + P_{data})}$ 2

and P_g is updated to improve. $\boxed{\text{then } G \to (P_g) \to (P_{data})}$

1

$$\mathbb{E}_{x \sim P_{data}} \left[\log D^*_G(x) \right] + \mathbb{E}_{x \sim P_g} \left[\log\left(1 - D^*_G(x)\right)\right]$$

II Then P_g converge to P_{data}.

⇓ optimize this instead of P_g

MLP via function $G(z ; \theta_g)$

(multiple critical points → optimal multiple local)

theory → (not yet)

Dis Ad: G must not be trained too much !! (D must be sync with G)

mode collapse !! convergence ??

Ad: only backpropagation

Large distribution learning.

→ if D is too strong G learns nothing??

(i) Paper 30

Joint dist $p(\underset{\text{obs}}{x}, \overset{\text{hidden}}{h} | \underset{\text{param}}{\eta}) = p(x|h)\, p(h|\eta)$

$$p(h|x,\eta) \propto p(h, x|\eta)$$

predictive, $p(x_{new}|x) = \int p(x_{new}|h)\, p(h|x,\eta)\, dh$.

$$p(\underset{\text{mean}}{\mu_{1:k}}, \overset{\text{mixture (dir)}}{\theta}, \underset{\text{class}}{z_{1,N}} | \underset{\text{observ}}{x_{1:N}})$$

Generative probabi mod:

global hid var

mixture prop $\theta \sim \text{Dirichlet}(\alpha)$

$\mu_k \sim N(0, \sigma_0^2)$

(4) mixture assignment $z_n | \theta \sim \text{Discrete}(\theta)$?

(5) Data point $x_n | z_n, \mu \sim N(\mu_{z_n}, 1)$.

easy pesy:- sample $\theta \rightarrow$ corresponding $\mu_{1-k} \rightarrow$ for each data point estimate $z_n \in {1 \ldots k} \rightarrow$ govern the distribution.

Paper 30

joint distribution:

$$p(\theta, \mu, z, x \mid \sigma_0^2, \alpha) = p(\theta \mid \alpha) \prod_{k=1}^{k} p(\mu_k \mid \sigma_0^2) \prod_{i=1}^{N} p(z_i \mid \theta) p(x_i \mid z_i, \mu)$$

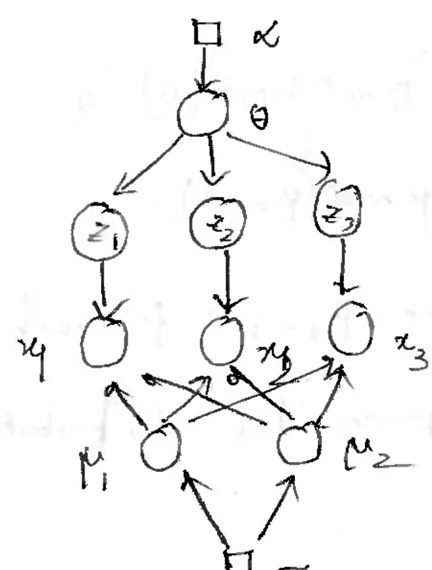
Posterior

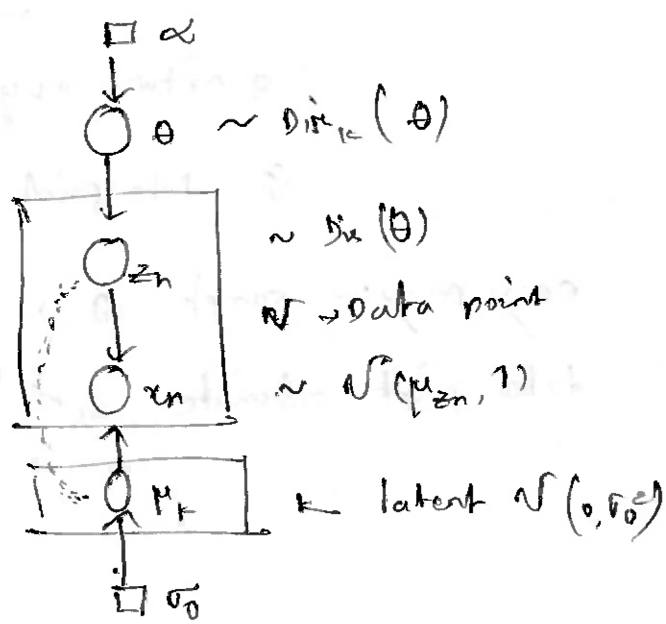
$$p(\theta, \mu, z \mid x, \sigma_0^2, \alpha) = \frac{p(\theta, \mu, z, x \mid \sigma_0^2, \alpha)}{\iint p(x \mid \sigma_0^2, \alpha)}$$

Predictive Distribution.

$$p(x_{new} \mid x, \sigma_0^2, \alpha) = \int \left(\sum_{z_{new}} p(z_{new} \mid \theta) \, p(x_{new} \mid z_{new}, \mu, \sigma_0^2) \, p(\theta, \mu \mid x, \sigma_0^2, \alpha) \right) d\theta \, d\mu$$

The graphical model:





P 30

Example model:

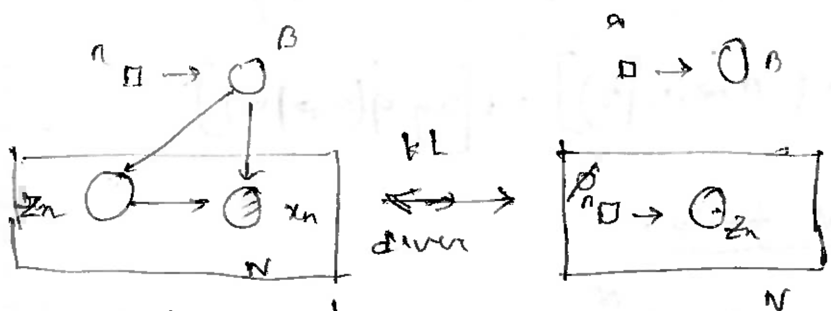
(i) linear factor model: PCA, factor model. (graph 3)

(ii) mixed membership model:

(iii) Matrix factorization model:

(iv) Time Series
  - ~~Time Series Models~~ Hidden Markov model
  - Kalman filter

Posterior Inference with mean field: Variational method.



Latent var. mod

Variational family.

Conditional Conjugate model:

local latent

Always fixed mixture prop

$$p(\beta, z, x \mid \lambda) = p(\beta \mid \lambda) \prod_{n=1}^{N} p(z_n \mid \beta)\, p(x_n \mid z_n, \beta)$$

global latent

obs

changes so local

Posteriore,

$$p(\beta, z \mid x) = \frac{p(\beta, z, x)}{\int p(\beta, z, x)\, d\beta, dz} \rightarrow \text{how? Problem!}$$