P7

score metric for (x, y) word pairs

$$\underline{\underline{S_{(a,b)}}}(x,y) = \begin{cases} \cos(\vec{a}-\vec{b}, \vec{x}-\vec{y}) & \text{if } \|\vec{x}-\vec{y}\| \le \delta \\ 0 \end{cases}$$

baseline

seed direction (she, he)

word $\vec{w} \in \mathbb{R}^d$

$\|\vec{w}\| = 1$

similarity th

1th experiment

Analogy

Direct gender bias

Definition

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

gender neutral word

set

strictness

?? confused ~~here.~~

projection

[gender subspace top principle component]

indirect Bias Definition

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

Described in next page

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

‖ 0 means no projection.

⇓

$\beta(w, v) = 0$

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

then $\Rightarrow \beta(w, v) = 1$ vv

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 $D_i \Rightarrow \mu_i := \sum_{w \in D_i} \vec{w} / |D_i|$.

$\{\vec{w} \in \mathbb{R}^d\}$ word vector

$k \geq 1$

Let Bias subspace

$$C := \sum_{i=1}^{n} \sum_{w \in D_i} (\vec{w} - \mu_i)^T (\vec{w} - \mu_i) / |D_i|$$

[k row of SVD is Bias subspace B]

P7

new embedding — orthogonal projection

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

→ got B matrix / projection earlier ✓ now ?? → step

∴ word to Neutralize $N \subseteq W$

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

$E_i \subseteq W$ → finally all words will have similar (component in gender neutral Direction)

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

$v := \mu - \mu_B$ // orthogonal — projection to B

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

this term varies only for words in E set

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

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

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 Neutral embedding words]

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

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

overall match.

P2

simplified version of pagerank:

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

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

$v \in B_u$: Back link of u (towards u)

$c \rightarrow$ Normalization.

$\rightarrow$ Recursive equation.

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

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

0 if no edge between u, v

$$R = c\,A\,R$$

c: eigen value

R $\rightarrow$ A vector! (bad notation)

R $\rightarrow$ eigen vector of A

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

$\downarrow$ All eivectors are orthogonal.

$\rightarrow$ Dominant eigenvector

$\rightarrow$ power iteration.

Ranksource modification.

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

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

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

Decay factor.

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

$\boxed{1}$: All 1 matrix

P9

$R_0 \leftarrow S$ // Random ints.

loop:

(Dominant) finding Eig vector for A

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

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

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

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

increases convergence.

maintain $\|R\|_1$

// Normalize.

while $\delta > e$ //convergence

P12: Recommendation problem formulation.

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

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

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

$u: C \times S \rightarrow R$ (rating value → 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.

↳ similar to previous s_i will be recommended

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

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

$n_i \rightarrow$ 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 to.

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

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(contentbased 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 ?

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

more like collaboration term.

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

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

(Average rating)

P 12

Pearson coefficient based similarity

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

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

interaction between user u & item i

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

How to get it?

SVD? but empty elements??

imputation → bad idea.

So, optimization problem:

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

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 $\boxed{b_{ui} = \mu + b_i + b_u}$

$\mu \to$ overall rating

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

Now the optimization problem changes to,

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

// 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 $\Rightarrow |N(u)|^{-0.5} \sum_{i \in N(u)} x_i^{4.5}$ // empirical.

user Attributes $A(u)$ set $\Rightarrow \boxed{y_a \in \mathbb{R}^f}$ elements. Associated factor 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 $\hat{r}_{ui}(t) = \underbrace{\mu}_{\text{static}} + \underbrace{b_i(t)}_{\text{item bias}} + \underbrace{b_u(t)}_{\text{user bias}} + \underbrace{q_i^T}_{\text{static}} \underbrace{p_u(t)}_{\text{human behavior dynamics.}}$

Dynamic

input with confidence level:

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

(Variational Lower bound)

$z \to x$ $P(x) \to$ Prob dist. over variable

Latent   observed. $p(x) \to$ Pdf of Distribution x

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

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

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

Reorganizing

$L = \underset{q}{E}[\log P(x|z)] - KL[q(z)\,\|\,p(z)]$

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

ELBO, $L = \underbrace{\underset{q}{E}[\log P(x,z)]}_{|\text{Negative}|} + \underbrace{H_q(z)}_{|+|}$ // entropy def.

$|\text{Negative}| > |+|$ // so overall (-)

$\boxed{\text{Always else } P(x) = 1}$: no info.

Interprete: (-) of information $\geq$ ELBO // reverse it - (interesting)

No more info than $-|ELBO|$ in $P(x)$

$\Rightarrow$ more info than $|ELBO|$

Derivation 2: $KL(q(z)\,\|\,P(z|x)) = \int q(z) \log \dfrac{q(z)}{P(z|x)}\,dz$

(Backward KL) ??

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

fixed

$\therefore \boxed{\log P(x)} = L + \boxed{KL(q(z)\,\|\,P(z|x))} > 0$

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

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 (event count) distribution (topic to word) (T of them)

(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)

$$\begin{cases} w_i \mid z_i, \phi^{z_i} \sim \text{Discrete}(\phi^{z_i}) & \text{from earlier} \\ \phi \sim \text{Dir}(\beta) & \text{// new [conjugate prior]} \\ z_i \mid \theta^{d_i} \sim \text{Discrete}(\theta^{d_i}) & \text{Earlier} \\ \theta \sim \text{Dir}(\alpha) & \text{// new [conjugate prior]} \end{cases}$$

α, β hyperparameter.

P15

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

No document here

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

where,

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

no. time word w ← j topic (Assign)

//see wiki multinomial Approx. → (2)

j topics

Gamma function.

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

similarly

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

distict from early. → (3)

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

// Again Intractable T^n terms
so require Approximation

Solve it By MCMC. (Gibbs Sampling).
require $P(z_i \mid z_{-i}, w)$

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

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

(prob of w_i under topic j)

→ prob of topic j in document d_i

//just need counter

$n_{-i}^{(\cdot)}$ → 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)

(Fixed it in experiment)

T → Topic Number ??

↓

This is model selection?

Target: $P(w \mid T)$

T → topic Number.

w → All words

Approximated by

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

P16 ①

Expected value. $\tilde{p}(f) = \sum_{x,y} \tilde{p}(x,y) f(x,y)$ empirical — $\frac{1}{N}$ × No of time (x,y) appears. (training data) — training data — calculate from data.

$\sum_{x,y} \tilde{p}(x) p(y|x) f(x,y) = p(f)$ (model) — empirical of x (training data) — given — indicator function — feature function

$P(x) \sim x_i$

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

Explicitly, $\sum_{x,y} \underbrace{\tilde{p}(x) p(y|x) f(x,y)}_{\text{model}} = \sum_{x,y} \underbrace{\tilde{p}(x,y) f(x,y)}_{\text{traing data.}}$ // constraint equation

Key goal

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

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

All the 2 possible Data prob

Conditional Entropy, $H(p) = - \sum_{x,y} \tilde{p}(x) \underbrace{p(y|x)}_{\text{model}} \log p(y|x) \geq 0$ — lower bound — sure model

(set of probabilities distribution) $H(p) \leq$ uniform case entropy. $= \log |y|$ = cardinality of y — upper bound.

Maximum Entropy: $P_* \in C$ satisfy $\tilde{p}(f) = p(f)$ — train — model

$P_* = \arg\max_{p \in C} H(p)$ // uniform case

⇓

well defined & unique.

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

Parametric Form:

find $P_* = \arg\max_{p \in C} H(p)$ // primal optimization.

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

Now, $P_\lambda = \arg\max_{p \in P} \Lambda(p, \lambda)$

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

II

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^*

$Z_\lambda(x)$ 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: $f_1 \dots f_n \rightarrow$ empirical $\tilde{p}(x,y)$

output λ_i^*, p_λ

1. $\lambda_i = 0 \; (i = 1 \dots n)$
2. $i \in \{1 \dots n\}$

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

Vapnik 1995

$$R(\alpha) \le R_{emp}(\alpha) + \underbrace{\overbrace{\sqrt{\left(\frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l}\right)}}^{\text{VC confidence.}}}$$

$l\downarrow\ h\uparrow$ $\eta\downarrow$

Here, $0 \le \eta \le 1$; 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~~ VC confidence lower → the better] Approximation (may overfit)

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

~~VC~~ 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

p.19 satisfying hyperplane (HP): $\underline{w}\cdot x + b = 0$ & Linear

projection to HP $= (-b)$ bias (All points projected as $-b$ amount) (Nice)

$\underline{w}$ ⇓ normal to HP.

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

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

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

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

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.

putting values →

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

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

kernel idea.

KKT condition:

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

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

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

$$\alpha_i \geq 0 \quad \forall i$$

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

(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 Condition:- $(\mathcal{H}, \phi)$ $(d \to \mathcal{H})$

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

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

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

then

$\int K(x, y)\, g(x)\, g(y)\, dx\, dy \geq 0$ ⟹ 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 $\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0} \log p(w_{t+j}|w_t)$

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

So, maximizing $p(w_{t+j}|w_t)$ ✓

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

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.

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

1 if true, else 0

[computation ∝ L(w)]

⇒ care about $\boxed{\text{input/output}}$ representation.

Negative Sampling

should be high (interesting)

NEG objective ⇒

$$\underbrace{\log \sigma(v'^{T}_{w_O} v_{w_I})}_{} + \sum_{i=1}^{K} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-v'^{T}_{w_i} v_{w_I})\right]$$

⇓ row of matrix — positive

$P_n(w)$ ↦ different choices

negative word how (neg)

⇓ should be low & negative

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

// modified NCE

Subsampling of freq words.

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

// Discarding prob.

✓ $f(w_i)\uparrow$ $P(w_i)\uparrow$

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

(High freq words discarded more)

Balance between rare & frequent words.

Bigram skip:

(Phrase selection)

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

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

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

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

P21

observation:

$D \times D \Rightarrow$ covariance matrix

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

$$\mu_t = \sum_{m=1}^{M} W^{(m)} S_t^{(m)}$$ // for all m-th contribution] ✓

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

probabilistic

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

⇒ Depends on $S_t^{(m)}$ value ✓

Learning & Inference:

Expectation Maximization: Param learning

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

ϕ^{new} → new params.

ϕ → current params

$P^{(m)} = P(S_t^{(m)} \mid S_{t-1}^{(m)})$

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

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

state occupation γ_t $\underbrace{K \times 1}$ vec.

?? (two states jointly)

state transition Σ_t $\underbrace{K \times K}$ mat

M step : maximize Q using $\underbrace{\text{Jensen's Inequality}}$.

solved by : weighted linear regression.

P21

Gibbs Sampling: Inference:

$$S_t^{(m)} \sim P\Big(S_t^{(m)} \Big| \underbrace{\{S_t^{n} : n \neq m\}}_{\text{Neigh}}, \underbrace{S_{t-1}^{(m)}}_{\text{past}}, \underbrace{S_{t+1}^{(m)}}_{\text{future}}, \overbrace{Y_t}^{\text{observation}}\Big)$$

$$\propto \underbrace{\underbrace{P(S_t^{(m)} \mid S_{t-1}^{(m)})}_{\text{state transition}} \underbrace{P(S_{t+1}^{(m)} \mid S_t^{(m)})}_{\text{state transition}}}_{\text{markovian}} \underbrace{P(Y_t \mid S_t^{(1)}, \dots S_t^{(m)}, \dots S_t^{(M)})}_{\boxed{\text{graphical model design itself}}}$$

completely factorized Varicational 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 $|(i) - (ii)|$ //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)

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

vector itself.

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

vector element

mth markovian chain state k, at time t

$$Q(S_t^{(m)} | \theta_t^{(m)}) = \underbrace{\prod_{k=1}^{K} \left(\theta_{t,k}^{(m)}\right)^{S_{t,k}^{(m)}}}_{} \; ; \quad S_{t,k}^{(m)} \in \{0,1\}$$

multiply. All the params.

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

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

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

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

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

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

Structured Variational Inference:

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

$\to$ normalized

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

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

(v)

P21

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

one hot vector.

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

$k \times 1 \to$ prob of observation $P(Y_t \mid S_t)$

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

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

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

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

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

connected to ELBO bound

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

(−) value. → reverse it (interesting)

No more info than |ELBO|

P22

Problem formulation:

$\{A_1, A_2 \cdots A_m\}$ — m smart phone.

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

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

Query $Q = \{Q_1 \cdots Q_f\}$; $f \ll L$ 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:

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

δ: time matching window; ϵ: spatial matching window

Both are application specific.

L_1, L_2 → time; x → x coordinate; y → y coordinate

[iterative Algorithm]

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

Pb2

Bounding Above Loss: Easier Computation.

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

MBE_Q : Minimum Bounding Envelop of Query Q

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

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

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

#P 29

1

unique Solution

{ G → recovers the training Data Distribution
D → 1/2 everywhere

Adversarial Nets:

gen ~ p_g

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

⇓

multi layer perceptrons

Differentiable function (MLP)

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

↳ Discreminator → x from data / G ??

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} \{\log[D(x)]\} + \mathbb{E}_{z \sim p_z(z)} \{\log[1 - D(G(z))]\}$$

⇒ Iterative Numerical approach:

{ k step of D → to keep near optimal (inner loop)
1 step of G → changes slowly enough.

[ML/PCD way??]

Theory: Algorithm 1 → crack of jack.

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

How / When G is optimal | D is optimal ✓

what happens when D is optimal ??. ✓

what happens when both are optimal]

for no of train

for _ k 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.

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

inner loop keep k near optimal

end for

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

Update the gen. θ_g descending gradient

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

$$C(G) = \max_D V(G, D)$$

$$= \mathbb{E}_{x\sim P_{data}}\left[\log D_G^*(x)\right] + \mathbb{E}_{z\sim P_z}\left[\log\left(1 - D_G^*(G(z))\right)\right]$$

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

$$= \mathbb{E}_{x\sim P_{data}}\left[\log \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}\right] + \mathbb{E}_{x\sim P_g}\left[\log \frac{P_g(x)}{P_{data}(x) + P_g(x)}\right]$$ — (1)

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

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]

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

$$= -\log 4 + \underbrace{KL\left(P_{data} \,\middle\|\, \frac{P_{data} + P_g}{2}\right) + KL\left(P_g \,\middle\|\, \frac{P_{data} + P_g}{2}\right)}$$

$$= -\log 4 + 2\, JSD\left(P_{data} \,\|\, P_g\right)$$ // Jenson Shanon divergence

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

[minimum for optimal D] ($P_g = P_{data}$)

P 29

IV

Convergence of Algo: 1st discriminator reaches optimal for G

$$D^*(x) = P_{data} / (P_g + P_{data})$$

and P_g 's updated to improve.

then ($G \rightarrow (P_g) \rightarrow (P_{data})$)

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

then P_g converge to P_{data}.

⇓

MLP via function $G(z; \theta_g)$ — optimize this instead of P_g

(multiple critical point) → (multiple local optimal) theory → (not yet)

DisAd: 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 learn nothing??

Paper 30

Joint dist $p(x, h \mid \eta) = p(x \mid h)\, p(h \mid \eta)$

hidden

obs param

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

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

mixture (dir)

$$p(\mu_{1:k}, \theta, z_{1:N} \mid x_{1:N})$$

mean class observ

Generative probabi mod:

global hid var

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

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

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

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

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

Paper 30

joint distribution:

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

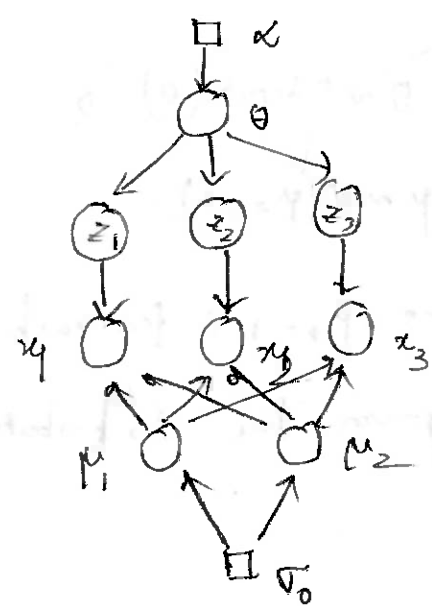
Posterior

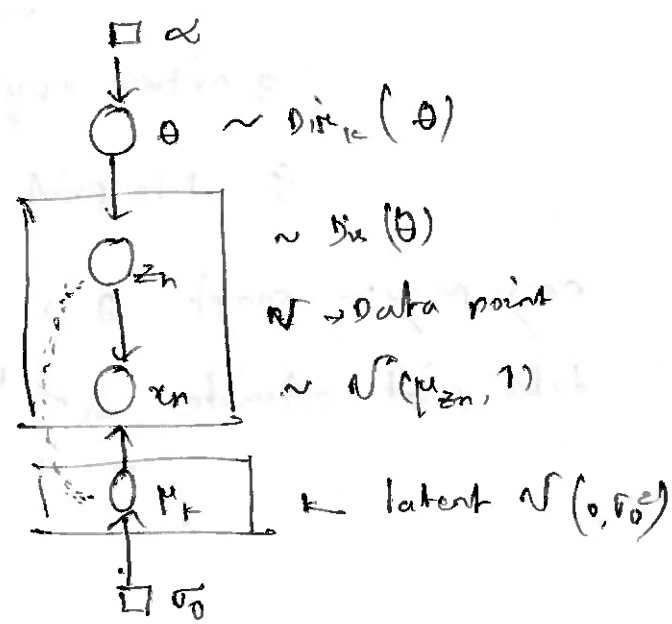
$$p(\theta, \mu, z \mid x, \sigma_0^2, \alpha) = \frac{p(\theta, \mu, z, x \mid \sigma_0^2, \alpha)}{\iint p(x \mid \sigma_0^2, \alpha)}$$

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)

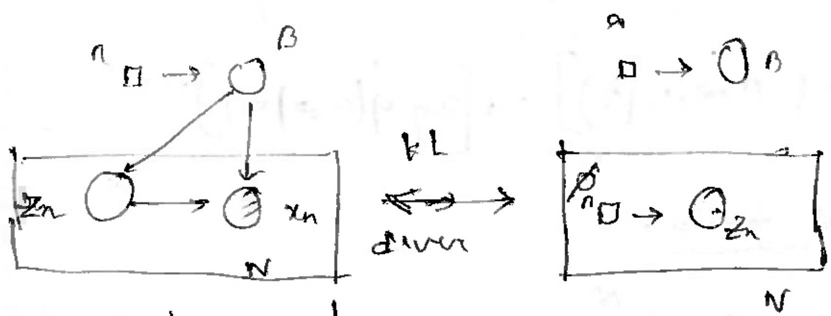
(ii) mixed membership model:

(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:

$$\underbrace{p(\beta, \overset{\substack{\text{local latent}\\ \uparrow}}{z}, \underset{\substack{\downarrow\\ \text{obs}}}{x} \mid \eta)}_{\text{global latent}} = p(\beta \mid \eta) \prod_{n=1}^{N} p(z_n \mid \beta)\, p(x_n \mid z_n, \beta)$$

Always fixed ↑ mixture prop

z_n ↓ changes so local

Posterior,

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

p 30

(iv)

$$p(x|\eta) = h(x)\exp\left(\eta^T t(x) - a(\eta)\right)$$ ↗ exponential family.

Dependent

natural param.

$h(x)$: base measure.

some suff. sufficient stat (great)

$a(\eta)$: log normalizer.

mean field variational model: Approximate Posterior:

variational Objective function

$$v^* = \arg\min_v KL\left(q(\beta, z|v) \,||\, p(\beta, z|x)\right)$$

(good read)

$$\mathcal{L}(v) = E\left[\log p(\beta, z, x|\eta)\right] - E\left[\log q(\beta, z|v)\right]$$

mean field variational family:

$$q(\beta, z|v) = q(\beta|\lambda)\prod_{n=1}^{N} q(z_n|\phi_n)$$

$v = \{\lambda, \phi_{1-N}\}$ } easy.

coordinate Ascent Varia. Inference:

$$\lambda^* = E_q\left[\eta_g(z, x)\right] \quad \text{global}$$

$$\phi_n^* = E_q\left[\eta_L(\beta, x_n)\right]$$

↳ local

$$q(\mu, z) = \prod_{i=1}^{k} q(\mu_k|\lambda_k)\prod_{n=1}^{N} q(z_n|\phi_n)$$

(v)

Model Criticism: Exploration & prediction.

⇓

inference about hidden vars.

distribution

$$P(x_{new}|x) = \int \underbrace{p(\beta|x)}_{\text{not available}} \left(\int p(z_{new}|\beta)\, p(x_{new}|z_{new},\beta)\, dz \right) d$$

Predictive Sample Reuse:

$p(\beta, z, x)$

$p(\beta, z \mid x_{[n]})$

$$l_n = \log p(x_n \mid x_{[n]}) \quad \text{n removed}$$

$$\left[= \log \int \left(\int p(x_n|z_n)\, q(z_n)\, dz \right) q_{[n]}(\beta)\, d\beta \right] q_{[n]}(\beta, z)$$

full likelihood $\sum_{n=1}^{N} l_n$

Posterior Predictive Check:

Test statistics

$$PPC = P(T(x^{rep}) > T(x) \mid x)$$

⇓

Data drawn from hypothetical feature. obs.

$$PPC = P(T(x^{rep}, \beta) > T(x, \beta) \mid x)$$

$$p(\beta, x^{rep} \mid x) = p(\beta|x)\, p(x^{rep}|\beta)$$

$$PPC = \int p(\beta|x) \int p(x^{rep}|\beta)\, \mathbb{1}[T(x^{rep},\beta) > T(x,\beta)]\, dx^{rep}\, d\beta$$

P 10

$T(x^+; \beta^+) > T(x; \beta^+)$

$T(x, \beta) = \frac{1}{N} \sum_{n=1}^{N} \log(x_n \mid \beta)$ // may be T)

PPC is adaptive.

Basics

Batch Normalization in CNN: input $[N, C, H, W] \Rightarrow [i, c, j, k]$

Batch size

Height

width

filter channel

for all items

same filter (c)

$$\Rightarrow \mu_{B,c} = \frac{1}{NHW} \sum_{i=1}^{N} \sum_{j=1}^{H} \sum_{k=1}^{W} x_{i,c,j,k}$$

For each channel we get 1 value
So total C value.

$$\Rightarrow \sigma^2_{B,c} = \frac{1}{NHW} \sum_{i=1}^{N} \sum_{j=1}^{H} \sum_{k=1}^{W} \frac{(x_{i,c,j,k} - \mu_{B,c})^2}{\cancel{N} \; 1}$$

$$\Rightarrow \text{final output } \hat{x}_{i,c,j,k} = \frac{x_{i,c,j,k} - \mu_{B,c}}{\sqrt{\sigma^2_{B,c} + \epsilon}}$$

$\Rightarrow$ Do calculation for each $c \in C$ channels.

More formally,

$$\hat{x}_{i,c,j,k} = \gamma \left(\frac{x_{i,c,j,k} - \mu_{B,c}}{\sqrt{\sigma^2_{B,c} + \epsilon}} \right) + \beta .$$

Solves: (i) Internal Covariate Shift. (each ~~feature~~ zero mean, var = 1)
→ features distribution differs internally.
→ inside the neural Network (layer - layer)

(ii) Robust Network creation → less prone to perturbation

(iii) Learning faster.

Basic

Instance Normalization (IN) / Layer Normalization

$$IN(x_{i,c,j,k}) = \gamma \left(\frac{x_{i,c,j,k} - \mu_{i,c}}{\sqrt{\sigma_{i,c}^2 + \epsilon}} \right) + \beta$$

γ → can be conditioned → conditional IN

where,

$$\mu_{i,c} = \frac{1}{HW} \sum_{j=1}^{H} \sum_{k=1}^{W} x_{i,c,j,k}$$

j, k → summed out.

$$\sigma_{i,c}^2 = \frac{1}{HW} \sum_{j=1}^{H} \sum_{k=1}^{W} (x_{i,c,j,k} - \mu_{i,c})^2$$

$\boxed{\text{instance : for each } n \text{ \& } c}$

Adaptive Instance Normalization (AdaIN)

$$AdaIN(x_{i,c}, y) = \sigma(y) \left(\frac{x_{i,c} - \mu_{i,c}}{\sqrt{\sigma_{i,c}^2 + \epsilon}} \right) + \mu(y)$$

var → $\sigma(y)$

Adaptive term.

$\mu(y)$ ← y mean

$AdaIN(x_{i,c}, y)$ → contional

$\sigma(y)$:
→ 0 if y is constant
→ High if y varies a lot.

$\mu(y)$:
→ retain if y is constant
→ should be 0 mean.

Alternative

Layer Normalization.

$$LN(x_{i,c,j,k}) = \gamma \frac{x_{i,c,j,k} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

$$\mu_i = \frac{1}{H} \cancel{\Sigma} \frac{1}{CW} \sum_{c=1} \sum_{j=1}^{H} \sum_{k=1}^{W} x_{i,c,j,k}$$

c → selecting subset leads to group Normaliza.

c, j, k → summed out.

Basic

Multinomial distribution:

total n trial

Each trial : Possible k outcomes $\{E_1, E_2 \cdots E_k\}$
with prob $\{P_1, \cdots P_k\}$ respectively.

Lets assume $E_1 \xrightarrow{\text{happens}} n_1$, E_2 happens n_2 $\cdots$ $E_k \to n_k$ times.

So, $n_1 + n_2 \cdots + n_k = n$ // as n trial.

So, $$P = \frac{n!}{n_1!\, n_2! \cdots n_k!} P_1^{n_1} P_2^{n_2} \cdots P_k^{n_k}$$

$$= \frac{n!}{n_1!\, n_2! \cdots n_k!} \prod_{i=1}^{k} P_i^{n_i}$$

$$= n! \prod_{i=1}^{k} \frac{P_i^{n_i}}{n_i!}$$

straight extentions.

Binomial distribution: Bernoulli trials.

n times fliping x positive $n-x$ negatives.

$$P(X = x \mid n, p) = {}^{n}C_{x}\, p^{x} (1-p)^{n-x}$$

$\downarrow$ (Distribution over Distribution)

Beta distribution (Prior to Dirichlet Distribution)

what if the p is a distribution itself ??

$p \in [0, 1]$

$\boxed{\text{Beta is conjugate prior for Binomial}}$

Now, $$P(p \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} p^{\alpha - 1} (1-p)^{\beta - 1}$$

$$B(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\, \Gamma(\beta)} = \binom{\alpha - 1}{\alpha + \beta - 2}$$ // continuous estimation.

Basic

(iv)

Conjugate prior:

for some likelihood function, if we choose certain prior the posterior ends up being the same function then conjugate prior.

Dirichlet Distribution: Extended from multinomial probs.

Conjugate prior to

what about the probs of multinomials $P_1 \dots P_k$??

(1) $\sum P_i = 1$ // already known as each events prob needs to be 1.

condition

$$P(P = \{P_i\} \mid \alpha_i) = \frac{\prod_i \Gamma(\alpha_i)}{\prod \Gamma(\sum_i \alpha_i)} \prod_i P_i^{\alpha_i - 1}$$

↳ to multinomial.

→ Generalization of Beta.

→ Distribution over multinomials.

→ Conjugate prior of multinomial

↳ given the data the $\{P_i\}$ will also be Dirichlet distribution.

* Beta/Gamma functions are different:

$$\text{Beta}(x, y) = B(x, y) = \frac{\Gamma(x)\,\Gamma(y)}{\Gamma(x+y)}$$ // binomial coeffs.

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\, dx \; ; \; \Re(z) > 0$$

$\Gamma(z) = (z-1)!$ if z is a natural number ≥ 0 positive integer.