Q1.MLP W = oldW + learning rate * label y(i) * data x(i). 这里 y 的值是 1or-1. Forward/back 如下

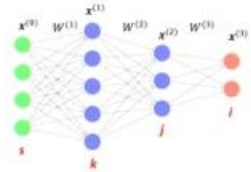
Example: 1 for "Y" and -1 for "N";

$\eta = 0.9$

x0	x1	x2	true label	w before update	predicted label	w after update
1	0	1	Y	(0.0, 0.0, 0.0)	N	(0.9, 0.0, 0.9)
1	1	1	N	(0.9, 0.0, 0.9)	Y	(0.0, -0.9, 0.0)
1	0	0	Y	(0.0, -0.9, 0.0)	N	(0.9, -0.9, 0.0)
1	1	0	Y	(0.9, -0.9, 0.0)	N	(1.8, 0.0, 0.0)

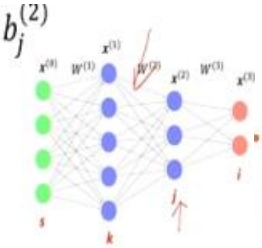
:

- Stochastic gradient for $W_{ij}^{(3)}$ and $b_i^{(3)}$
  - *Recall:*
    - $\mathcal{L} = \frac{1}{2}\left\|\mathbf{y} - \mathbf{x}^{(3)}\right\|^2 = \frac{1}{2}\sum_i \left(y_i - x_i^{(3)}\right)^2$
    - $x_i^{(3)} = f^{(3)}(z_i^{(3)})$
    - $z_i^{(3)} = \sum_j W_{ij}^{(3)} x_j^{(2)} + b_i^{(3)}$



- $\frac{\partial \mathcal{L}}{\partial W_{ij}^{(3)}} = \frac{\partial \mathcal{L}}{\partial x_i^{(3)}}\frac{\partial x_i^{(3)}}{\partial z_i^{(3)}}\frac{\partial z_i^{(3)}}{\partial W_{ij}^{(3)}} = \underbrace{-(y_i - x_i^{(3)})f'^{(3)}\left(z_i^{(3)}\right)}_{\delta_i^{(3)}} x_j^{(2)}$
- $\frac{\partial \mathcal{L}}{\partial b_i^{(3)}} = \frac{\partial \mathcal{L}}{\partial x_i^{(3)}}\frac{\partial x_i^{(3)}}{\partial z_i^{(3)}}\frac{\partial z_i^{(3)}}{\partial b_i^{(3)}} = -(y_i - x_i^{(3)})f'^{(3)}\left(z_i^{(3)}\right)$

- Stochastic gradient for $W_{jk}^{(2)}$ and $b_j^{(2)}$
  - *Recall:*
    - $\mathcal{L} = \frac{1}{2}\left\|\mathbf{y} - \mathbf{x}^{(3)}\right\|^2 = \frac{1}{2}\sum_i \left(y_i - x_i^{(3)}\right)^2$
    - $x_i^{(3)} = f^{(3)}\left(z_i^{(3)}\right)$; $z_i^{(3)} = \sum_j W_{ij}^{(3)} x_j^{(2)} + b_i^{(3)}$
    - $x_j^{(2)} = f^{(2)}\left(z_j^{(2)}\right)$; $z_j^{(2)} = \sum_k W_{jk}^{(2)} x_k^{(1)} + b_j^{(2)}$



- $\frac{\partial \mathcal{L}}{\partial W_{jk}^{(2)}} = \sum_i \frac{\partial \mathcal{L}}{\partial x_i^{(3)}}\frac{\partial x_i^{(3)}}{\partial z_i^{(3)}}\frac{\partial z_i^{(3)}}{\partial x_j^{(2)}}\frac{\partial x_j^{(2)}}{\partial z_j^{(2)}}\frac{\partial z_j^{(2)}}{\partial W_{jk}^{(2)}}$

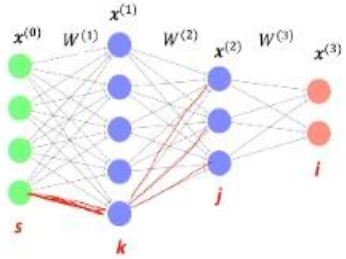
$$= \sum_i \underbrace{-(y_i - x_i^{(3)})f'^{(3)}\left(z_i^{(3)}\right)}_{\delta_i^{(3)}} w_{ij}^{(3)} f'^{(2)}\left(z_j^{(2)}\right) x_k^{(1)}$$

$\delta_i^{(3)}$

$\delta_j^{(2)}$

- Stochastic gradient for $W_{ks}^{(1)}$

$$\frac{\partial \mathcal{L}}{\partial W_{ks}^{(1)}} = \sum_j \delta_j^{(2)} W_{jk}^{(2)} f'^{(1)}\left(z_k^{(1)}\right) x_s^{(0)}$$



- Y: *continuous value* $(-\infty, +$
  - $y = \boldsymbol{x}^T\boldsymbol{\beta} + \varepsilon = \beta_0 + x_1\beta_1 + x_2\beta$
    - $\varepsilon \sim N(0, \sigma^2)$
  - $y|\boldsymbol{x}, \beta \sim N(\boldsymbol{x}^T\beta, \sigma^2)$

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad MSE$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Q2 linear Reg(加 1to x for bias)　　LogReg(Bernoulli 和 Poisson 的 MLE)

$$P_X(x_i; p) = p^{x_i}(1-p)^{(1-x_i)}, \text{ where } p \in [0,1], x_i \in \{0,1\}.$$

$$\ln p_X(x_1, \cdots, x_n; p) = \sum_{i=1}^{n} x_i \ln p + \sum_{i=1}^{n}(1 - x_i)\ln(1-p)$$

$$\frac{d \ln p_X}{dp} = \frac{\sum_{i=1}^{n} x_i}{p} + \frac{\sum_{i=1}^{n} x_i - n}{1-p} \overset{set}{=} 0$$

$$p_{MLE} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$P_X(x_i; \lambda) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

$$\ln p_X(x_1, \cdots, x_n; \lambda) = \sum_{i=1}^{n} x_i \ln(\lambda) - n\lambda - \sum_{i=1}^{n} x_i!$$

$$\frac{d \ln p_X}{d\lambda} = \frac{\sum_{i=1}^{n} x_i}{\lambda} - n \overset{set}{=} 0$$

$$\lambda_{MLE} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Naive Bayes MLE with Lagrangian Multiplier

Recall that given a corpus and labels for each document $D = \{(x_d, y_d)\}_{d=1}^{|D|}$, the log likelihood function of a Bayes model parameterized by $\Theta = (\beta_1, \beta_2, \beta_3, \pi)$ is:

Full unconstrained objective

$$L^*(\Theta) = L(\Theta) - \sum_j \lambda_j \left(\sum_{n=1}^{N} \beta_{j,n} - 1\right)$$

MLE derivation

$$\frac{\partial L^*(\Theta)}{\partial \beta_{j,n}} = \sum_{d:y_d=j} \beta_{j,n}^{-1} x_{dn} - \lambda_j \overset{set}{=} 0$$

$$\beta_{j,n}^{-1} = \frac{\lambda_j}{\sum_{d:y_d=j} x_{dn}} \quad \Longrightarrow \sum_{n=1}^{N} \beta_{j,n} = 1 = \sum_{n=1}^{N} \sum_{d:y_d=j} x_{dn}$$

$$\Longrightarrow \beta_{jn} = \frac{\sum_{d:y_d=j} x_{dn}}{\lambda_j} \quad \Longrightarrow \lambda_j = \sum_{n=1}^{N} \sum_{d:y_d=j} x_{dn}$$

$$\Longrightarrow \beta_{j,n} = \frac{\sum_{d:y_d=j} x_{dn}}{\sum_{n=1}^{N} \sum_{d:y_d=j} x_{dn}}$$

Step 4: get the full gradient: $\nabla_\beta L^*(\Theta)$ is a 3 by N matrix whose entries are $\beta_{j,n}$, $j \in \{1,2,3,\}, n \in \{1, \cdots, N\}$.

Recall the Dirichlet distribution, a family of continuous multivariate probability distribution used as prior to the multinomial distribution, with the PDF specified as:

$$f(\mathbf{x};\alpha) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}x_i^{\alpha_i-1}$$

where $\Gamma(x)$ is the Gamma function with derivative $d\ln(\Gamma(x))/dx = \frac{\Gamma'(x)}{\Gamma(x)} = \psi(x)$, the digamma function. Derive the maximum likelihood estimator for parameter α for the Dirichlet distribution.

找 log-likelihood function

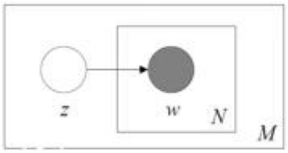
$$\ln f(x;\alpha) = \ln(\Gamma(\tilde{\alpha})) - \sum_{i=1}^{K}\ln\Gamma(\alpha_i) + \sum_{i=1}^{K}(\alpha_i - 1)\ln(x_i)$$

where $\tilde{\alpha} = \sum_{i=1}^{K}\alpha_i$. Now find its derivative w.r.t α_i and set it to zero to find an algebraic equation:

$$\frac{d\ln f(x;\alpha)}{d\alpha_i} = \psi(\tilde{\alpha})\frac{d\tilde{\alpha}}{d\alpha_i} - \frac{d}{d\alpha_i}(\sum_{j=1}^{K}\ln\Gamma(\alpha_j)) + \frac{d}{d\alpha_i}(\sum_{i=1}^{K}(\alpha_j - 1)\ln(x_i))$$

$$= \psi(\sum_{i=1}^{K}\alpha_i) - \psi(\alpha_i) - \sum_{i\neq j}\ln(x_j) \overset{set}{=} 0$$

Then use a numerical scheme to find root. The full gradient is $\nabla_\alpha \ln f(x;\alpha) = [\frac{d\ln f(x;\alpha)}{d\alpha_1}, \cdots, \frac{d\ln f(x;\alpha)}{d\alpha_K}]$



Q3: N:word 数 M:topic 数.Generative process:先 sample 一个 topic 再产生 N words given that topic

Joint dis 和 conditional dis

$$p(w,z) = p(w|z)p(z)$$

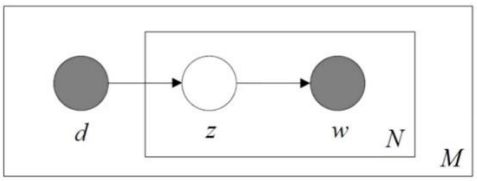
$$p(w = w_l, z = k) = p(w_l = w|z = k)p(z = k),$$

Joint distribution of N variables 和 entire one

$$p(w_{1:N}, z) = \prod_{n=1}^{N} p(w_n|z)p(z)$$

$$p(w_{1:N}, z_{1:M}) = \prod_{m=1}^{M} p(z_m)\prod_{n=1}^{N} p(w_n|z_m) \quad p(w_{1:N}) = \prod_{n=1}^{N} p(w_n) = \prod_{n=1}^{N}\sum_{z} p(w_n, z) = \prod_{n=1}^{N}\sum_{z} p(w_n|z)p(z)$$

pLSA model M doc, N word



Generative process:

- first sample a document from M documents
- For each document, generate N words, each with its own topic. (one Z responsible for 1 w's) e.g., 1-to-1 map between Z and W
- Note that one d is responsible for N z's and N w's. e.g., 1-to-N map between d and Z and between d to w.

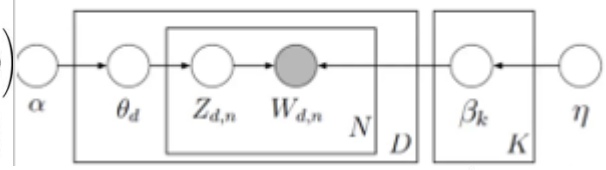
$$p(w,d,z) = p(w|z,d)p(z|d)p(d)$$

$$p(w_{1:N}, z_{1:N}, d) = \prod_{n=1}^{N} p(w_n|z_n,d)p(z_n|d)p(d)$$

$$p(w_{1:N}, z_{1:N}, d_{1:M}) = \prod_{m=1}^{M} p(d_m)\left(\prod_{n=1}^{N} p(w_n|z_n,d_m)p(z_n|d_m)\right)$$

$$p(w_{1:N}, d_{1:M}) = \prod_{m=1}^{M} p(d_m)\prod_{n=1}^{N}\left(\sum_{z} p(w_n|z_n,d_m)p(z_n|d_m)\right)$$

Bayesian pLSA



$$p(\theta, Z, W, \beta) = p(W|Z,\beta)p(Z|\theta)p(\theta)p(\beta) \quad p(\theta_{1:D}, Z_{1:N}, W_{1:N}, \beta_{1:K}) = \prod_{k=1}^{K} p(\beta_k)\prod_{d=1}^{D} p(\theta_d)\left(\prod_{n=1}^{N} p(Z_{d,n}|\theta_d)p(W_{d.n}|Z_{d,n},\beta_k)\right)$$

Derive the following quantities in the pLSA model and show all your steps: (1) The joint density of all random variables $p(w,d,z;\theta,\beta)$; (2) The conditional density $p(z|w,d;\theta,\beta)$. and (3) The conditional density of $p(w|z,d;\theta,\beta)$. (15 pts) (note in this notation, θ,β are given as model hyperparameters, hence they are separated from the random variable using ';')

(a) $p(w,d,z,;\theta,\beta) = p(w|d,z)p(z|d)p(d)$.

(b) $p(z|w,d;\theta,\beta) = p(w,d,z)/p(w,d) = \frac{p(w|d,z)p(z|d)p(d)}{\sum_{z'} p(w|d,z')p(z'|d)p(d)}$

(c) $p(w|z,d;\theta,\beta)$ already given in lecture slide.

Write the log-likelihood function of the pLSA model as sum of two terms: the Evidence Lower Bound (ELBO) and the KL divergence between a variational distribution and the latent posterior. Point out each term in your answer. (10 pts)

Solution: The general ELBO-KL decomposition for a latent variable model with latent variable Z and data X can be written as:

$$\log p(X) = \mathbb{E}_q[\log p(X,Z) - \log q(Z)] - KL(q(Z)||p(Z|X))$$

In this problem we have $Z = \{z\}$ the topic, and $X = \{w,d\}$. Therefore, for the decomposition in this case we just plug in the values abstractly (this is enough for this problem):

$$\log p(w,d;\theta,\beta) = \mathbb{E}_q[\log p(w,d,z;\theta,\beta) - \log q(z|w,d;\phi)] - KL(q(z|w,d;\phi)||p(z|w,d))$$

Where $q(z|w,d;\phi)$ is a variational family of distribution parameterized by free parameter ϕ. If you attempted to derive it in more detailed form, you also get the full mark.

$$p(\theta) = p(\mu) = N(\mu; 0, I) \propto \exp\{-\frac{1}{2}\mu^T\mu\}$$

Use the *pdf kernel method* to derive the posterior distribution the following conjugate pair: prior $p(\theta) = \mathcal{N}(\theta; 0, I)$ and likelihood function $p(x; \theta) = \mathcal{N}(x; \mu, \Sigma)$, show all steps. (5 pts)

$$p(x|\theta) = p(x|\mu) = N(x; \mu, \Sigma) \propto \exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

Solution: The PDF kernel method simply looks at the kernel part of the pdf. In this case we have:

$$p(\theta|x) = p(\mu|x) \propto p(\mu)p(x|\mu) \propto \exp\{-\frac{1}{2}(\mu^T\mu + x^T\Sigma x - \mu^T\Sigma^{-1}x - x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu + \mu^T\Sigma^{-1}\mu)\}$$

$$\propto \exp\{(u-\hat{\mu})^T\hat{\Sigma}^{-1}(\mu-\hat{\mu}\}, \hat{\mu} = (\Sigma^{-1}+I)^{-1}\Sigma x, \hat{\Sigma} = (\Sigma^{-1}+I)$$

Given a set of binary outcomes (successes and failures) from independent Bernoulli trials, and assuming a Beta distribution as the prior for the probability of success p, derive the posterior distribution of p after observing the data.

1. **Write down the prior distribution:**
   $p(p) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha,\beta)}$
   where $B(\alpha, \beta)$ is the beta function.
2. **Write down the likelihood for Bernoulli trials:**
   $p(x|p) = p^s(1-p)^{n-s}$
   where s is the number of successes and n is the number of trials.
3. **Write the posterior distribution:**
   The posterior is proportional to the product of the prior and the likelihood:
   $p(p|x) \propto p(p) \times p(x|p)$
4. **Combine the terms and simplify:**
   $p(p|x) \propto p^{\alpha+s-1}(1-p)^{\beta+n-s-1}$
5. **Recognize the kernel of a Beta distribution:**
   The expression matches the kernel of a Beta distribution with updated par:
   $p(p|x) = \text{Beta}(\alpha+s, \beta+n-s)$

Q4:Eigenval 0 的数量 in L=连接的 Component 数 quadratic form: $x^TLx = \sum_{(i,j)\in E}(x_i - x_j)^2$ normalized graph L:

$L_{\text{rw}} = D^{-1}L = I - D^{-1}W$ $L_{\text{sym}} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}$ Weighted graph: $x^TLx = \sum_{(i,j)\in E} w_{ij}(x_i - x_j)^2$

Minimization of the Rayleigh quotient relies on the theorem

Theorem 2.2. *If $A_{n\times n}$ is a real PSD matrix with eigenvalues $0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ wit* *eigenbasis $(v_1, \cdots, v_n)$ then*

Note that for $\mathcal{L}$, $\lambda_1 = 0$ and $v_1 = (1, \cdots, 1)$, so that

$$\min_{\boldsymbol{x}\perp\langle v_1, v_2, \cdots, v_k\rangle, \|\boldsymbol{x}\|\neq 0} R_{\mathcal{L}}(\boldsymbol{x}) = \lambda_{k+1}$$

$$\min_{\boldsymbol{x}\perp(1,1,\cdots,1), \|\boldsymbol{x}\|\neq 0} R_{\mathcal{L}}(\boldsymbol{x}) = \lambda_2$$

Lemma 7. *A real symmetric matrix $\mathbf{M}$ is PSD if and only if $\lambda_1(\mathbf{M}) \ge 0$*

Proof. If $\mathbf{M}$ is PSD then for $v_1 = v_1(\mathbf{M})$ and $\lambda_1 = \lambda_1(\mathbf{M})$ we have

$$0 \le v_1^\top \mathbf{M} v_1 = \lambda_1 \cdot v_1^\top v_1 = \lambda.$$

On the other hand if $\lambda_1(\mathbf{M}) \ge 0$ then for all x we have

$$x^\top \mathbf{M} x = \sum_{i\in[n]} \lambda_i(\mathbf{M}) \cdot \left[v_i(\mathbf{M})^\top x\right]^2 \ge \lambda_1(\mathbf{M}) \sum_{i\in[n]} \left[v_i(\mathbf{M})^\top x\right]^2 \ge 0.$$

With this in mind, what is $\lambda_1(\mathcal{L})$? Well, we have already shown that $\mathcal{L}\vec{1} = \vec{0}$ and consequently $\mathcal{L}$ has a 0 eigenvalue. Furthermore, since all eigenvalues of $\mathcal{L}$ are non-negative this implies that $\lambda_1(\mathcal{L}) = 0$.

Now what is $\lambda_2(\mathcal{L})$? Can it be the case that $\lambda_2(\mathcal{L}) = 0$? Or more broadly, what are the 0 eigenvalues of $\mathcal{L}$, or what is $\ker(\mathcal{L})$?

A common starting point in spectral graph theory to address such questions is to look at the quadratic form of $\mathcal{L}$. Suppose $x \in \ker(\mathcal{L})$ with $x \neq 0$, i.e. $\mathcal{L}x = \vec{0}$. In this case

$$0 = x^\top \mathcal{L} x = \sum_{\{i,j\}\in E} w_{ij}(x_i - x_j)^2.$$

Now, since $x \neq 0$ there must be some $i \in V$ with $x_i \neq 0$. Given this, what are the values of x_j for $j \in N(i)$. Note that if $x_j \neq x_i$ then since $\{i,j\} \in E$, $w_{ij} > 0$, and $(x_i - x_j)^2 > 0$ we have that $x^\top \mathcal{L} x > 0$. Consequently, $x_j = x_i$ for all $j \in N(i)$. Repeating this argument we see that $x_j = x_i$ for all j that are neighbors of neighbors of i. As we have seen, repeating this argument ultimately implies that $x_j = x_i$ for all j that are in the same connected component as i. Moreover, it is not to hard to see that this property suffices for a vector to be in the kernel of $\mathcal{L}$.

- The smallest eigenvalue of L is 0, and the corresponding eigenvector is the constant one vector $\mathbf{1}$

$L\vec{v} = \lambda\vec{v}$

$L\vec{1} \neq 0\cdot\vec{1} = \vec{0}$

$L\vec{1} = (D-W)\vec{1} = D\cdot\vec{1} - W\cdot\vec{1}$

$$\begin{pmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{pmatrix}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix}$$

$$W\cdot\vec{1} = \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ w_{21} & & w_{2n} \\ \vdots & & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix}$$

- For any vector $f \in R^n$, $f'Lf = \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2$

left: $f'Lf = f'(D-W)f = f'Df - f'Wf$

$$= \sum_i d_i f_i^2 - \sum_{ij} w_{ij} f_i f_j$$

right: $\frac{1}{2}\sum_{ij} w_{ij}(f_i^2 + f_j^2 - 2f_if_j)$

$$= \frac{1}{2}\sum_{ij} w_{ij}f_i^2 + \frac{1}{2}\sum_{ij} w_{ij}f_j^2 - \sum_{ij} w_{ij}f_if_j$$

$$= \frac{1}{2}\sum_i f_i^2\left(\sum_j w_{ij}\right) + \frac{1}{2}\sum_j f_j^2\left(\sum_i w_{ij}\right) - \sum_{ij} w_{ij}f_if_j$$

$$= \frac{1}{2}\sum_i f_i^2 d_i + \frac{1}{2}\sum_j f_j^2 d_j - \cdots$$

$$= \sum f_i^2 d_i - \cdots$$

$$(f_1, f_2, \cdots f_n)\begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$$

- L is symmetric and positive semi-definite
  - Undirected graph => symmetric
  - Property 1 => $f'Lf \geq 0$ => positive semi-definite, by definition

LogisReg: **概念：** 属于 **Discriminative** models(比如 logistic regression), model the conditional probability P(Y | X) directly. + 能用于 binary 和 multi-class 分类. 适合(tasks 有大量 labeled data 和 clear decision boundaries)
MSE 不适用(会有 non-convex optimiz 问题) + decision boundary 是 linear, 但可以用 non-linear transform of input features(比如 polynomial features) log reg 能捕捉 feature 间复杂关系(->flexible) + 相当于 1 层 NN + sigmoid(cross-entropy loss 训练). sigmoid activation 输出 prob(0~1) + 2 个 class linearly separable, scale up β 变成 cβ, decision boundary 不变,但 likelihood func 会变,prob 更靠近 0 or 1,不会 converge,最后 explode(所以需要 L1 or L2 regularization) + logistic func $\sigma(z)$ 代表 probability p(属于某 class) (e.g., spam), not log odds. The log odds would be the logarithm of odds ratio $\log_{f_0}(p/1-p)$ + k class 对应 k neuron,每个 neuron 用的 Softmax activation func,输出 prob 加起来是 1 + minimize CE loss = maximize log-likelihood func~~~~Bernoulli dis 用 Maximum Likelihood Estimation(MLE)来确定模型 param, 通过 likelihood func 最大化观测到的数据属于其观测 class 的概率。对于 binary 问题, logistic reg 的输出是[0~1]间的一个概率值，表示属于 class 的概率.**Input**: p hat(predicted prob of ith observ belong to class1),观测到的 y 的概率可以用 Bernoulli dis 表示, θ 是 model param, x 是 input data,y 是 observed output(target), binary 里 input 是个 neuron $P(y|x;\theta) = \hat{p}^y(1-\hat{p})^{1-y}$ likelihood func(观测值的概率的乘积,n 个 sample): $L(\theta|x,y) = \prod_{i=1}^{n} P(y_i|x_i;\theta)$

log-likelihood func(方便计算和优化，找最佳 param θ) $\ell(\theta|x,y) = \sum_{i=1}^{n} (y_i \log(\hat{p}_i) + (1-y_i)\log(1-\hat{p}_i))$ **Output**: numerical measure of how well the model parameters explain the observed data!! ~~~~

The output of the logistic function $\sigma(z)$ represents the log odds (i.e., $\log(\frac{p}{1-p})$), where p is the probability of an email being classified as spam, and $z = \theta^\top x$.

对, logistic function (sigmoid function) outputs values 0~1,表示 prob of the positive class (prob email 被分类成 spam). The log odds transformation (logit function) maps these probabilities to the real number line, making it suitable for regression//Decision threshold is 0.5, then model 分类 email 是 spam(if log func ouput>0.5);反之，分类成不是 spam--对，binary classification(用 logistic reg),output=分类成 positive class 的概率(>0.5 的话就属于 positive class) **LinearReg**: 相当于 1 层 NNwith identity activation(no activation,因为 only pass input) (用 MSE loss 训练) + 不用 ReLU acti(会 introduce non-linearities) 需要最小化的 **Optimization func**:y 是真实值,yi hat 是 model 预测值,输出 RSS 和 MSE,公式在 page1

Naive Bayes, Bayes' threorem $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$ **概念：Generative models**, model joint probability distribution P(X, Y)--data X, labels Y , 用来产生 new data(by sample from distribution). + 能用于 binary 和 multi-class 分类 + 不属于 latent variable models,因为没 unobserved latent factors. 某个 word a 出现在 class c 的概率 β = (word 在 class 出现的次数+1)/c 里的总 word 数 +Vocabulary 里的 word 数 With add-1 smoothing 意思是在分子+1，分母+Vocabulary 里的 word 数
Π(某 class) = 属于那个 class 的 doc 数/总 doc 数

Data:

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Vocabulary:

Index	1	2	3	4	5	6
Word	Chinese	Beijing	Shanghai	Macao	Tokyo	Japar

- Learned parameters (with smoothing):

$\hat{\beta}_{c1} = \frac{5+1}{8+6} = \frac{3}{7}$, $\hat{\beta}_{j1} = \frac{1+1}{3+6} = \frac{2}{9}$
$\hat{\beta}_{c2} = \frac{1+1}{8+6} = \frac{1}{7}$, $\hat{\beta}_{j2} = \frac{0+1}{3+6} = \frac{1}{9}$
$\hat{\beta}_{c3} = \frac{1+1}{8+6} = \frac{1}{7}$, $\hat{\beta}_{j3} = \frac{0+1}{3+6} = \frac{1}{9}$
$\hat{\beta}_{c4} = \frac{1+1}{8+6} = \frac{1}{7}$, $\hat{\beta}_{j4} = \frac{0+1}{3+6} = \frac{1}{9}$
$\hat{\beta}_{c5} = \frac{0+1}{8+6} = \frac{1}{14}$, $\hat{\beta}_{j5} = \frac{1+1}{3+6} = \frac{2}{9}$

$\hat{\pi}_c = \frac{3}{4}$, $\hat{\pi}_j = \frac{1}{4}$

- For the test document d=5, compute

$p(y_5 = c|\boldsymbol{x}_5) \propto p(y_5 = c) \times \prod_n \beta_{cn}^{x_{5n}} = \frac{3}{4} \times \left(\frac{3}{7}\right)^3 \times \left(\frac{1}{14}\right) \times \left(\frac{1}{14}\right) \approx 0.0003$

$p(y_5 = j|\boldsymbol{x}_5) \propto p(y_5 = j) \times \prod_n \beta_{jn}^{x_{5n}} = \frac{1}{4} \times \left(\frac{2}{9}\right)^3 \times \left(\frac{2}{9}\right) \times \left(\frac{2}{9}\right) \approx 0.0001$

所以 x5 属于 class c。3/7 因为 doc5 里的 Chinese 的 beta 是 3/7,3 次方因为出现在 Doc5 里出现 3 次, 两个 1/14 因为 Tokyo 和 Japan 的 beta 都是 1/14.**Attributes are conditionally independent given class(class conditional independency)**

Word Embedding **Skip-gram**:用 target word predict 旁边 word,更好得处理罕见(rare)words,计算效率比 CBOW(平均 context's embedding)低,但对大数据量更 effective 而且能捕捉更 detailed semantic 关系(更高 semantic accuracy,因为预测了很

多 context words). **问题**:word 多计算贵(要对每个词评分和 nomal)->用 **Negative Sampling** 把 multi classification 变成 binary classification set(从预测 context 位置->对于 positve sample(actual context word)选 negative sample(随机词,不是 target 的上下文),再最大化 positve sample 的 score, 最小化 negative 的 **Transformer 和 LLM(概念)**Encode(represent)+Decode(Generation).**Input**(Tokenizer-embedd 层(word2vector)-positional encode)-**Encoding**(Self-Attention)-**Decoding**(Self-Attention)-**Output**(Linear Layer+Softmax). En/Decode 步有 add 和 norm 加和 norm 层. Special token EOS 加在末尾,之后开始 decode,但序列 len 太长不行,用 attention 把 input 里的子集 word 去预测 target 而不是全用+不用把序列变成 vector, 计算查询与一系列 key 之间的相似度或关联度决定对应 value 的加权重要性.**BERT:**only Encoder, 随机把 input token 变成 mask,每个 word attain 两侧所有 word,双向 context.**GPT :only Encoder**, pre-train 预测 next token,从左到右每个 word attain 之前的那一个 **Graph Spectral Clustering** 用 eigenvalue 和 eigenvector 分组 data. node=data,edge=similarity. 算 **Laplacian** 矩阵的前 k 个 eigenvector,每个节点用 k 个 val 表示，用 eigenval 降维，再用如 K-means 来 cluster. 适合复杂 shape 的 data 以及 connectivity or graph structures 更 informative for clustering than distance in the original feature space.**Laplacian** 矩阵 L=D(diagonal matrix,每个点是 node 的度,除了对角线都是 0)-W(adjcency matrix). L 最小的 eigenVal 是 0(其他都大于 0),对应的 eigenVec 是 vector1(all-one vector,n dimension,能用 nonzero 数 scale) For any vector $f \in R^n$, $f'Lf = \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2$ 对角 matrix L 有 k 个 eigenval 0,eigvectors:(1,1,0,0)(0,1,1,0) **Label Propagatio** 图中两节点紧密连接->很可能属于同一 class. 少数 node 有 class,算法通过图结构把 class 从 labeled 传到 unlabeled node.Laplacian 定义 propagation rule**K-Means** 适合 dataset 的(clusters 是球形+有相似 size 和 density), 不同 initial 导致不同结果 + k 的选择影响 result + converge when 簇心变化小于某值 or 最大迭代次数. 属于 hard classificaiton**GMM** 有多个 class 所以多个 distribution + dataset cluster 可以非球形和变化 densities,基于 EM 优化 param, 属于 latent variable models + 属于 soft Classificaiton(方差平方=covariance=0 后 soft assign 变成 hard assign) + GMM=soft K-means + covariance matrix 决定分布的形状(方向,宽度) + Joint probability density function(pdf) of marginal distribution of entire dataset under(GMM)：

$$f_X(x_1, \cdots, x_n) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n; \mu_k, \Sigma_k)$$

$f_X(x_1, \ldots, x_n)$: This represents the joint probability density function of the dataset X, where X consists of N data points $(x_1, \ldots, x_n)$ in a multidimensional space (e.g., $\mathbb{R}^d$).

$\prod_{n=1}^{N}$: This is the product operator, indicating that the expression to the right is multiplied together for each data point x_n from 1 to N, where N is the total number of data points in the dataset. This reflects the assumption of independence among the data points in terms of their generation process.

$\sum_{k=1}^{K}$: This is the summation operator, summing over the K components (or clusters) in the mixture model. It represents the mixture aspect of the model, where each data point is

π_k: These are the mixing coefficients (or mixture weights) for each of the K Gaussian components in the mixture model. Each π_k represents the prior probability that a randomly selected data point belongs to cluster k. The mixing coefficients must satisfy two conditions: $0 \leq \pi_k \leq 1$ for all k, and $\sum_{k=1}^{K} \pi_k = 1$, ensuring that they form a valid probability distribution over the clusters.

$\mathcal{N}(x_n; \mu_k, \Sigma_k)$: This denotes the probability density function of a Gaussian (or normal)

$$P(x|\gamma_1) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right)$$

$$P(x|\gamma_2) = \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)$$

distribution for the k^{th} component, evaluated at data point x_n. Each Gaussian component is characterized by:

- μ_k: The mean vector of the k^{th} Gaussian component, indicating the center of the cluster in the data space.
- Σ_k: The covariance matrix of the k^{th} Gaussian component, defining the shape and orientation of the cluster. The covariance matrix determines how spread out the cluster is in each dimension and the correlations between dimensions.

Mutinomial Mixture model Doc=多个 topic，topic=multinomial distribution，描述该主题下每个词出现的概率分布。生成 doc 步骤: 随机选个主题，从该主题的词汇分布生成单词。 likelihood func

For each document d

- Sample its cluster label $z \sim Categorical(\boldsymbol{\pi})$
  - $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)$, π_k is the proportion of jth cluster
  - $p(z = k) = \pi_k$
- Sample its word vector $\boldsymbol{x}_d \sim multinomial(\boldsymbol{\beta}_z)$
  - $\boldsymbol{\beta}_z = (\beta_{z1}, \beta_{z2}, \ldots, \beta_{zN})$, β_{zn} is the parameter associate with nth word in the vo
  - $p(\boldsymbol{x}_d | z = k) = \frac{(\sum_n x_{dn})!}{\prod_n x_{dn}!} \prod_n \beta_{kn}^{x_{dn}} \propto \prod_n \beta_{kn}^{x_{dn}}$

$$= \prod_d \frac{(\sum_n x_{dn})!}{\prod_n x_{dn}!} \sum_k p(z = k) \prod_n \beta_{kn}^{x_{dn}}$$

pLSA

Probabilistic Latent Semantic Analysis **概念:**latent variable 是 topic,word 被 latent topic 产生 + 不用 Dirichlet dis + models 每个 doc as mixture of topics(topics are distributions over words) + 不是 generative model,不能 model new doc(因为 pLSA 分配一组 topic distribution

param 到每个 doc(在 train set)->param 很难 assign 到 unseen doc(without retrain). LDA 通过 add priors to θ 和 β 变成 generative model 来解决:model doc as random mixtures over latent topic where the mixture weight drawn from Dirichlet dis)

- Probability of a word w

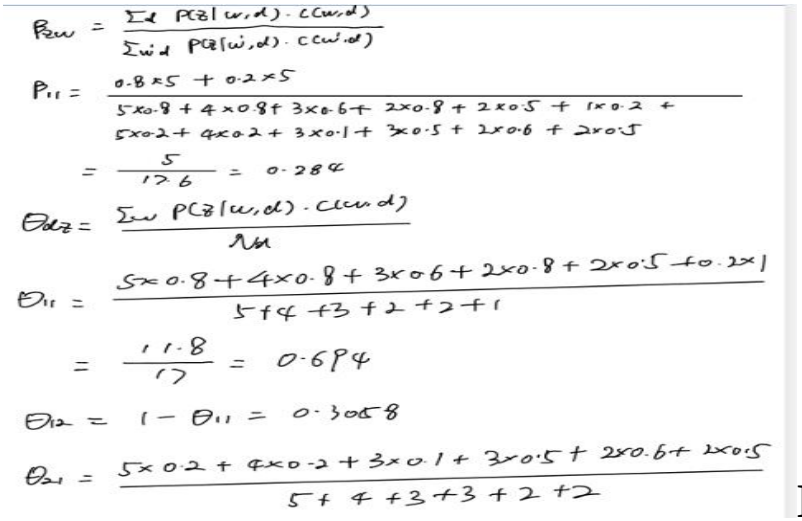
$$p(w|d,\theta,\beta) = \sum_k p(w, z=k|d,\theta,\beta) = \sum_k p(w|z=k,d,\theta,\beta)p(z=k|d,\theta,\beta) = \sum_k \beta_{kw}\theta_{dk}$$

- Likelihood of a corpus

$$\begin{aligned} & \prod_{d=1} P(w_1,\cdots,w_{N_d},d|\theta,\beta,\pi) \\ = & \prod_{d=1} P(d)\left\{\prod_{n=1}^{N_d}\left(\sum_k P(z_n=k|d,\theta_d)P(w_n|\beta_k)\right)\right\} \\ = & \prod_{d=1} \pi_d\left\{\prod_{n=1}^{N_d}\left(\sum_k \theta_{dk}\beta_{kw_n}\right)\right\} \end{aligned}$$

π_d *is usually considered as uniform*, i.e., 1/M

likelihood func, EM 估计文档-主题分布和主题-词汇分布的参数:通过 Maximize likelihood func 找最可能生成观察到的文本数据的模型参数.E 步:根据参数估计算每个词属于每个主题的概率。M 步更新模型参数来最大化，基于 E 步得到的条件概率分配



- M-step

$$\beta_{11} = \frac{0.8*5+0.5*2}{11.8+5.8} = 5/17.6$$
$$\beta_{12} = \frac{0.8*4+0.5*3}{11.8+5.8} = 4.7/17.6$$
$$\beta_{13} = 3/17.6$$
$$\beta_{14} = 1.6/17.6$$
$$\beta_{15} = 1.3/17.6$$
$$\beta_{16} = 1.2/17.6$$
$$\beta_{17} = 0.8/17.6$$

$$\theta_{11} = \frac{11.8}{17}$$
$$\theta_{12} = \frac{5.2}{17}$$

LDA

概念： 1.Dirichlet dis + distributions of topics over documents and words over topics 2.能 model unseen doc by model distri of doc. doc=mixtures of topics,topics =mixtures of words. doc 里的 topic dis 用 conjugate pair(简化 infer hidden structure) of Dirichlet dis model(允许 model infer topic dis for doc unseen during train)3.不用 EM(因为 posterior dis 的复杂性),infer 和 learn 用 Variational EM,Variational Bayes,MC(Gibbs Sampling)**EM** 知道 data point 属于哪个 Gaussin distribution,每个 dis 的 Mixing Coefficients 的和=1, GMM 是 unsupervised 分类模型，没有用点的类别学习模型 param. E 步预测每个点属于不同类别的概率(Implicit variable),计算 tight original objective func 的 lower bound at θold;M 步用估计的分类(Implicit variable)来更新 Gaussion 分布的 mean,方差和先验概率来 maximum 数据的 likelihood func,maxinmize lower bound at θnew。用 jensen inequality 找 lower bound. 不同的 initial paramθ 会有不同 result，难优化，迭代次数多，容易到 local optim + k-means 属于 EM 特例 **Deep Learning, CNN** Relu 正区间内保持 gradient 恒定，缓解梯度消失问题,input<0 时梯度为 0,可能导致部分神经元不再更新(die). Sigmoid 输出 0~1 适用 binary 的输出层.Tanh 输出-1~1，比 Sigmoid 有更宽的输出范围.后两个 func 在 input 较大或较小时梯度接近 0 可能梯度消失,过大会 gradient explode. **1.**backpropagation 时候 error(loss) 从输出层到第一层,更新 weight 去最小化 loss func **2.**CNN 里 Pooling 层减小 representation 的 spatial size(宽高)和共享 weight 来减小 param 数(不太会 overfitting)而且 contributes to model's invariance to small translations of input image. **3.**convolutional 层作用是通过 filter(检测 patterns or texture)从 input image 学 features representation **4.**CNN 里 Softmax 输出层用 non-linear activation func 确保 output values are distributed(ouput sum=1),适合 multi-class classification **5**.CNN 通过堆叠多个卷积和池化层来增加网络的深度，使得网络能够学习从简单到复杂的特征层次结构 **Graph Embedding** 降维到 vector 表示(limit:high dimension, No global structure info integrated,permutaion-variant 输入变输出不变),1st order,节点相连则临近性=1,object func 最小化 empirical link dis 和 modeled link dis 的 KL divergence. 2nd:邻居相似则两节点相似(negative sampling 优化)