$$J(m,b) = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

want to find m, b to minimize J, i.e. $\frac{\partial J}{\partial m} = 0$, $\frac{\partial J}{\partial b} = 0$

$$\frac{\partial J}{\partial m} = \sum_{i=1}^{n} 2(y_i - mx_i - b)(-x_i) = 0 \qquad ①$$

$$\frac{\partial J}{\partial b} = \sum_{i=1}^{n} 2(y_i - mx_i - b)(-1) = 0 \qquad ②$$

$$\Rightarrow \sum_{i=1}^{n} y_i x_i - m\sum x_i^2 - b\sum x_i = 0 \qquad \text{from } ①$$

and $$\sum_{i=1}^{n} y_i - m\sum x_i - b\cancel{\sum 1}^{\,n} = 0 \qquad \text{from } ②$$

This is a system of equations of the form

$$m a_1 + b b_1 = c_1$$
$$m a_2 + b b_2 = c_2$$

Using Cramer's Rule gives a closed form

$$m = \frac{n\sum x_i y_i - (\sum x_i)(\sum y_i)}{n\sum x_i^2 - (\sum x_i)^2} \qquad b = \frac{(\sum x_i^2)(\sum y_i) - (\sum x_i)(\sum x_i y_i)}{n\sum x_i^2 - (\sum x_i)^2}$$

$N \gg n$

$$\underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}}_{\vec{y}} = \underbrace{\begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & & \ddots & \vdots \\ 1 & x_1^{(N)} & \dots & x_n^{(N)} \end{bmatrix}}_{\underline{X}} \underbrace{\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}}_{\vec{\theta}}$$

design matrix/
domain data matrix

$\vec{y} = \underline{X}\,\vec{\theta}$ ← variables we need to determine

known

Let cost function $J(\vec{\theta}) = \|(\vec{y} - X\theta)\|^2 = \|\vec{\epsilon}\|^2$

We want to minimize ∇J, set $\nabla J = \vec{0}$.

$$\nabla_{\vec{\theta}} J(\vec{\theta}) = (-X)\cdot(\vec{y} - X\vec{\theta}) + (\vec{y} - X\vec{\theta})\cdot(-X)$$
$$= -2X\cdot(\vec{y} - X\vec{\theta}) = 0$$
$$\Rightarrow X^T(\vec{y} - X\vec{\theta}) = 0$$
$$\Rightarrow \boxed{X^T\vec{y} - X^T X\vec{\theta} = 0} \quad \text{normal equation}$$

If $X^T X$ invertible,

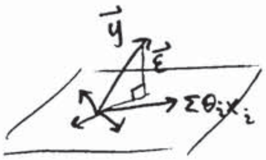
$$\vec{\theta} = (X^T X)^{-1} X^T \vec{y}$$

Geometric Approach:

$$\vec{y} = \underbrace{\theta_0 \vec{x}_0 + \dots + \theta_n \vec{x}_n}_{\text{in } \operatorname{span}\{\vec{x}_0, \dots, \vec{x}_n\}} + \vec{\varepsilon} \qquad \vec{\varepsilon} = \vec{y} - X\vec{\theta}$$

in $\underline{\operatorname{span}\{\vec{x}_0, \dots, \vec{x}_n\}}$

column space of X



Geometrically, $\vec{\varepsilon}$ is minimized iff

$$\vec{\varepsilon} \perp \operatorname{span}\{\vec{x}_0, \dots, \vec{x}_n\}$$

iff

$$\begin{cases} \vec{\varepsilon} \perp \vec{x}_0 = 0 \Rightarrow \vec{x}_1^T \vec{\varepsilon} = 0 \\ \vdots \\ \vec{\varepsilon} \perp \vec{x}_n = 0 \Rightarrow \vec{x}_n^T \vec{\varepsilon} = 0 \end{cases}$$

$$\Longleftrightarrow \begin{pmatrix} \vec{x}_0^T \\ \vdots \\ \vec{x}_n^T \end{pmatrix} \vec{\varepsilon} = 0 \Rightarrow X^T \varepsilon = 0$$

$$\Rightarrow X^T(\vec{y} - X\vec{\theta}) = 0$$

$$\Rightarrow X^T\vec{y} - (X^T X)\vec{\theta} = 0 \quad \text{(same normal equation)}$$

Probabilistic Approach:

$$y^{(i)} = \theta^T \vec{x}^{(i)} + \varepsilon^{(i)}$$

$\leftarrow$ error term $\rightarrow$ ① unmodel facts, ② random noise

Assume $\varepsilon^{(i)}$ are distributed IID (independently and identically distributed) according to a Gaussian distribution, i.e. $\varepsilon^{(i)} \sim N(\underset{\text{mean}}{0}, \underset{\text{variance}}{\sigma^2})$

$$P(\varepsilon^{(i)}) = \frac{1}{2\pi\sigma} e^{-\frac{(\varepsilon^{(i)} - 0)^2}{2\sigma^2}} = \frac{1}{2\pi\sigma} e^{-\frac{(y^{(i)} - \theta^T \vec{x}^{(i)})^2}{2\sigma^2}} \triangleq \underbrace{P(y^{(i)} \mid \vec{x}^{(i)}; \theta)}_{\text{The distribution of } y^{(i)} \text{ given } \vec{x}^{(i)} \text{ and parameterized by } \theta.}$$

denoted by

parameterized probability model

Also OK to write $y^{(i)} \mid \vec{x}^{(i)}; \theta \sim N(\theta^T \vec{x}^{(i)}, \sigma^2)$.

Cost function: $L(\theta) = \prod_{i=1}^{N} P(y^{(i)} \mid x^{(i)}; \theta) = \prod_{i=1}^{N} \frac{1}{2\pi\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$

(Also written as $L(\theta) = L(\theta; X, \vec{y}) \triangleq p(\vec{y} \mid X; \theta)$)

Q: Given this probabilistic model relating $y^{(i)}$'s and $x^{(i)}$'s, what is a reasonable way of choosing our best guess of the parameter θ?

Key: The principle of Maximum Likelihood Estimation (MLE) says that "choosing θ so as to make data as high probability as possible", i.e. we choose θ to maximize $L(\theta)$

To maximize $L(\theta)$, it is equivalent to maximize any strictly increasing function. Since $L(\theta)$ is defined by exp, and the key term is in the exponent, and log is an increasing function, we consider $\log L(\theta)$ (denoted $l(\theta)$, called log likelihood function)

$$l(\theta) = \log L(\theta) = \log \prod_{i=1}^{N} \frac{1}{2\pi\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$= N \log \frac{1}{2\pi\sigma} - \frac{1}{\sigma^2} \frac{1}{2} \sum_{i=1}^{N} (y^{(i)} - \theta^T x^{(i)})^2$$

<u>Note</u>: Maxmizing $l(\theta)$ is equivalent to minimizing $\sum_{i=1}^{N} (y^{(i)} - \theta^T x^{(i)})^2$, which we recognize as the original cost function in the least squares problem.

Def: A symmetric matrix $A_{n\times n}$ is called <u>positive definite</u> if $\forall \vec{x} \in \mathbb{R}^n$, $\vec{x}^T A \vec{x} \geq 0$ and equality holds only when $\vec{x} = \vec{0}$.

Quadratic Surfaces

$$\vec{x}^T A \vec{x} = g$$

$n=2$

$$(x, y)\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = g \Rightarrow a_{11}x^2 + 2a_{12}xy + a_{22}y^2 = g$$

So given a quadratic curve, we can put it in matrix form.

<u>Ex</u>: $2x^2 + 3xy - 5y^2 = 1 \Rightarrow [x \;\; y]\begin{bmatrix} 2 & 1.5 \\ 1.5 & -5 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 1$

Why does $\vec{x}^T A \vec{x}$ resemble conjugation?

A is symmetric $\Rightarrow$ A is orthogonally diagonalizable i.e. $\exists P$ (w/ $P^{-1} = P^T$ or $PP^T = I$) such that $P^{-1}AP = D$ (diagonalizable) i.e. $P^TAP = D$ (orthogonally diagonalizable) $\Rightarrow A = PDP^T$. Plugging into $\vec{x}^T A \vec{x} = g$ gives $\vec{x}^T PDP^T \vec{x} = g \Rightarrow (P^T\vec{x})^T D (P^T\vec{x}) = g \Rightarrow \vec{y}^T D \vec{y} = g$

$$(y_1 \; y_2)\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \Rightarrow \lambda_1 y_1^2 + \lambda_2 y_2^2 = g$$

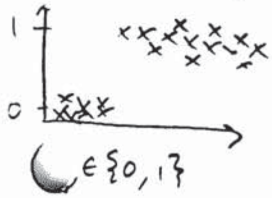
<u>Thm</u>: A is positive definite iff all the eigenvalues of A are positive.

<u>Def</u>: A symmetric matrix $A_{n\times n}$ is called <u>positive semi-definite</u> if $\forall \vec{x} \in \mathbb{R}^n$, $\vec{x}^T A \vec{x} \geq 0$.

<u>Thm</u>: A is positive semi-definite iff all the eigenvalues of A are nonnegative.

In machine learning, we often have $X^T X$ (e.g. the normal equation). Now $X^T X$ is symmetric ($(X^T X)^T = X^T (X^T)^T = X^T X$). Note $X^T X$ is positive semi-definite (for any $\vec{v} \in \mathbb{R}^n$, $\vec{v}^T (X^T X) \vec{v} = (X\vec{v})^T (X\vec{v})$ $= \|X\vec{v}\|^2 \geq 0$

Logistic Regression



$g(z) = \frac{1}{1+e^{-z}}$ is called the <u>logistic</u> or <u>sigmoid function</u>

Choose $h_\theta(x) \in [0,1]$ $\quad h_\theta(x) \triangleq g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$

$P(y=1 | x; \theta) = h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} > 0$, $P(y=0|x;\theta) = 1 - h_\theta(x) > 0$

$P(y|x;\theta) = h_\theta(x)^y (1-h_\theta(x))^{1-y}$ (compact way to write)

Cost function: $L(\theta) = P(\vec{y} | x, \theta) = \prod_i P(y^{(i)} | x^{(i)}; \theta)$

$$= \prod_i h_\theta(x^{(i)})^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{1-y^{(i)}}$$

It is much easier to maximize the log likelihood ratio.

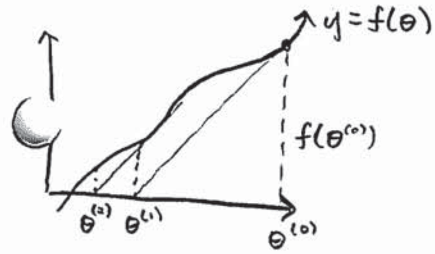
$$\ell(\theta) = \log L(\theta) = \Sigma \left[y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right]$$

How to maximize? Set $\nabla \ell(\theta) = 0$ and use the gradient method.

$\theta_i = \theta + \alpha \nabla_\theta \ell(\theta)$ (use $g'(z) = g(z)(1-g(z))$)

$$\theta_{j+1} := \theta_j + \alpha (y^{(i)} - h(x^{(i)})) x_j^{(i)}$$

Newton's Method



Find θ such that $f(\theta)=0$.

$$f'(\theta^{(0)}) = \frac{f(\theta^{(0)})}{\theta^{(0)}-\theta^{(1)}}$$

$$\Rightarrow \theta^{(0)} - \theta^{(1)} = \frac{f(\theta^{(0)})}{f'(\theta^{(0)})} \Rightarrow \theta^{(1)} = \theta^{(0)} - \frac{f(\theta^{(0)})}{f'(\theta^{(0)})}$$

In general, $\underline{\theta^{(i+1)} = \theta^{(i)} - \frac{\ell'(\theta^{(i)})}{\ell''(\theta^{(i)})}}$

This works well for logistic functions. (quadratic convergence)

For vector version, the generalization is $\theta^{(t+1)} = \theta^{(t)} - H^{(t)}\nabla_\theta^{(t)}\ell$

<u>Recall</u>: $f(\vec{x}) = f(\vec{x}_0) + \nabla f(\vec{x}_0)(\vec{x}-\vec{x}_0) + \frac{1}{2}(\vec{x}-\vec{x}_0)^T H(\vec{x}_0)(\vec{x}-\vec{x}_0) + \ldots$

Hessian

Generalized Linear Model

$P(y|x;\theta)$

$y \in \mathbb{R}$ Gaussian $\longrightarrow$ Least Squares

$y \in \{0,1\}$ Bernoulli $\longrightarrow$ Logistic regression

Exponential Family

<u>Claim</u>: All distributions you know can be put into the exponential family.

$P(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta))$ (often $T(y)=y$)

$T(y) \triangleq$ sufficient statistics

$\eta \triangleq$ natural parameter

For fixed a, b, T, as η changes $\Rightarrow$ get a family of distributions

<u>Ex</u>: Gaussian distribution

$\mathcal{N}(\mu,\sigma^2)$ set $\sigma=1$. $P(y;\mu) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}(y-\mu)^2) = \underbrace{\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}y^2)}_{b(y)}\exp(\mu y - \frac{1}{2}\mu^2)$

η $\quad T(y)=y$ $\quad a(\eta)$

<u>Ex</u>: Bernoulli distribution Ber(ϕ), $p(y=1;\phi)=\phi$, $p(y=0;\phi)=1-\phi$

$$P(y,\phi) = \phi^y(1-\phi)^{1-y}$$

$$= \exp\log(\phi^y(1-\phi)^{1-y}) = \exp[y\log\phi + (1-y)\log(1-\phi)] = \exp[y\log\frac{\phi}{1-\phi} + \log(1-\phi)]$$

Note: $\phi = \frac{1}{1+e^{-\eta}}$ $\quad \underline{b(y)=1}$ $\quad \underline{\eta = \log\frac{\phi}{1-\phi}}$ $\quad \underline{T(y)=y}$ $\quad \underline{a(\eta) = -\log(1-\phi)}$

Ex: Poisson distribution $P(\lambda) = \frac{\lambda^k}{k!} e^{-\lambda}$

$$P(\lambda) = \exp \log \frac{\lambda^k}{k!} e^{-\lambda}$$

Multinomial distribution

If a 6-sided die has 3 faces painted red, 2 white, 1 blue and is rolled 100 times,

find $P(60R, 30W, 10B) = \binom{100}{60}\binom{40}{30}\binom{10}{10}\left(\frac{1}{2}\right)^{60}\left(\frac{1}{3}\right)^{30}\left(\frac{1}{6}\right)^{10}$

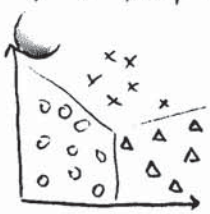
Generally, an experiment with m outcomes with repeated probabilities $p_1, p_2, \ldots, p_m$ is permuted N times independently. Let $X_i = \#$ of times i appears $i = 1, 2, \ldots, m$. Then

$$P(X_1 = k_1, X_2 = k_2, \ldots, X_m = k_m) = \underbrace{\binom{N}{k_1, k_2, \ldots, k_m}}_{N!/(k_1! k_2! \ldots k_m!)} p_1^{k_1} p_2^{k_2} \ldots p_{k_m}^{k_m}$$

$$k_1 + k_2 + \ldots + k_m = N$$

Soft Max Algorithm

$y \in \{1, 2, \ldots, k\}$ (e.g. auto classify emails into groups)



Goal: Write $p(y) \overset{\text{into a one line function}}{\underset{\text{claim}}{=}} \phi_1^{?} \phi_2^{?} \cdots \phi_k^{?}$

Parameters: $\phi_1, \phi_2, \ldots, \phi_k$

$\Sigma \phi_i = 1$

Define $T(1) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{k-1}$, $T(2) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, $\ldots$, $T(k-1) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$, $T(k) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

or more concisely $T(y)_i = 1\{y = i\}$ where $1\{\text{true}\} = 1$, $1\{\text{false}\} = 0$

called the indicate function

$$P(y) = \phi_1^{1\{y=1\}} \phi_2^{1\{y=2\}} \cdots \phi_{k-1}^{1\{y=k-1\}} \phi_k^{1 - \Sigma 1\{y=i\}}$$

$$= \ldots = b(y) \exp(\eta^T T(y) - a(\eta)), \text{ where}$$

$$\eta = \begin{bmatrix} \log(\phi_1/\phi_k) \\ \vdots \\ \log(\phi_{k-1}/\phi_k) \end{bmatrix} \in \mathbb{R}^{k-1} \qquad a(\eta) = -\log(\phi_k) \quad b(y) = 1$$

Let $\eta_i = \log(\phi_i/\phi_k) \Rightarrow \phi_i = \frac{e^{\eta_i}}{1 + \sum_j e^{\eta_j}}$ Let $\eta_i = \theta_i^T x$

Define $h_\theta(x) = E[T(y) \mid x; \theta] = E\left[\begin{pmatrix} 1\{y=1\} \\ \vdots \\ 1\{y=k-1\} \end{pmatrix} \middle| x; \theta\right] = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_{k-1} \end{bmatrix} = \begin{bmatrix} e^{\theta_1^T x} / 1 + \sum_j e^{\theta_j^T x} \\ \vdots \\ e^{\theta_{k-1}^T x} / 1 + \sum_j e^{\theta_j^T x} \end{bmatrix}$

Jacobian of $\vec{F}: \mathbb{R}^n \to \mathbb{R}^m$

$$= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla f_1 \\ \vdots \\ \nabla f_n \end{bmatrix}$$

Hessian of $f: \mathbb{R}^n \to \mathbb{R}$

$$Hf = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Thm: A function f is convex iff $Hf \succ 0$ (positive definite)

Ex: Getting a continuous distribution from a discrete distribution

Binomial $\Rightarrow$ Poisson
(discrete) (continuous)

Binomial: $\binom{n}{k} p^k (1-p)^{n-k}$

$$\frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \quad \text{Consider } n \gg k.$$

$$\Rightarrow \frac{n(n-1)\cdots(n-k+1)}{k!} p^k (1-p)^{n-k}$$

$$\Rightarrow \frac{n^k}{k!} p^k (1-p)^{n-k} \qquad \text{Let } \lambda = np$$

$$\Rightarrow \frac{\lambda^k}{k!} \left(1 - \frac{\lambda}{n}\right)^n \xrightarrow{n \to \infty} \frac{\lambda^k}{k!} e^{-\lambda} : \text{Poisson}$$

$$\text{Corr}[X,Y] \triangleq \frac{\text{cov}[X,Y]}{\sqrt{\text{var}[X]\text{var}[Y]}}$$

Notation for probabilities:

$$P(A,B) = P(A \cap B) = P(A \wedge B)$$
$$P(A \cup B) = P(A \vee B)$$
$$P(X_{1:D}) = P(X_1, X_2, \ldots, X_D)$$

$$E[X] = \int_{-\infty}^{\infty} x f(x)\, dx$$

$f(x)$ is the pdf of X

The moment generating function is defined as $\Phi(t) = E[e^{tX}]$

The covariance between two variables X and Y measures the degree to which X and Y are linearly related.

$$\text{cov}[X,Y] \triangleq E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y]$$

$$\mathcal{N}(x \mid \mu, \Sigma) \triangleq \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

where $\mu = E[x] \in \mathbb{R}^D$ and $\Sigma = \text{cov}[x]$

The inverse of the covariance matrix $\Lambda = \Sigma^{-1}$ is the <u>precision</u> or <u>concentration matrix</u>.

Independence (denoted $X \perp Y$)

$$X \perp Y \iff p(X,Y) = p(X)p(Y)$$

Conditional Independence

$$X \perp Y | Z \iff p(X,Y|Z) = p(X|Z)\,p(Y|Z)$$

<u>Thm:</u> $X \perp Y | Z$ iff $\exists g, h$ such that $p(x,y|z) = g(x,z)\,h(y,z)$, $\forall x,y,z$ s.t. $p(z) > 0$

General Linear Models

measured dependent variable

residual/error term

$$Y = \underbrace{\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \dots + \theta_n X_n}_{X^T\theta} + \varepsilon$$

$X^T\theta$ ← estimation of Y, may be inaccurate

The θ's are regression weights, or parameters of the linear model. Each assess the feature/factor X_i's contribution to predict the value of the dependent variable Y. Note $X_0 = 1$.

<u>Principal component analysis</u> (PCA) is a procedure to find an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables.

Procedure to find principle directions.

1) Find covariance matrix
   - Can standardize data first by subtracting all x_i-values by μ_{x_i} (geometrically, move the axes to the data center)
2) Find eigenvalues and eigenvectors of the covariance matrix
3) Order eigenvalues from largest to smallest. The eigenvector corresponding to the largest eigenvalue is called the 1st principle axis, etc.
4) Form the rotation matrix using corresponding eigenvectors.

If two eigenvectors have the same eigenvalue, they might not be orthogonal. Using Gram-Schmidt will give you an orthonormal basis.

<u>Thm</u>: The eigenvectors of a symmetric matrix A corresponding to different eigenvalues are orthogonal to each other.

<u>Thm</u>: For any matrix X, the nonzero eigenvalues of XX^T and X^TX are the same.

Singular Value Decomposition (SVD)

It is a generalization of the notion of eigenvectors from square matrices to any matrix.

$$X^TX = VDV^T \quad \text{where } V = \underset{\text{eigenvectors}}{\text{evec}}(X^TX)$$

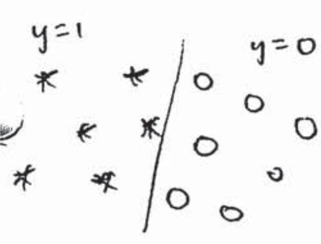
$$XX^T = UD_1U^T \quad \text{where } U = \text{evec}(XX^T)$$

$$\underbrace{X}_{N\times D} = \underbrace{U}_{N\times N}\ \underbrace{S}_{N\times D}\ \underbrace{V^T}_{D\times D}$$

$$\underbrace{X}_{N\times D} \simeq \underbrace{U_L}_{N\times L}\ \underbrace{S_L}_{L\times L}\ \underbrace{V_L^T}_{L\times D}$$

You can use SVD to do data compression. Set singular values less than an error threshold equal to 0. For example, if you want less than 5% error, find the singular value σ_L, say $L=30$, close to 5% and set $\sigma_L = 0$ for $L > 30$.

Generative Learning Algorithm (GLA)



We model $p(x|y)$ and $p(y)$ to predict $p(y|x)$. We do this with Bayes' Rule

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad \text{where } P(x) = P(x|y=1)P(y=1) + P(x|y=0)P(y=0)$$

In order to make a prediction, we want

$$\arg\max_y P(y|x) = \arg\max_y \left[\frac{P(x|y)P(y)}{P(x)}\right] = \arg\max_y \left[P(x|y)P(y)\right]$$

<u>Ex:</u> Gaussian Discriminant Analysis (GDA)

$y=1$ cancer

$y=0$ no cancer

$$\Rightarrow \quad \underset{\text{random variable}}{y} \overset{\text{obeys}}{\sim} \overset{\text{distribution}}{\text{Bernoulli}(\phi)}$$

$$x|y=0 \sim \mathcal{N}(\mu_0, \Sigma)$$

$$x|y=1 \sim \mathcal{N}(\mu_1, \Sigma)$$

$$\Sigma = \frac{1}{n} XX^T$$

That is, $p(y) = \phi^y (1-\phi)^y$

$$P(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right)$$

$$P(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)$$

The parameters of our model are ϕ, μ_0, μ_1, Σ. We have different mean values but the same covariance matrix.

The log-likelihood of the data is given by

$$\arg\max \, \ell(\phi, \mu_0, \mu_1, \Sigma) = \arg\max \log \prod_{i=1}^{m} P(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma)$$

$$= \arg\max \log \prod_{i=1}^{m} p(x^{(i)}|y^{(i)}, \underbrace{\mu_0, \mu_1, \Sigma}_{\text{parameters of } x}) \, p(y^{(i)}; \underbrace{\phi}_{\text{parameters of } y})$$

By maximizing ℓ w.r.t. to the parameters, we find the maximum likelihood estimation of the parameters to be

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)}=1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=0\}\, x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}} \qquad \mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=1\}\, x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$$

<u>Thm:</u> For a $D \times N$ matrix X, the set of nonzero eigenvalues of $X^T X$ and XX^T are the same.

<u>Proof:</u> Idea: Show the two sets of eigenvalues are the same by containment in both directions.

nonzero eigenvalues

Let $\lambda \in \text{eval}(XX^T) \Rightarrow (XX^T)\vec{v} = \lambda\vec{v} \Rightarrow X^T X(X^T\vec{v}) = \lambda(X^T\vec{v})$

$\Rightarrow \lambda \in \text{eval}(X^T X)$ w/ eigenvector $X^T\vec{v} \Rightarrow \text{eval}(XX^T) \subseteq \text{eval}(X^T X)$.

Similarly, $\text{eval}(X^T X) \subseteq \text{eval}(XX^T)$. Thus, $\text{eval}(XX^T) = \text{eval}(X^T X)$.

Schur Complement

$$M = \begin{array}{cc} & \begin{array}{cc} p & q \end{array} \\ & \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{array}{c} p \\ q \end{array} \end{array} \qquad \begin{array}{cc} A_{p\times p} & B_{p\times q} \\ C_{q\times p} & D_{q\times q} \end{array}$$

The Schur complement of the block D of the matrix M is the $p \times p$ matrix

$$M/D \triangleq A - BD^{-1}C$$

If A is invertible, then the Schur complement of the block A of the matrix M is

$$M/A \triangleq D - CA^{-1}B$$

If $M = \left[\begin{array}{c|c} A & 0 \\ \hline 0 & D \end{array}\right]$, we say D is the complement of A & vice versa for A.

What if $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$?

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \xrightarrow[+\text{ col } 1]{\text{col } 2 \cdot (-D^{-1}C)} \begin{bmatrix} A - BD^{-1}C & B \\ 0 & D \end{bmatrix} \xrightarrow[+\text{row } 1]{\text{row } 2 \cdot (-BD^{-1})} \begin{bmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{bmatrix}$$

<u>Recall:</u>

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \xrightarrow{\text{row manip.}} \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix} \xrightarrow{\text{col. manip.}} \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}$$

$$E_1 = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} = E_2$$

is done by row operations but we can represent this as matrix multiplication.

$$E_1 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} E_2 = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}$$

Matrices representing row operations are multiplied on the left and matrices representing column operations are multiplied on the right.

We perform a similar operation on the Schur complement. We start with

$$\begin{bmatrix} I_{p\times p} & 0 \\ 0 & I_{q\times q} \end{bmatrix} \xrightarrow[+\text{ col } 1]{\text{col } 2 \cdot (-D^{-1}C)} \begin{bmatrix} I_{p\times p} & 0 \\ -D^{-1}C & I_{q\times q} \end{bmatrix}$$

$$\begin{bmatrix} I_{p\times p} & 0 \\ 0 & I_{q\times q} \end{bmatrix} \xrightarrow[+\text{ row } 1]{\text{row } 2 \cdot (-BD^{-1})} \begin{bmatrix} I_{p\times p} & -BD^{-1} \\ 0 & I_{q\times q} \end{bmatrix}$$

this gives

$$\begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} = \begin{bmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{bmatrix}$$

Applications of Schur's Complement to Probability and Statistics

Suppose the random column vectors $X \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ form $\begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^{n+m}$ has a multivariate normal distribution whose covariance matrix (which is symmetric and positive definite) is $\Sigma = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$ where $A \in \mathbb{R}^{n\times n}$ is the covariance matrix of X, $C \in \mathbb{R}^{m\times m}$ is that of y, and $B \in \mathbb{R}^{n\times m}$ is that of X and y. Then the <u>conditional covariance</u> of X given y is the Schur complement of C in Σ.

$$\text{cov}(X|y) = A - BC^{-1}B^T$$

$$E[X|y] = E[X] - BC^{-1}(y - E[y])$$

Naive Bayes — assume an independent feature model

Given data $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$ where $x^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)}) \in \mathbb{R}^D$, $y^{(i)} \in Y = \{1, \dots, n\}$. Assume a family of distributions P_θ such that for $X \in \mathbb{R}^d$, $y \in Y$, $P_\theta(x,y) = P_\theta(x|y)P_\theta(y) \underset{\uparrow\text{ key assumption}}{=} P_\theta(x_1|y)P_\theta(x_2|y)\dots P_\theta(x_d|y)P_\theta(y)$, i.e. assume $\{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\} \sim P_\theta$ is iid for some Θ.

Goal: For new $x \in \mathbb{R}^d$, predict its y, or compute

$$\tilde{y} \in \arg\max_y P_{\tilde{\Theta}}(y|x) = \arg\max_y \frac{P_{\tilde{\Theta}}(x|y)P_{\tilde{\Theta}}(y)}{P_{\tilde{\Theta}}(x)}$$

doesn't depend on y

$$= \arg\max_y P_{\tilde{\Theta}}(x|y)P_{\tilde{\Theta}}(y)$$

using naive assumption

$$= \arg\max_y P_{\tilde{\Theta}}(x_1|y)P_{\tilde{\Theta}}(x_2|y)\dots P_{\tilde{\Theta}}(x_d|y)P_{\tilde{\Theta}}(y)$$

Method of Binning

Look at entire data set.

$$\max_{i,j}\{x_j^{(i)}\} = 500 \qquad \min_{i,j}\{x_j^{(i)}\} = 0$$

$$\begin{bmatrix} 0 \\ 0.2 \\ 10 \\ \hline 10.11 \\ \vdots \\ 20 \\ \hline \vdots \\ 500 \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{level 1} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{level 2} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{level 50} \end{array}$$

Least Absolute Shrinkage and Selection Operator (LASSO)

- Regression method that penalizes the absolute size of the regression coefficients

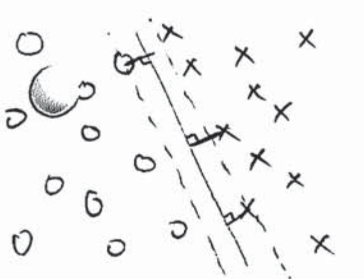
$$\hat{\beta}^{lasso} = \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$= \arg\min_{\beta \in \mathbb{R}^p} \underbrace{\|y - X\beta\|_2^2}_{\text{loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{penalty}}$$

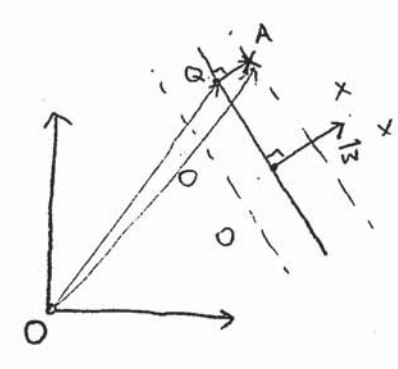
Compare this with $\hat{\beta}^{ridge} = \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$

Support Vector Machine (SVM)

We want to maximize the margin to increase the confidence of your prediction.



Hyperplane in $\mathbb{R}^n$ characterized by $W^T\vec{x} + \vec{b} = 0$



Our trick is to find the point Q on the hyperplane π s.t. it is written in a way involving $\gamma^{(i)}$. Then $Q \in \pi$ and satisfies the plane equation $\Rightarrow \gamma_i$ will be in

$$\overrightarrow{OQ} + \overrightarrow{QA} = \overrightarrow{OA} \qquad Q \in \pi \qquad W^T Q + b = 0$$

$$Q + \gamma^{(i)} W = X^{(i)} \Rightarrow Q = X^{(i)} - \gamma^{(i)} W \qquad \Rightarrow W^T (X^{(i)} - \gamma^{(i)} W) + b = 0$$

$$\Rightarrow W^T X^{(i)} - \gamma^{(i)} (W^T W) + b = 0$$

Kernel Methods

In general, data may not be separable by a hyperplane.

We can now separate it with a hyperplane.

How to construct the kernel?

can we find this clever map

For $X, Z \in \mathbb{R}^n$, $\langle X, Z \rangle = X^T Z$, $K(X,Z) = (X^T Z)^2 \overset{?}{=} [\varphi(X)]^T \varphi(Z)$

We want to find φ.

$$K(x,z) = (x^T z)^2 = (x \cdot z)^2 = (x \cdot z)(x \cdot z) = \left(\sum_{i=1}^n x_i z_i\right)\left(\sum_{j=1}^n x_j z_j\right)$$

$$= \sum_{i=1}^n \sum_{j=1}^n x_i z_i x_j z_j = \sum_{i=1}^n \sum_{j=1}^n (x_i x_j)(z_i z_j) = \varphi(x)^T \varphi(z)$$

Say $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$.

$$x^T z = x \cdot z = x_1 z_1 + x_2 z_2 \qquad K(x^T z) = (x_1 z_1 + x_2 z_2)^2$$

$$= x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2$$

$$= \begin{bmatrix} x_1^2 \\ x_1 x_2 \sqrt{2} \\ x_2^2 \end{bmatrix} \cdot \begin{bmatrix} z_1^2 \\ z_1 z_2 \sqrt{2} \\ z_2^2 \end{bmatrix} = \varphi(x)^T \varphi(z)$$

Define

$$\varphi(x) = \varphi\underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\in \mathbb{R}^2} = \underbrace{\begin{pmatrix} x_1^2 \\ x_1 x_2 \sqrt{2} \\ x_2^2 \end{pmatrix}}_{\in \mathbb{R}^3}$$

Therefore $K(x,z) = (x^T z)^2$ is a kernel.

A clustering problem is an unsupervised learning problem.

k-means Clustering Algorithm

1. Initialize cluster centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence

for every i, set $\quad c^{(i)} := \arg\min_j \|x^{(i)} - \mu_j\|^2$

for each j, set $\quad \mu_j := \dfrac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}$

The algorithm is guaranteed to converge but it might converge to a local optimization point instead of a global one.

Stochastic Gradient Descent (SGD)

- repeatedly runs through the training set, and each time it encounters a training example, it updates the parameters
- may never "converge" to the unique minimum
- faster than batch gradient descent (BGD)
- gets θ "close" to the minimum much faster than BGD

A probability mass/density function $P(x)$ is just a special kind of function

① $p(x) \geq 0$

② $\int p(x)\,dx = 1 \quad (\Sigma p(x_i) = 1)$

Consider $P(\theta) = \sum_{i=1}^{k} \phi_i P_i(\theta)$. This is also a probability density function, if $\phi_i \geq 0$ and $\Sigma \phi_i = 1$. Such a sum is called a <u>convex sum</u>.

Multivariate Gaussian Mixture Model

$$p(\theta) = \sum_{i=1}^{k} \phi_i \mathcal{N}(\mu_i, \Sigma_i)$$

$$\phi_j \geq 0, \quad \sum \phi_j = 1$$

Like k-means, GMM clusters have centers. They have probability distributions that indicate the probability a point belongs to the cluster. This model is more complex than k-means since distance from the center can matter more in one direction than another.

EM (Expectation - Maximization) Algorithm

Given a training set $\{x^{(1)}, \ldots, x^{(m)}\}$, we want to model the data by specifying a joint distribution.

$$p(x^{(i)}, z^{(i)}) = p(x^{(i)} | z^{(i)})\, p(z^{(i)})$$

where $z^{(i)} \sim \text{Multinomial}(\phi)$, $\phi_j \geq 0$, $\sum \phi_j = 1$, and the parameter ϕ_j gives $p(z^{(i)} = j$ and $x^{(i)} | z^{(i)} = j) \sim \mathcal{N}(\mu_j, \Sigma_j)$

Steps for algorithm:

① Guess values for $z^{(i)}$'s - Expectation

② Maximize the likelihood to determine the parameters - Maximization

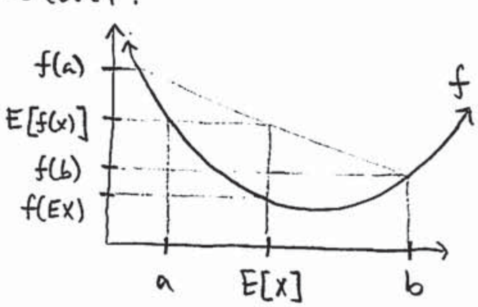
③ Repeat.

Jensen's Inequality

<u>Thm:</u> Let f be a convex function, and let X be a random variable. Then,

$$E[f(X)] \geq f(EX)$$

If f is stricty convex, $E[f(X)] = f(EX)$ iff $X = E[X]$ with probability 1, i.e. if X is a constant.



Maximum a posteriori (MAP)

Setup: Given data $D = (x_1, \dots, x_n)$, assume a joint distribution

random variable

$$P(D, \theta) = P(D|\theta)P(\theta)$$

prior distribution on θ

Goal: Choose a good value of θ for D. We choose

$$\theta_{MAP} = \arg\max_{\theta} P(\theta|D) \quad = \frac{P(D|\theta)\,P(\theta)}{P(D)}$$

maximize a posterior distribution θ

$P(D)$ doesn't involve θ

$$= \arg\max_{\theta} P(D|\theta)P(\theta)$$

$$= \arg\max_{\theta} \left(\log P(D|\theta) + \log P(\theta)\right)$$

We then set the derivative wrt θ to 0.

$$0 = \frac{\partial}{\partial\theta}\left(\log P(D|\theta) + \log P(\theta)\right)$$

$\log p(\theta) = \log \frac{1}{\sqrt{2\pi}} - \frac{1}{2}(\theta - \mu)^2$ if we assume $p(\theta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\theta-\mu)^2}$

$$\Rightarrow 0 = \frac{1}{\sigma^2}\left(\sum_{i=1}^{n} x_i - n\theta\right) + \mu - \theta$$

$$\Rightarrow \sum_{i=1}^{n} \frac{x_i}{\sigma^2} + \mu - \left(\frac{n}{\sigma^2} + 1\right)\theta = 0$$

$\frac{\partial}{\partial\theta^2} = -\left(\frac{n}{\sigma^2}+1\right) < 0$ so the critical point is a max

$$\Rightarrow \theta = \frac{\frac{1}{\sigma^2}\sum x_i + \mu}{\frac{n}{\sigma^2} + 1}$$

$$\Rightarrow \theta_{MAP} = \frac{n}{n+\sigma^2}\bar{x} + \frac{\sigma^2}{n+\sigma^2}\mu$$

$$\xrightarrow{n \to \infty} \theta_{MAP} = 1\bar{x} + 0\mu = \bar{x} = \theta_{MLE}$$

So, we can think of MLE as a special case of MAP.

parameterized model → MLE (special case of MAP $(n \to \infty)$)

parameterized model → Bayesian learning { Naive Bayes; MAP → optimize your estimation/classification results and minimize the error *if* you know the exact probability distributions involved }

In real life, we don't know the probability distributions.

Parameterized estimation: Assume a model (ex. GMM) and use EM algorithm.

Nonparameterized estimation: Use KNN.

Kernel PCA

It is restricted in that it computes not the principal components themselves, but the projections of our data onto those components.

Recall: The inner product is the key metric in Euclidean space.

① $\|\vec{v}\| = \sqrt{\langle \vec{v}, \vec{v} \rangle}$

② $\cos\theta = \dfrac{\langle \vec{v}, \vec{w} \rangle}{\|\vec{v}\|\,\|\vec{w}\|} = \dfrac{\langle \vec{v}, \vec{w} \rangle}{\sqrt{\langle \vec{v}, \vec{v} \rangle}\sqrt{\langle \vec{w}, \vec{w} \rangle}}$

③ $\|\vec{v} - \vec{w}\| = \sqrt{\langle \vec{v}-\vec{w}, \vec{v}-\vec{w} \rangle}$

④ $\vec{v} \perp \vec{w} \iff \langle \vec{v}, \vec{w} \rangle = 0$

In a matrix representation,

$$\langle \vec{v}, \vec{w} \rangle_A = \vec{v}^T A \vec{w}$$

as long as A is a positive definite matrix.

If $A = I$, the inner product is the dot product.

An inner product is a symmetric, positive definite, bilinear form.

Let $f(\vec{x}) = \frac{1}{2}\vec{x}^T A \vec{x} - \vec{x}^T \vec{b}$ so $f : \mathbb{R}^n \to \mathbb{R}$. How do we take its derivative?

Say $\vec{x} = \binom{x_1}{x_2}$.

$$f(x_1, x_2) = \frac{1}{2}(x_1\ x_2)\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - (x_1\ x_2)\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$= \frac{1}{2}\left(a_{11}x_1^2 + a_{21}x_2x_1 + a_{12}x_1x_2 + a_{22}x_2^2\right) - (b_1x_1 + b_2x_2)$$

$$\nabla f = \frac{A + A^T}{2}\vec{x} + \vec{b}$$

$$\frac{\partial f}{\partial x_1} = \frac{1}{2}(2a_{11}x_1 + a_{21}x_2 + a_{12}x_2) - b_1$$

$$\frac{\partial f}{\partial x_2} = \frac{1}{2}(a_{21}x_1 + a_{12}x_1 + 2a_{22}x_2) - b_2$$

$$\Rightarrow \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} a_{11} & \frac{a_{12}+a_{21}}{2} \\ \frac{a_{21}+a_{12}}{2} & a_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

So we have $\nabla f = \text{sym}(A)\vec{x} + \vec{b}$ where

$$A = \underbrace{\frac{A+A^T}{2}}_{\text{sym}(A)} + \underbrace{\frac{A-A^T}{2}}_{\text{skew}(A)}$$

Here is a second method. We start with $f(\vec{x}) = \frac{1}{2}\vec{x}^T A\vec{x} - \vec{x}^T\vec{b}$

$$\nabla f(\vec{x}) = \nabla(\tfrac{1}{2}x^T A x) + \nabla(-x^T b)$$

For $\nabla(x^T b)$,

$$\nabla(\vec{x}^T\vec{b}) = \nabla(b_1 x_1 + \dots + b_n x_n) = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \vec{b}$$

$$\nabla(\vec{x}\cdot\vec{b}) = D\vec{x}\cdot\vec{b} + \vec{x}\cdot\nabla\vec{b} = I\vec{b} + \vec{x}\cdot\vec{0} = \vec{b}$$

For $\nabla(x^T A x)$,

$$\begin{aligned} \nabla(\vec{x}^T A\vec{x}) &= \nabla(\vec{x}\cdot(A\vec{x})) \\ &= D\vec{x}\cdot A\vec{x} + [D(A\vec{x})]^T\vec{x} \\ &= I(A\vec{x}) + A^T\vec{x} \end{aligned}$$

Thus,

$$\begin{aligned} \nabla f(\vec{x}) &= \nabla(\tfrac{1}{2}x^T A x) + \nabla(-x^T b) \\ &= \tfrac{1}{2}\nabla(x^T A x) - \nabla(x^T b) \\ &= \tfrac{1}{2}(A + A^T)\vec{x} - \vec{b} \\ &= \text{sym}(A)\vec{x} - \vec{b} \end{aligned}$$

LU Decomposition

If $A = LU$, $Ax = b$ can be solved quickly. $Ax = b$ can be written as $LUx = b$. Let $y = Ux$. Then $Ly = b$ and we solve for y easily. Then we solve $y = Ux$ easily.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix}\begin{bmatrix} l_{11} & l_{12} \\ 0 & l_{22} \end{bmatrix}$$

$a_{11} = l_{11}^2 \Rightarrow l_{11} = \sqrt{a_{11}}$

$a_{12} = l_{11} l_{21} \Rightarrow l_{21} = \frac{a_{12}}{l_{11}} = \frac{a_{12}}{\sqrt{a_{11}}}$

$\vdots$

Can we always decompose a symmetric matrix into LU where $U = L^T$?

Say A is symmetric and $A \succeq 0$. Then $\exists P$ st $P^TP = I$ and $P^TAP = D$.

Thus,

$$
\begin{aligned}
A &= PDP^T \qquad \leftarrow \lambda_i \geq 0 \\
&= P \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 \ddots & \\ 0 & & \lambda_n \end{pmatrix} P^T \\
&= P \begin{pmatrix} \sqrt{\lambda_1} & & \\ & \ddots & \\ & & \sqrt{\lambda_n} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & & \\ & \ddots & \\ & & \sqrt{\lambda_n} \end{pmatrix} P^T \\
&= P\sqrt{D}\sqrt{D}P^T \\
&= P\sqrt{D}P^TP\sqrt{D}P^T \\
&= (P\sqrt{D}P^T)^2 = Q^2
\end{aligned}
$$

This is the Pfaffian decomposition.

Applications of Cholesky Decomposition

- use to solve $Ax = b$ when A is real, symmetric, and positive definite
- in regression analysis, use to estimate the parameter if X^TX is positive definite
- use in kernel principal component analysis

Overview

Linear Algebra

- spectral decomposition
- singular value decomposition (SVD)
- Cholesky decomposition
- LU decomposition
- Schur complement
- QR decomposition
- Jordan decomposition

Geometry

- least squares
- Support vector machine (SVM)
- L_p norms
- nonparametric density estimation

Consider $y^{(i)} = \theta_0 + \theta_1 x_1^{(i)} + \dots + \theta_n x_n^{(i)} = [1 \; x_1^{(i)} \dots x_n^{(i)}][\theta_0 \; \theta_1 \dots \theta_n]^T$.

$$\underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(i)} \\ \vdots \\ y^{(N)} \end{bmatrix}}_{\vec{y}} = \underbrace{\begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_n^{(1)} \\ \vdots & \vdots & \ddots & \\ 1 & x_1^{(i)} & \cdots & x_n^{(i)} \\ \vdots & \vdots & & \ddots \\ 1 & x_1^{(N)} & \cdots & x_n^{(N)} \end{bmatrix}}_{X_{N\times n}} \underbrace{\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}}_{\theta}$$

$y = X\theta$ ← overdetermined system

$n << N$

We multiply to get $X^T\vec{y} = X^TX\theta$ or $\vec{b} = A\theta$ where $\vec{b} = X^T\vec{y}$ and $A = X^TX$. Geometrically, $X_i \perp \varepsilon \Rightarrow X_i^T\varepsilon = 0$ so $X^T(\vec{y} - X\theta) = 0$. We know X^TX is positive semidefinite so the eigenvalues are all ≥ 0. Say $\text{rank}(X^TX) = n$.

LU decomposition

<u>Ex:</u> $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ is square and nonsingular ($\det(A) \neq 0$). Recall how we find A^{-1}.

$$(A|I) \xrightarrow{\text{elementary operations}} (I|A^{-1})$$

Here, we make A upper triangular.

$$\underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}_{A} \Bigg| \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{I} \xrightarrow{-3\cdot R_1 + R_2} \underbrace{\begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}}_{U} \Bigg| \underbrace{\begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix}}_{E_1}$$

This is expressed with matrices by

$E_1 A = U$ or $A = E_1^{-1} U$

Note that the inverse of an elementary matrix is still an elementary matrix. So we get $E_1^{-1} = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}$ by reversing what we did to get E_1. We continue elementary row operations on A until we get an LU decomposition.

In general,

$$A_{n\times n} \xrightarrow{O^1_{E_1}} A' \xrightarrow{O^2_{E_2}} \cdots \xrightarrow{O^k_{E_k}} \underbrace{\begin{bmatrix} \diagdown & \\ 0 & \diagdown \end{bmatrix}}_{U}$$

$$\Longrightarrow E_1 A = A', \; E_2 A' = A^2, \ldots \Longrightarrow E_k E_{k-1} \cdots E_2 E_1 A = U$$

$$\Longrightarrow A = (E_k \cdots E_1)^{-1} U = E_1^{-1} E_2^{-1} \cdots E_k^{-1} U \Longrightarrow A = LU$$

Note that the product of lower triangular matrices is lower triangular. Each E_i represents an elementary row operation.

QR Decomposition

If A is an $m\times n$ matrix with linearly independent columns, then A can be decomposed as $A = QR$ where Q is an $m \times n$ matrix whose columns form an orthogonal basis for the column space of A and R is a nonsingular upper triangular matrix.

<u>Proof:</u> Suppose $A = [u_1 | u_2 | \ldots | u_n]$ and $\text{rank}(A) = n$. We apply the Gram-Schmidt process to $\{u_1, \ldots, u_n\}$ to get $\{q_1, \ldots, q_n\}$. Let $Q = [q_1 \cdots q_n]$ so Q is an $m \times n$ matrix whose columns form an orthonormal basis for the column space of A. Now,

v_i are the orthogonal vectors

$$A = [q_1 \cdots q_n] \begin{bmatrix} \|v_1\| & q_1\cdot u_2 & q_1 \cdot u_3 & \cdots & q_1 \cdot u_n \\ & \|v_2\| & & & \vdots \\ & & \ddots & & \\ & 0 & & \ddots & q_{n-1}\cdot u_n \\ & & & & \|v_n\| \end{bmatrix} = QR$$

We use QR decomposition to solve $X^TX\Theta = X^Ty$. Note that $\text{rank}(X) = \text{row rank}(X) = \text{col rank}(X)$ and is almost always n. Now, $X = QR$ where $Q^TQ = I$. Wlog assume the columns are linearly independent. Then,

$$(QR)^T QR\Theta = (QR)^T y$$

$$\Rightarrow \quad R^TQ^TQR\Theta = R^TQ^Ty$$

$$\Rightarrow \quad R^TR\Theta = R^TQ^Ty$$

$$\Rightarrow \quad R\Theta = Q^Ty$$

We know R^T is invertible since $n = \text{rank}(X) = \text{rank}(R)$ and R is an $n \times n$ matrix. Now, R is upper triangular and Q^Ty is known so we can easily solve for Θ by back substitution.

Bayesian Inference

- we assume a prior distribution $P(\Theta)$
- Bayesian procedure: min, exp, loss, optimization
- objective vs subjective: belief-based prior

Pros:

- directly answer questions you are interested in
- avoid pathologies
- avoid overfitting
- automatically selects model (prefers simpler model)

Cons:

- must assume a prior (sometimes no such prior exists)
- exact computation often incontractable (often involves integrals that can't be solved analytically)
  - computers will take too long so clever approximation methods are needed

The Bayesian procedure is admissible meaning no other method will be uniformly better if the prior is accurate.

Since the prior is so important, what are some good priors?

- non-informative (e.g. $P(\theta) = 1$)
- conjugate priors (often can give closed formula solutions)

If f is a PDF or PMF and $f \propto g$ (proportional) then g uniquely determines f and

$$f(x) = \frac{g(x)}{\int g(x)\,dx}$$

Bayes' Theorem

Likelihood
How probable is the evidence given that our hypothesis is true?

Prior
How probable was our hypothesis before observing the evidence?

$$P(H|e) = \frac{P(e|H)\,P(H)}{P(e)}$$

Posterior
How probable is our hypothesis given the observed evidence? (Not directly computable)

Marginal
How probable is the new evidence under all possible hypothesis?

$$P(e) = \sum P(e_i|H_i)\,P(H_i)$$

Conjugate Priors

<u>Def:</u> A family $\mathcal{F}$ of prior distributions $p(\theta)$ is conjugate to a likelihood $p(D|\theta)$ if the posterior $P(\theta|D)$ is in $\mathcal{F}$.

<u>Ex:</u> - Beta distribution is conjugate to Bernoulli distribution
- Gaussian - - - - - - - - - - - - - Gaussian - - - - -
- any exponential family distribution has a conjugate prior

So what?

Ex: Beta — Bernoulli : $X \sim$ Bernoulli, $X \sim Ber(\theta)$

$x_1, x_2, \ldots, x_n \sim Ber(\theta)$ $\qquad\qquad \theta \sim Beta(a,b)$

$$P(x=1|\theta) = \theta = \theta^{1\{x=1\}}(1-\theta)^{1\{x=0\}}$$

$$P(\theta) = \frac{\theta^{a-1}(1-\theta)^{b-1}}{B(a,b)} \propto \theta^{a-1}(1-\theta)^{b-1}$$

Now let's see Beta is conjugate to Bernoulli.

$$P(\theta|D) \underset{\theta}{\propto} P(D|\theta)\,P(\theta) = P(\theta)\prod_{i=1}^{n} P(x_i|\theta)$$

$$\propto \theta^{a-1}(1-\theta)^{b-1}\,\theta^{\underbrace{\Sigma 1\{x_i=1\}}_{=n_1}}(1-\theta)^{\underbrace{\Sigma 1\{x_i=0\}}_{=n_0}}$$

(Note $a=b=1 \Rightarrow$ Uniform)

$$= \theta^{a+n_1-1}(1-\theta)^{b+n_0-1} \propto Beta(\theta|a+n_1, b+n_0)$$

$\Rightarrow$ Beta is conjugate to Bernoulli

In general, if $\theta \sim Beta(a,b)$ then $E(\theta) = \frac{a}{a+b}$, $\sigma^2(\theta) = \frac{ab}{(a+b)^2(a+b+2)}$, mode $= \frac{a-1}{a+1-2}$. Thus for our posterior distribution

$$\theta|D \sim Beta(a+n_1, b+n_0)$$

$$\Rightarrow E[\theta|D] = \frac{a+n_1}{a+b+n}$$

where $n = n_0 + n_1$

$$\sigma^2[\theta|D] = (HW)$$

$$\text{mode}(\theta|D) = \frac{a+n_1-1}{a+b+n-2}$$

If we use θ_{MLE},

$$\theta_{MLE} = \text{empirical probability} = \frac{n_1}{n}$$

$$\left(\frac{n_0}{n}, \frac{n_1}{n}\right)$$

If we use Θ_{MAP},

$$\Theta_{MAP} = \text{max of the mode} = \frac{a+n_1-1}{a+b+n-2}$$

Claim:

$$\underbrace{\frac{a+n_1}{a+b+n}}_{E[\theta|D]} = \alpha \underbrace{\frac{a}{a+b}}_{E[\theta]} + \underbrace{(1-\alpha)\frac{n_1}{n}}_{E[\theta_{MLE}]}$$

it's a convex combination

this is why $E(\theta)$ being accurate is very important

We check this with $\frac{a+n_1}{a+b+n} = \left(\frac{a+b}{a+b+n}\right)\frac{a}{a+b} + \left(\frac{n}{a+b+n}\right)\frac{n_1}{n}$

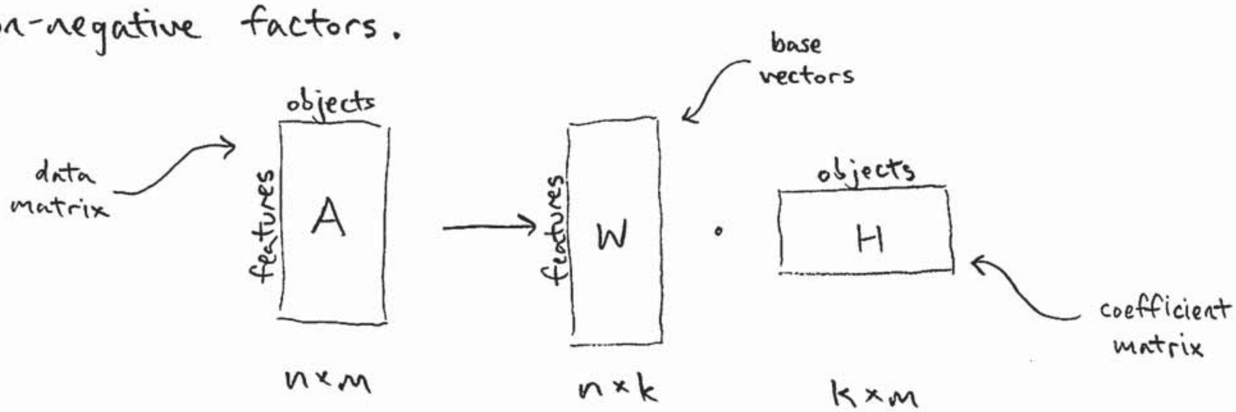
We claim we can also calculate the predictive probability in closed form. This is called "predictive distribution".

Topic Modeling

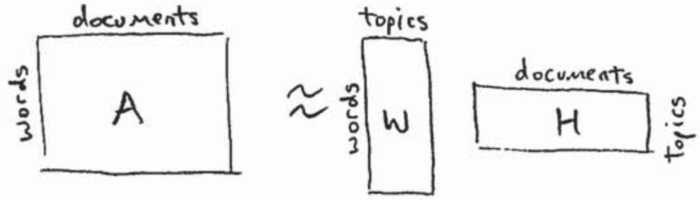
The goal is to discover hidden thematic structure in a corpus of text. This is an unsupervised approach, no prior annotation is required. The output of topic modeling is a set of k topics. Each topic has a descriptor, based on highest-ranked terms for the topic, and membership weights for all documents relative to the topic.

Non-negative Matrix Factorization

Given a non-negative data matrix A, A is approximated by WH where each element of W and H are nonnegative. W and H are called non-negative factors.



In topic modeling, we have the following topic representation:



We want to minimize the error between A and the approximation WH. We can use EM optimization.

$$\frac{1}{2}\|A - WH\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(A_{ij} - (WH)_{ij}\right)^2$$

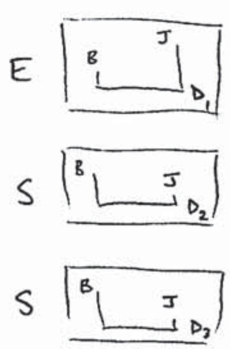
Where $\|X\|_F$ is the <u>Frobenius norm</u>. We have

$$\|A\|^2 = tr(A^T A) = tr(AA^T) = \|v_A\|^2$$

$$\|A-B\|_F^2 = tr((A-B)^T(A-B)) = \|v_A - v_B\|^2 \qquad v_A, v_B \in \mathbb{R}^{n\times m}$$

Latent Dirichlet Allocation (LDA)

Suppose we have 3 documents with 2 topics: S(ports) and E(ntertainment).



Documents

There are words in each document.

words = ["Bryant", "Jordan", ...]

$\updownarrow$ S (Bryant) $\qquad$ $\updownarrow$ E (Jordan)

In general there are many words in a document.

L = # of words in the dictionary (or $\bigcup_{i=1}^{\#doc.} D_i$ for efficiency)

β_{E_i} = probability that the i^{th} word occurs

$$\begin{cases} \beta_{E_i} \geq 0 \\ \sum_{i=1}^{L} \beta_{E_i} = 1 \end{cases}$$

Also in general there are many topics $T_1, \dots, T_j, \dots, T_k$

where k is a hyperparameter. We have the word distribution.

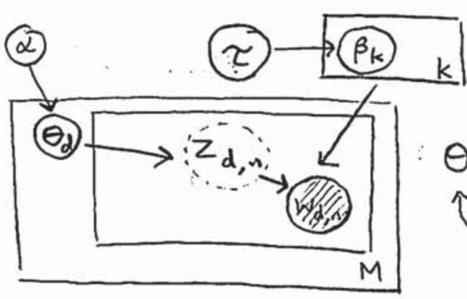
$$\begin{array}{l} [\beta_{11}, \dots, \beta_{1L}] \\ \quad\vdots \\ [\beta_{j1}, \dots, \beta_{jL}] \\ \quad\vdots \\ [\beta_{k1}, \dots, \beta_{kL}] \end{array}$$

the word distribution in topic j

How do we model this? We need to put distributions in everything in Bayes' rule. Then hopefully to get a closed form expression for the posterior probability distribution. In order to achieve that, we know we need to choose a good prior and make the prior and posterior conjugates.

$$\begin{tikzcd} \text{Bernoulli} \arrow[r, leftrightarrow, "\text{conjugate}", "\text{to}"'] \arrow[d, Rightarrow, "\text{extends to}"'] & \text{Beta} \arrow[d, Rightarrow, "\text{extends to}"] \\ \text{Multinoulli} \arrow[r, leftrightarrow, "\text{conjugate}", "\text{to}"'] & \text{Dirichlet} \end{tikzcd}$$

Plate Notation



M = # documents

shading means it is observed

$$\theta \sim Dir(\alpha_1, \dots, \alpha_k) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)} \prod_{i=1}^{k} \theta_{d_i}^{\alpha_i - 1}$$

$\alpha_i \geq 0$

topics distribution (hidden/latent)

For each word n (denoting the n^{th} word) in document d, let

$$Z_{d,n} \sim Multi(\theta_d)$$

Recommender Systems

These systems automatically learn important data features instead of hard hand coding.

Collaborative Filtering

We use an alternating method of guessing iteratively to predict.

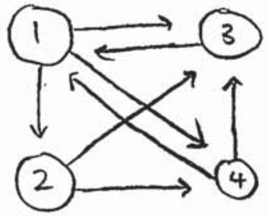
Conjugate Gradient Descent

This method is applicable to sparse systems that are too large to be handled by a direct implementation.

PageRank

- represent network as matrix (adjacency matrix)
- make it symmetric (orthogonally diagonalizable)
- decompose it by eigenvalues and eigenvectors
- understand its meaning with respect to the graph

- networks are large in real life so we need scalable algorithms



$$A = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \end{array}$$

- each arrow represents a link from the tail to the head

probability column matrix/
probability transition matrix

Dynamical Systems Point of View

Suppose initially the importance is uniformly distributed among the 4 pages, each getting 1/4 probability. Denote v to be the initial rank vector

$$v = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

We can update these values by multiplying by A iteratively.

$$v^* = \lim_{k \to \infty} A^k v = \begin{bmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{bmatrix}$$

Linear Algebra Point of View

Denote x_1, x_2, x_3, x_4 to be the importance of the 4 pages. Analyzing the graph via incoming edges gives

$$\begin{aligned} x_1 &= 1 \cdot x_3 + \tfrac{1}{2} \cdot x_4 \\ x_2 &= \tfrac{1}{3} \cdot x_1 \\ x_3 &= \tfrac{1}{3} \cdot x_1 + \tfrac{1}{2} \cdot x_2 + \tfrac{1}{2} \cdot x_4 \\ x_4 &= \tfrac{1}{3} \cdot x_1 + \tfrac{1}{2} \cdot x_2 \end{aligned} \implies A\vec{x} = \vec{x}$$

This translates to $\vec{x}$ being an eigenvector of A with eigenvalue 1. We choose v^* to be the eigenvector with the sum of all entries equal to 1.

$$\begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix} \propto \begin{bmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{bmatrix} = v^*$$

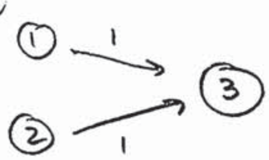
Probabilistic Point of View

We can model the process as a random walk on graphs. Each page has equal probability $1/4$ to be chosen as a starting point so the initial probability distribution is given by

$$\begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}^T$$

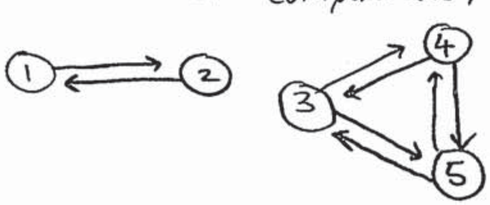
The probability that page i will be visited after one step is Ax and the probability after k steps is $A^k v$. This sequence converges to v^*, called the stationary distribution, and it will be our PageRank vector. The i^{th} entry in v^* is the probability that at any moment a random person visits page i.

Subtleties and Issues

1. 

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \qquad v_0 = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \qquad A^2 v = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

2. (Disconnected Components)



$$A = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 & 0 \end{array}\right]$$

$$v = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad u = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Here, u and v are not scaled versions of each other but both are eigenvectors with eigenvalue $\lambda = 1$.

In order to overcome these problems, fix a positive constant p between 0 and 1, which we call the damping factor (a typical value is $p = 0.15$). Define the PageRank matrix (also known as the Google matrix) to be

$$M = \underbrace{(1-p)\cdot A + p\cdot B}_{\text{convex combination}} \quad \text{where} \quad B = \frac{1}{n}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ 1 & \cdots & \cdots & 1 \end{bmatrix}$$

M remains a column stochastic matrix with only positive entries. These characteristics guarantee that M has a unique dominant eigenvalue $\lambda = 1$ that has multiplicity 1 (Perron-Frobenius Theorem). The eigenvector corresponding to this eigenvalue does not have 0 as an entry and the sum of its entries is 1.

Stochastic Matrix (probability/transition/Markov matrix)

A matrix used to describe the transitions of a Markov chain. Each entry is a nonnegative real number representing a probability. There are different definitions and types of stochastic matrices.

- right stochastic matrix is a real square matrix with each row summing to 1
- left " ———— "
  — " column " ————
- doubly " ————
  — " row and column " ———— "

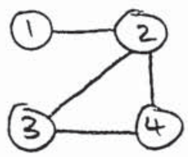
Laplacian Matrix

Given a simple graph G with n vertices, its Laplacian matrix is defined as

$$L = D - A$$

where D is the degree matrix and A is the adjacency matrix of the graph. Since G is a simple graph, A only contains 1's or 0's and its diagonal elements are all 0's. Note L is symmetric so it is orthogonally diagonalizable.

Ex:



$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{D} - \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}}_{L}$$

Assume eigenvalues of L are arranged as $\lambda_0 \le \lambda_1 \le \dots \le \lambda_{n-1}$. The number of connected components in the graph is the algebraic multiplicity of the 0 eigenvalue, i.e. the dimension of the nullspace of the Laplacian. The second smallest eigenvalue of L is the approximation of the sparsest cut of a graph (algebraic connectivity / Fiedler value of G).

Other properties of L are that L is positive semi-definite and every row sum and column sum of L is zero. This implies $\lambda_0 = 0$ because $v_0 = [1, 1, \dots, 1]^T$ satisfies $Lv_0 = 0$ so L is singular.