1 Basics

Gaussian

$f(x) = \frac{1}{\sqrt{(2\pi)\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad \mathcal{N}(x|\mu,\sigma^2)$

$f(x) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}, \quad \mathcal{N}(x|\mu,\Sigma)$

$X \sim \mathcal{N}(\mu,\Sigma),\ Y = A + BX \Rightarrow Y \sim \mathcal{N}(A + B\mu, B\Sigma B^T)$

Conditionate Gaussians

$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right) \Rightarrow a_2|a_1 \sim \mathcal{N}\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(a_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)$

Primal Dual problem

Let $\mathcal{P} = \begin{cases} \min_w f(w) \\ g_i(w) = 0\ \forall i \\ h_j(w) \le 0\ \forall j \end{cases}$

Then the Slater's condition is:
$\exists w\ |g_i(w) = 0, h_j(w) < 0\ \forall i,j$

The lagrangian is:
$\mathcal{L}(w,\lambda,\alpha) = f(w) + \sum_i \lambda_i g_i(w) + \sum_j \alpha_j h_j(w)$

$\mathcal{D} = \begin{cases} \max_{\lambda,\alpha} \theta(\alpha,\lambda) \\ \theta(\alpha,\lambda) = \min_w \mathcal{L}(w,\lambda,\alpha) \\ \alpha_j(w) \ge 0\ \forall j \end{cases}$

$\mathcal{D}$ is always a convex optimization problem. In general the solution of the $\mathcal{D}$ is smaller then $\mathcal{P}$. But if Slater's condition holds then they are equal. And we get the complementary slackness: $\alpha_j^* h_j(w^*) = 0\ \forall$

The optimal $w^* = min_w \mathcal{L}(w, \lambda^*, \alpha^*)$

Moments

- $Var[X] = E[XX^T] - E[X]E[X^T]$
- $Var[X+Y] = Var[X] + Var[Y] + 2Cov[X,Y]$
- $Cov[X,Y] = E[(X - E[X])(Y - E[Y])]$
- $Cov[aX, bY] = abCov[X,Y]$

Calculus

- $\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{A}\mathbf{x}) = (\mathbf{A}^\top + \mathbf{A})\mathbf{x} \overset{\text{A sym.}}{=} 2\mathbf{A}\mathbf{x}$
- $\frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^\top \mathbf{A}\mathbf{x}) = \mathbf{A}^\top \mathbf{b}$ • $\frac{\partial}{\partial \mathbf{X}}(\mathbf{c}^\top \mathbf{X}\mathbf{b}) = \mathbf{c}\mathbf{b}^\top$
- $x^T Ax = Tr(x^T Ax) = Tr(xx^T A) = Tr(Axx^T)$
- $\frac{\partial}{\partial A} Tr(AB) = B^T$ • $\frac{\partial}{\partial A}|A| = |A|A^{-T}$
- $\sigma(x) = \frac{1}{1+e^{-x}}$
- $\nabla\sigma(x) = \sigma(x)(1-\sigma(x)) = \sigma(x)\sigma(-x)$
- $\nabla \tanh(x) = 1 - \tanh^2(x)$
- $tanhx = \frac{sinhx}{coshx} = \frac{e^x - e^{-x}}{e^x + e^x}$ • $2\sigma(x) - 1 = tanh(x/2)$
- $f(x) \sim f(x_0) + (x - x_0)^T \nabla_f(x_0) + \frac{1}{2}(x - x_0)^T H_f(x_0)(x - x_0)$

Newton's Method

$x^{(n+1)} \leftarrow x^{(n)} - H_F^{-1}(x^{(n)})\nabla_F(x^{(n)})$

$f(x^*) = 0, f'(x^*) \neq 0 \implies Q$

$f(x^*) = 0, f^{(k)}(x^*) = 0 \implies L$

$x^{(n+1)} \leftarrow x^{(n)} - kH_F^{-1}(x^{(n)})\nabla_F(x^{(n)}) \implies Q$

Jensen's inequality

φ: convex $\rightarrow \varphi(\mathbb{E}[X]) \le \mathbb{E}[\varphi(X)]$

2 Gaussian Processes

$p(y_{n+1}|x_{n+1}, X, y) = \mathcal{N}(y_{n+1}|\mu_{y_{n+1}}, \sigma^2_{y_{n+1}})$

$\mu_{y_{n+1}} = k^T C_n^{-1} y = k^T(K + \sigma^2 I)^{-1} y$

$\sigma^2_{y_{n+1}} = c - k^T C_n^{-1} k$

$C_n = K + \sigma^2 I$

$c = k(x_{n+1}, x_{n+1}) + \sigma^2$

$k = (x_{n+1}, X), K = (X, X)$

2.1 Kernels

A function $k : \mathcal{X} \times \mathcal{X}$ is a kernel if : $k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$

$k(x,y)$ is a kernel if it's symmetric semidefinite positive:
$\forall \{x_1, \dots, x_n\}$ then for the Gram Matrix $[K]_{ij} = k(x_i, x_j)$ holds $c^T Kc \ge 0 \forall c$

Closure Properties:

$k(x,y) = k_1(x,y) + k_2(x,y), \quad k(x,y) = k_1(x,y)k_2(x,y)$

$k(x,y) = f(x)f(y),\ k(x,y) = k_3(\phi(x), \phi(y))$

$k(x,y) = \exp(\alpha k_1(x,y)), \alpha > 0,\ |X \cap Y| = kernel$

$k(x,y) = p(k_1(x,y)),\ p(\cdot)$

$k(x,y) = k_1(x,y)/\sqrt{(k_1(x,x)k_1(y,y)}$

Gaussian (rbf): $k(x,y) = \sigma^2 \exp(-\frac{\|x-y\|^2}{2l^2})$

Sigmoid: $k(x,y) = \tanh(k \cdot x^T y - b)$

Polynomial: $k(x,y) = (x^T y + c)^d, d \in N, c \ge 0$

Periodic: $k(x,y) = \sigma^2 exp(-\frac{2\sin^2(\pi|x-y|/p)}{\ell^2})$

Linear: $k(x,y) = \sigma_b^2 + \sigma^2(x-c)(y-c) = xy$

Rational Quadratic: $k(x,y) = \sigma^2(1 + \frac{(x-y)^2}{2\alpha l^2})^{-\alpha} \rightarrow RBF$

2.2 Kernel Properties

$k(u,u) \ge 0\ \ \forall u$

$k(u,v)^2 \le k(u,u)k(v,v)\ \ \forall u,v$

$\binom{n+k-1}{k}$ Comb. w rep. $\binom{n+d}{d}$ poly kernel

3 Statistics Recap

Estimation

- Consistency: $\hat{\theta}_n \xrightarrow{P} \theta$, i.e. $\forall \epsilon P\{|\hat{\theta}_n - \theta| \ge \epsilon\} \xrightarrow{n\to\infty} 0$
- Asymptotic normality: $\sqrt{N}(\theta - \hat{\theta}_n) \rightarrow \mathcal{N}(0, J^{-1}IJ^{-1})$
- Asymptotic efficiency: $\hat{\theta}_n$ minimizes $E[(\hat{\theta}_n - \theta)^2]$ as $n \rightarrow \infty$
- Not necessarily efficient for finite samples (e.g. Stein estimator of $\mathcal{N}(\theta, \sigma^2 I)$ for $d \ge 3$ is better)
- Equivariant: if $\hat{\theta}_n$ is the MLE of θ then $g(\hat{\theta}_n)$ is the MLE of $g(\theta)$ (w/ nice g)
- Possibly Biased

Rao-Cramer

$\Lambda = \frac{\partial \log \mathbb{P}(x|\theta)}{\partial \theta}$ (score function), $E[\Lambda] = 0$

Fisher information: $\mathcal{I}(\theta) = \mathbb{V}[\Lambda]$

$\mathcal{I}(\theta) = E[\Lambda^2] = -E[\frac{\partial^2 \log \mathbb{P}(x|\theta)}{\partial\theta\partial\theta^T}] = -E[\frac{\partial \Lambda}{\partial \theta}]$

Oss: For the whole model:
$\mathcal{I}_n = \mathbb{V}\left[\frac{\partial \log \mathbb{P}(x_i, i=1:n|\theta)}{\partial \theta}\right] = n\mathcal{I}$

MSE bound: $E[(\hat{\theta}_n - \theta)^2] \ge \frac{[1+b'(\hat{\theta}_n)]^2}{nE[\Lambda^2]} + b(\hat{\theta}_n)^2$

Cauchy-Schwarz: $|E(XY)|^2 \le E(X^2)E(Y^2)$

4 Linear Regression

$y = X\beta + \epsilon$ where $y \in \mathbb{R}^n,\ X \in \mathbb{R}^{n\times d},\ \beta \in \mathbb{R}^d$

Risk Decomposition Theorem

$\mathbb{E}_{Y,D}\left[\left(Y - \hat{f}(x_0)\right)^2\right] = Bias^2 + Var. + Noise$

$Bias = \left(\mathbb{E}_D\left[\hat{f}(x_0)\right] - \mathbb{E}[Y|X = x_0]\right)$

$Variance = \mathbb{E}_D\left[\left(\mathbb{E}_D\left[\hat{f}(x_0)\right] - \hat{f}(x_0)\right)^2\right]$

$Noise = \mathbb{E}_Y\left[(Y - \mathbb{E}[Y|X = x_0])^2\right]$

Combination of Regression Models:

$\text{bias}[\hat{f}(x)] = \frac{1}{B}\sum_{i=1}^B \text{bias}[\hat{f}_i(x)]$

$\mathbb{V}[\hat{f}(x)] = \frac{1}{B^2}\sum_i \mathbb{V}[\hat{f}_i(x)] + \frac{1}{B^2}\sum_{i\neq j} cov[\hat{f}_i(x), \hat{f}_j(x)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \approx \frac{\sigma^2}{B}$

Minimum square linear regression

$\hat{\beta} = \arg\min_\beta \|X\beta - y\|^2 \Rightarrow \hat{\beta} = \left(X^T X\right)^{-1} X^T y.$

Here $\hat{\beta}$ is the BLUE (Best Linear Unbiased Estimator)

Projection of Y on space of X: $X\hat{\beta}$

Lasso regression

$\hat{\beta} = \arg\min_\beta \|X\beta - y\|^2 + \lambda\|\beta\|_1 \Rightarrow \hat{\beta} =$ No closed form (LARS algorithm) but it is a convex problem

Bayesian prior: $p(\beta_i) = \frac{1}{4\sigma^2} exp\left(-|\beta_i| \frac{\lambda}{2\sigma^2}\right)$

Const. opt. $\hat{\beta} = \arg\min_\beta \|X\beta - y\|^2$ s.t. $\|\beta\|_1 < s_\lambda$

Ridge regression

$\hat{\beta} = \arg\min_\beta \|X\beta - y\|^2 + \lambda\|\beta\|_2^2 \Rightarrow \hat{\beta} = \left(X^T X + \lambda I\right)^{-1} X^T y$

Bayesian prior $p(\beta) = N(0, \frac{\sigma^2}{\lambda} I)$

Const. opt. $\hat{\beta} = \arg\min_\beta \|X\beta - y\|^2$ s.t. $\|\beta\|_2 < s_\lambda$

$X\hat{\beta}_{ridge} = X(X^T X + \lambda I)^{-1} X^T y = UD(D^2 + \lambda I)^{-1} DU^T y = \sum_{j=1}^d u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T y$

The shrinkage factor shrinks small singular values and it approaches 1 for large singular values.

5 Classification

Loss-Functions

True class: $y \in \{-1, 1\}$, pred. $z \in [-1, 1]$

Cross-entropy (log loss): ($y' = \frac{(1+y)}{2}$ and $z' = \frac{(1+z)}{2}$) $L(y', z') = -[y' log(z') + (1 - y') log(1 - z')]$

Hinge Loss: $L(y,z) = max(0, 1 - yz)$

Perceptron Loss: $L(y,z) = max(0, -yz)$

Logistic loss: $L(y,z) = log(1 + exp(-yz))$

Square loss: $L(y,z) = \frac{1}{2}(y - z)^2$

Exponential loss: $L(y,z) = exp(-yz)$

0/1 Loss: $L(y,z) = \mathbb{I}\{z \neq y\}$

Perceptron Algo

Classifier $c(x) = w^T x + w_0$. Loss $\mathcal{L}(w) = \sum_{i: y_i w^T x_i < 0} -y_i w^T x_i$. Train using (S)GD. Converges if data is linearly separable, and learning rate $\eta(k) \ge 0$, $\sum_k \eta(k) \rightarrow \infty$ and $\left(\sum_k \eta^2(k)\right)/\left(\sum_k \eta(k)\right)^2 \rightarrow 0$. Update rule (on misclassified points): $w^{(k+1)} = w^{(k)} + \eta^{(k)} x_n y_n$

Fisher Discriminant

$w^* = \arg\max_w \frac{w^T S_B w}{w^T S_w w} \propto S_w^{-1}(m_2 - m_1)$ where:

$S_B = (m_2 - m_1)(m_2 - m_1)^T$

$S_w = \sum_{i=1}^2 \sum_{n \in C_i} (x_n - m_i)(x_n - m_i)^T$

6 SVM

Like Percepton but maximizing the margin. Equivalent to

$\mathcal{P} = \begin{cases} \min_{w,w_0} \frac{\|w\|^2}{2} \\ y_i(w^T x_i + w_0) \ge 1\ \forall i \end{cases}$ where the margin size is $\frac{2}{\|w\|^2}$.

Slater conditions $\Rightarrow$

$\mathcal{D} = \begin{cases} \max_\alpha \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \alpha_i \ge 0\ \forall i \\ \sum_i \alpha_i y_i = 0 \end{cases}$

Complementary slackness $\alpha_i^* h_i(w^*) = 0$ so either $\alpha_i^* = 0$ or x_i is a Support Vector

Soft margin SVM

C small $\implies$ more misclassifications C high $\implies$ hard margin

$\mathcal{P} = \begin{cases} \min_{w,w_0,\xi} \frac{\|w\|^2}{2} + C\sum_i \xi_i \\ y_i(w^T x_i + w_0) \ge 1 - \xi_i\ \forall i \\ \xi_i \ge 0\ \forall i \end{cases}$

$$\mathcal{D} = \begin{cases} \max_{\alpha_i} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ 0 \le \alpha_i \le C\ \forall i \\ \sum_i \alpha_i y_i = 0 \end{cases}$$

$\xi_i^* = \max(0, 1 - y_i(w^{*T}x_i + w_0^*))$

$w^* = (\sum_i \alpha_i^* y_i x_i)$

$y = sgn\left(w^{*T}x\right) = sgn\left(\left(\sum_i \alpha_i^* y_i x_i)\right)^T x_j\right)$

Non linear SVM: $x_i^T x_j \to \phi(x_i)^T\phi(x_j) \to k(x_i, x_j)$

Multiclass SVM (one v. rest)

Train a binary classifier for each class (one vs the rest). Then assign a score $f_c(x) = w_c^T x$. Predicitons: $c^* = \arg\max_c f_c(x)$

Structured SVM

Too many class for ovr. $\Psi : X \times Y \to \mathbb{R}^{m+d}$ is called Joint feature map $\mathcal{P} =$

$$\begin{cases} \min_{w,w_0} \frac{\|w\|^2}{2} + \frac{C}{n}\sum_{i=1}^n \xi_i \\ w^T\Psi(x_i,y_i) \ge \Delta(y_i,y') + w^T\Psi(x_i,y') - \xi_i\ \forall i\ \forall y' \ne y_i \\ \xi \ge 0\ \forall i \end{cases}$$

Theorem Δ as Loss (Structured SVM in Statistical Learning):

$\hat{\mathcal{R}}(\mathcal{Z}_{train}) \doteq \frac{1}{n}\sum_{i=1}^n \Delta(y_i, c_{w^*}(x_i)) \le \frac{1}{n}\sum_{i=1}^n \xi_i^*$

7 Ensemble method

Bagging

We train $b^{(1)}, \ldots, b^{(M)}$ different classifiers.

Then $\overline{b}(x) = \begin{cases} \frac{1}{M}\sum_{i=1}^M b^{(i)}(x) & \text{regression} \\ \text{majority}\left(b^{(i)}\right) & \text{classification} \end{cases}$

Works if: the $b^{(i)}$ are almost independent. Bagging classifiers worse than random chance does not achieve good results

7.0.1 Theorem:

if $|y| < \infty$ then $\exists M$ large enough s.t.

$\mathbb{E}_{Z,Z',Y|x}\left[(Y - \overline{b}(x))^2\right] \le \mathbb{E}_{Z,Z',Y|x}\left[(Y - b^{(i)}(x))\right]$

Random Forest

Sample B datasets $Z^1, \ldots, Z^B$ from Z with replacement. For each Z^b train a full decision tree $f^b(x)$ with one small modification: before each split randomly subsample $k \le d$ features and only consider these for your split.

Adaboost

Boosting: Train weak learners sequentially on all data, but reweight misclassifed samples higher, Bias $\downarrow$

Initialize weights $w_i = 1/n$, for b=1:B do:

1. Fit classifier $c_b(x)$ with weights w_i
2. Compute error $\epsilon_b = \sum_i w_i^{(b)}\mathbb{1}_{[c_b(x_i)\ne y_i]} / \sum_i w_i^{(b)}$
3. Compute coeff. $\alpha_b = log(\frac{1-\epsilon_b}{\epsilon_b})$
4. Update weights $w_i = w_i \exp(-\alpha_b y_i c_b(x_i))$
5. Normalize w_i dividing by $Z = 2\sqrt{\epsilon(1-\epsilon)}$

Return $\hat{c}_B(x) = \text{sign}\left(\sum_{b=1}^B \alpha_b c_b(x)\right)$

Loss: Exponential loss $L(y, y') = exp(-yy')$

Model: Forward Stagewise Additive

Oss: Self averaging algos that train Spiky interpolating classifiers.

AdaBoost trains max-margin classifier.

8 Mixtures Models (Unsupervised Learning)

K-means

We find $\mu_1, \ldots, \mu_k$ such that our predictions are $c(x) : \mathbb{R}^d \to \{1, \ldots, k\}$.

Find $c(\cdot)$ and $\mu_i \forall i$ that minimize:

$\mathcal{R}^{km}(c, \mu_i \forall i) = \sum_x \left\|x - \mu_{c(x)}\right\|^2$

> Initialize $\mu_i \forall i$;
> **while** *μ_i are changing* **do**
> $\quad c(x) \leftarrow \arg\min_c \left\|x - \mu_c\right\|^2\ \forall x$;
> $\quad \mu_\alpha = \frac{1}{n_\alpha}\sum_{x:c(x)=\alpha} x\ \forall \alpha$;

Gaussian Mixtures

1) Draw $z \sim \pi$ Categorical.
2) Draw $x \sim N(\mu_z, \Sigma_z)$

Expectation Maximization

> Initialize $\theta^0 = \pi^0, \mu^0, \sigma^{2\,0}$;
> **while** $\left\|\theta^{j+1} - \theta^j\right\| > \epsilon$ **do**
> $\quad$ E-step:
> $\quad \gamma_{xc} \doteq \mathbb{E}\left[M_{xc}|X, \theta^j\right] =$
> $\quad \frac{p(X|c,\theta^j), p(c|\theta^j)}{p(x|\theta^j)} = \frac{N(\mu_c^j, \sigma_c^{2j})\pi_c^j}{\sum_\nu \pi_\nu N(\mu_\nu, \sigma_\nu^{2j})}$
> $\quad Q(\theta, \theta^j) = \mathbb{E}\left[\sum_i \log p(x_i, z_i|\theta)\right]$
> $\quad = \sum_{x\in X}\sum_c (\gamma_{xc}\log(\pi_c P(x|\theta_c)))$
> $\quad$ M-step: $\theta_{j+1} = \arg\max_\theta Q(\theta, \theta_j)$
> $\quad \pi_c^{j+1} = \frac{1}{|X|}\sum_{x\in X}\gamma_{xc}$
> $\quad \mu_c^{j+1} = \frac{\sum_{x\in X}\gamma_{xc}x}{\sum_{x\in X}\gamma_{xc}}$
> $\quad \sigma_c^{2\,j+1} = \frac{\sum_{x\in X}\gamma_{xc}(x-\mu_c)^2}{\sum_{x\in X}\gamma_{xc}}$

Where $M_{xc} = \mathbb{I}_{\{x \text{ generated by } c\}}(x))$

9 Neural Network

Backpropagation

Let $\Phi(x) = f_{\theta_n}^{(n)} \circ f_{\theta_{n-1}}^{(n-1)} \circ \cdots \circ f_{\theta_1}^{(1)}(x)$

$\partial_\Phi f^{(i)} \doteq \partial_z f^{(i)}(z, \theta_i)|_{z=\Phi^{(i-1)}(x)}$

$\partial_\theta f^{(i)} \doteq \partial_z f^{(i)}(\Phi^{(i-1)}(x), \theta)|_{\theta=\theta_i}$

> **Result:** $\partial_{\theta_i}\Phi(x)\forall i$
> Initialize $B = 1$;
> **for** $i \leftarrow n, n-1, \ldots, 1$ **do**
> $\quad \partial_{\theta_i}\Phi(x) \leftarrow B\partial_\theta f^{(i)}$;
> $\quad B \leftarrow B\partial_\Phi f^{(i)}$;

Once we have this we can $\nabla \downarrow$

Robinson-Monro

Given $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $\underline{Z}$ random variable over $\mathbb{R}^m$, compute $\underline{\theta}^*$ s.t. $\mathbb{E}_{\underline{Z}}[f(\underline{Z}, \underline{\theta})] = 0$. Iteratively sample $\underline{z}^{(k)} \sim \underline{Z}$ and set $\underline{\theta}^{(k)} \leftarrow \underline{\theta}^{(k-1)} - \eta(k)f(\underline{z}_k, \underline{\theta}^{(k-1)})$.

For SGD, $f(\underline{z}, \underline{\theta}) = \nabla_\theta \mathcal{L}(y, NN_{\underline{\theta}}(x))$.

$\underline{z} \to X, y$

Thm: R–M (and thus SGD) converges if $\eta(k) \ge 0$, $\sum_{k=1}^\infty \eta(k) = \infty$, $\sum_{k=1}^\infty \eta^2(k) < \infty$ and some regularity conditions on $\mathbb{E}_Z[f(z, \theta)]$ hold.

Stocastic Gradient Descent

> **Result:** optimal θ^*
> Initialize θ;
> **while** *Test error is decreasing* **do**
> $\quad \nabla_\theta Loss = \sum_{(x,y)\in S_k}\nabla_\theta \mathcal{L}(NN(x), y)$;
> $\quad \theta \leftarrow \theta - \eta(k)\nabla_\theta Loss$;

Oss: $S_k \in D$ and changes at each iteration (Mini Batch)

Oss: As long as $\sum_k \eta(k) = \infty$ and $\sum_k \eta^2(k) < \infty$ the SGD converges

Advantages over Normal Gradient Descent: 1) Can handle large Dataset 2) Faster improvment (w.r.t. time, not iterations) 3) Escapes local minima 4) Lower generalization error

Regularization techniques: 1)Dropout 2)Batch norm. 3)Early stop 4)Weight decay

Avoid dying ReLu:

$$\begin{cases} \alpha g(z), \text{if } z < 0 \\ z, \text{if } z \ge 0 \end{cases}$$

where $g(z)$ is $(\exp(z) - 1)$ for ELU and z for LeakyReLU. Dropout slows down training (not inference): noisy updates.

10 Autoencoders

Infomax principle

$I(X, Y) \doteq H(X) - H(X|Y) =$ mutual inform

$\theta^* = \arg\max_\theta I(X, enc_\theta X) = \mathbb{E}_Z[\log(P(X, Z) - \log(P(X)P(Z))]$

$\theta^* \simeq \arg\max_\theta \sum_i \mathbb{E}_Z\left[\log p(x_i|Z)\right]$

It is informative but not Disentangled and Robust

Variation Autoencoders

Find encoder $q_\phi(z|x)$ and decoder $p_\theta(x|z)$ as: $argmax_{\theta,\phi}\sum_i \log p_{\theta,\phi}(x_i)$:

$\log p_{\theta,\phi}(x_i) = E_{z\sim q_{\phi(\cdot|x_i)}}\left[\log\left(\frac{p_\theta(x_i,z)}{p_\theta(z|x_i)}\frac{q_\phi(z|x_i)}{q_\phi(z|x_i)}\right)\right] =$

$E_{z\sim q_{\phi(\cdot|x_i)}}\left[\log(\frac{p_\theta(x_i,z)}{q_\phi(z|x_i)})\right] + E_{z\sim q_{\phi(\cdot|x_i)}}\left[\log\frac{q_\phi(z|x_i)}{p_\theta(z|x_i)}\right] =$

$E_{z\sim q_{\phi(\cdot|x_i)}}[\log p_\theta(x_i|z)] - D_{KL}(q_\phi(\cdot|x_i)\|p(\cdot)) + D_{KL}(q_\phi(\cdot|x_i)\|p_\theta(\cdot|x_i))$

First term is Infomax and the second one is a regularization term.

11 Nonparametric Bayesian methods

$\beta(x|a, b) = \frac{x^{(a-1)}\cdot(1-x)^{(b-1)}}{B(a,b)}$, $B(\alpha) = \frac{\prod_{k=1}^n \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^n \alpha_k)}$

$Dir(x|\alpha) = \frac{1}{B(\alpha)}\prod_{k=1}^n x_k^{\alpha_k - 1}$

Chinese Restourant Process

$p(\text{cust}_{n+1} \text{ joins table } \tau|\mathcal{P}) = \begin{cases} \frac{|\tau|}{\alpha+n} & \tau \in \mathcal{P} \\ \frac{\alpha}{\alpha+n} & \tau \notin \mathcal{P} \end{cases}$

de Finetti: $p(X_1, \ldots, X_n) = \int(\prod_{i=1}^n p(x_i|G))dP(G)$

Stick breaking: $\rho = \{\rho_i\}_{i\in\mathbb{N}} \sim GEM(\alpha)$ if:

$\rho_k = \beta_k\left(1 - \sum_{i=1}^{k-1}\rho_k\right)$Higher α, smaller pieces

Then $G(\theta) = \sum_{i=1}^\infty \rho_k\delta_{\theta_k}(\theta)$, $\theta_k \sim H$

$\Rightarrow G \sim DP(\alpha, H)$

Gibbs Sampling

DP generative model:

- Centers of the clusters: $\mu_k \sim \mathcal{N}(\mu_0, \sigma_0)$
- Prob.s of clusters: $\rho = \{\rho_k\}_{k=1}^\infty \sim GEM(\alpha)$
- Assignments to clusters: $z_i \sim Categorical(\rho)$
- Coordinates of data points: $\mathcal{N}(\mu_{z_i}, \sigma)$

$p(z_i = k|\boldsymbol{z}_{-i}, \boldsymbol{x}, \alpha, \boldsymbol{\mu}) = \begin{cases} \frac{N_{k,-i}}{\alpha+N-1}p(x_i|\boldsymbol{x}_{-i,k}, \boldsymbol{\mu})\ \exists k \\ \frac{\alpha}{\alpha+N-1}p(x_i|\boldsymbol{\mu}) \text{ otherwise} \end{cases}$

12 PAC Learning

Empirical error: $\hat{\mathcal{R}}_n(c) = \frac{1}{n}\sum_{i=1}^n \mathbb{I}_{\{c(x_i)\ne y\}}$

Expected error: $\mathcal{R}(c) = P\{c(x) \ne y\}$

ERM: $\hat{c}_n^* = \arg\min_{c\in\mathcal{C}}\hat{\mathcal{R}}_n(c)$

Generalization error: $\mathcal{R}(\hat{c}_n^*) = P\{\hat{c}_n^*(x) \ne y\}$

$\mathcal{A}$ can learn c if $\exists\pi \in$ Polynomials s.t.:

- $\forall$ distribution $\mathcal{D}$ over X
- $\forall\epsilon \in (0, \frac{1}{2})$, $\forall\delta \in (0, \frac{1}{2})$
- $\forall n \ge \pi(\frac{1}{\epsilon}, \frac{1}{\delta}, \text{size}(c))$

then $\mathbb{P}_{\mathcal{Z}\sim\mathcal{D}}(\mathcal{R}(\mathcal{A}(\mathcal{Z})) - \inf_{c\in\mathcal{C}}\mathcal{R}(c) \le \epsilon) \ge 1 - \delta$

If A runs in time polynomial in $1/\epsilon, 1/\delta$, we say that C is efficiently PAC learnable.

VC ineq.:

$\mathcal{R}(\hat{c}_n^*) - \inf_{c\in\mathcal{C}}\mathcal{R}(c) \le 2\sup_{c\in\mathcal{C}}|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)|$

$P\{\mathcal{R}(\hat{c}_n^*) - \mathcal{R}(c^*) > \epsilon\} \le P\{\sup_{c\in\mathcal{C}}|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)| > \frac{\epsilon}{2}\}$

$P\{\mathcal{R}(\hat{c}_n^*) - \mathcal{R}(c^*) > \epsilon\} \le 2|\mathcal{C}|exp(-2n\epsilon^2/4)$ if $\mathcal{C}$ is finite

$P\{\mathcal{R}(\hat{c}_n^*) - \mathcal{R}(c^*) > \epsilon\} \le 9n^{\mathcal{VC}_\mathcal{C}}exp(-n\epsilon^2/32)$ if $|\mathcal{C}|$ is infinite

where the $\mathcal{VC}$ dimension of a function class $\mathcal{C}$ is the maximum number of points that can be arranged so that $\mathcal{C}$ shatters them.