Evaluating predict

Monday, April 24, 2017 3:25 PM

Have test set $\{x_i, y_i\}$, want to evaluate

predict method $\hat{y}_i = \begin{cases} 1, & s_i > \tau \\ 0, & s_i \leq \tau \end{cases}$

Confusion Matrix

	Predict 1	Predict 0
Actual 1	True Positive (TP)	False Negative (FN)
Actual 0	False Positive (FP)	True Negative (TN)

Predict

1. Calculate score s_i for each i.
2. Order scores: $s_{a_1} \geq s_{a_2} \geq s_{a_3} \geq \dots \geq s_{a_n}$

   where $a_1, \dots, a_n$ is permutation of $\{1, \dots, n\}$.
3. Predict $\hat{y}_{a_1}, \dots, \hat{y}_{a_T} = 1$ and $\hat{y}_{a_{T+1}}, \dots, \hat{y}_{a_n} = 0$.

s:	3.1	3	2.8	2.6	2.3	2.1	1.1	1	.8	.6	.5	.2	
$\hat{y}$	1	1	1	1	1	1	0	0	0	0	0	0	$\tau = 6$
y	1	0	1	1	0	1	0	1	0	0	1	0	

Conf. TP = 4 | FN = 2

Conf. Matrix

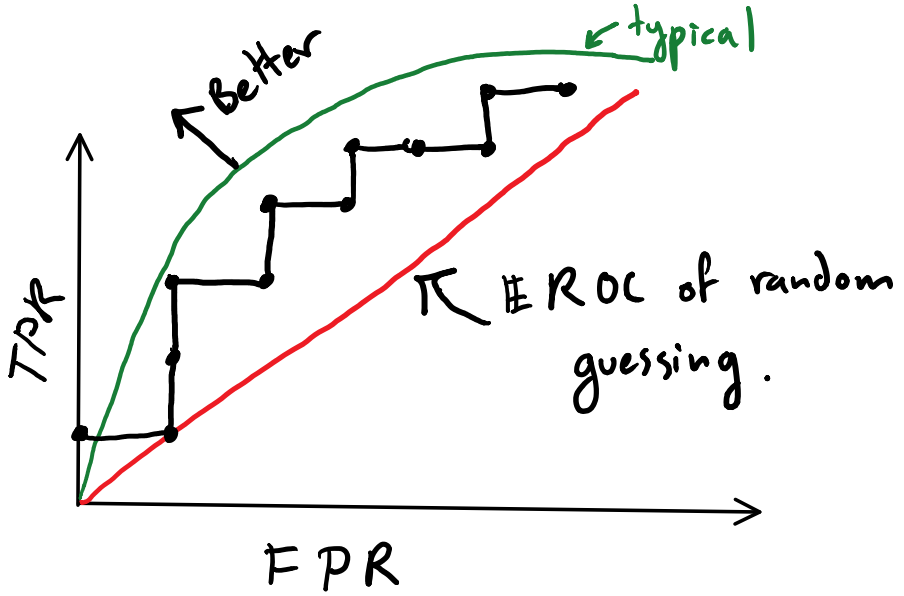
TP = 4	FN = 2
FP = 2	TN = 4

Reciever - Operating Characteristic (ROC)

True positive rate (TPR) = $\frac{TP}{TP + FN}$ } Actual Positive
(aka recall)

False positive rate (FPR) = $\frac{FP}{FP + TN}$ } Actual Neg.

T	1	2	3	4	5	6	7	8
TPR	1/6	1/6	2/6	3/6	3/6	4/6	4/6	5/6
FPR	0/6	1/6	1/6	1/6	2/6	2/6	3/6	3/6



- Every element of confusion matrix is represented in ROC

Precision - Recall Curve

In recommendation systems, many 0's – few 1's.

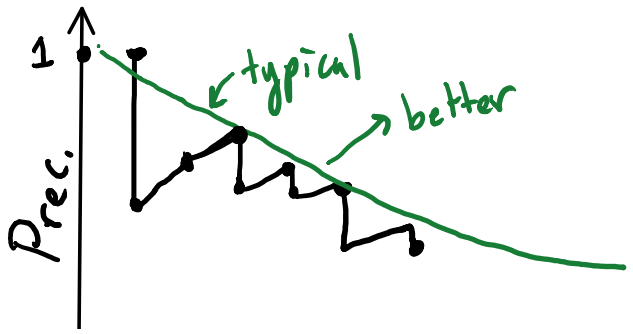
ex
- Link prediction in sparse graphs (friend recomm.)
- Search engine (most pages are not relevant)
- Document retrieval

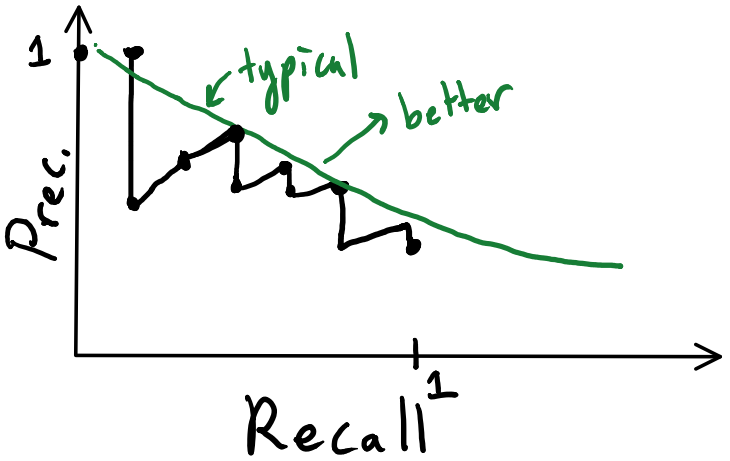
$$\text{Recall} = \text{TPR} = \frac{TP}{TP + FN}$$

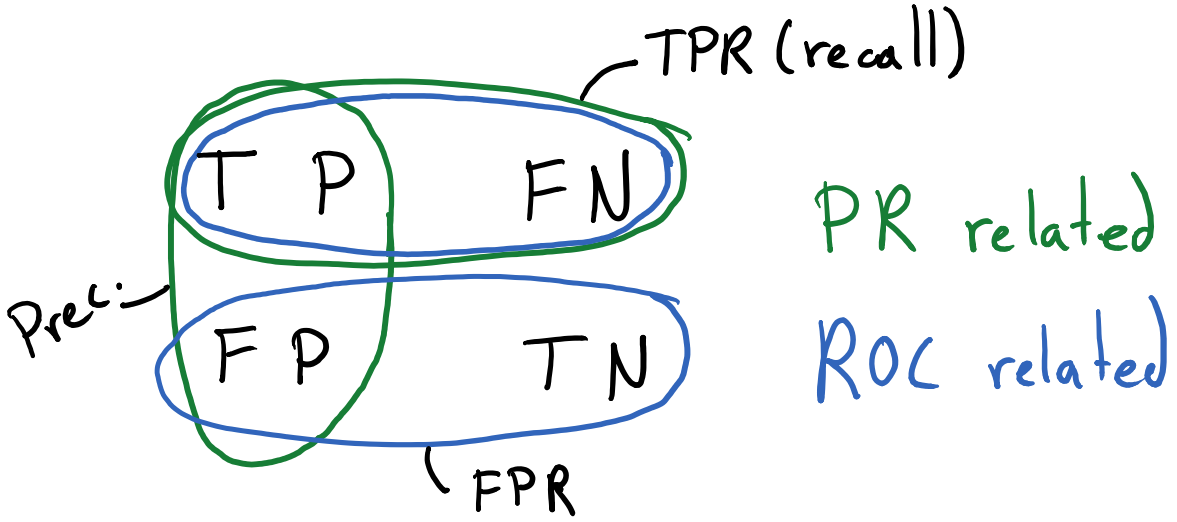
$$\text{Precision} = \frac{TP}{TP + FP}$$

} Neither is sensative to increasing TN

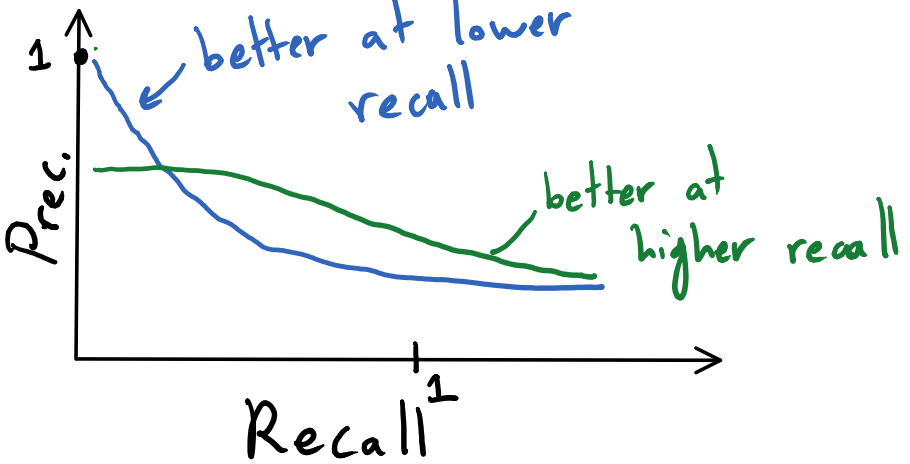
T	1	2	3	4	5	6	7	8	
Recall	1/6	1/6	2/6	3/6	3/6	4/6	4/6	5/6	...
Prec.	1	1/2	2/3	3/4	3/5	4/6	4/7	5/8	...







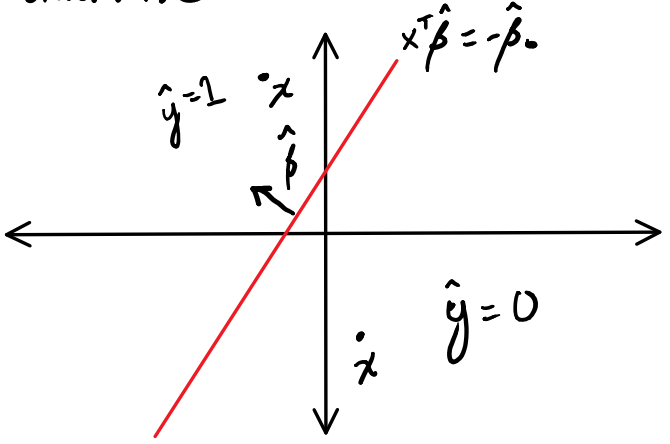
Comparing PR Curves



Margin Based Methods

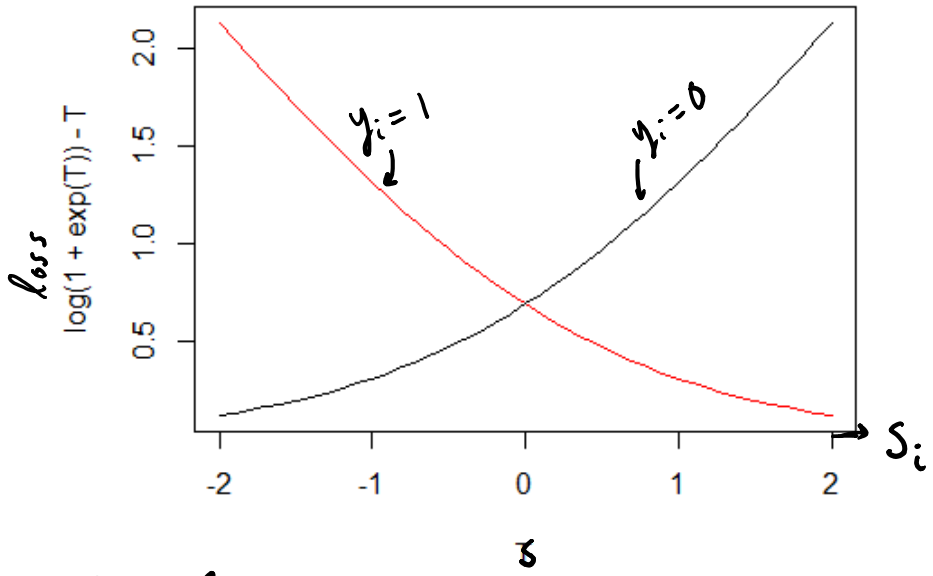
Tuesday, April 25, 2017 4:22 PM

$$\hat{y} = \begin{cases} 1, & \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p \geq 0 \\ 0, & \text{otherwise} \end{cases}$$



Empirical Risk Minimization

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \text{loss}(y_i, x_i, \beta)$$



$$s_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}$$

0-1 loss $(y_i = 1)$

$$\text{loss}_{0\text{-}1} = \begin{cases} 1, & s_i < 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\min \frac{1}{n} \sum_i \text{loss}_{0\text{-}1}(y_i, x_i, \beta)$$

is very hard to optimize!

Logistic regression $(y_i = 1)$

Support vector machines

Logistic regression ($y_i = 1$)

$$\text{loss} = \log\left(1 + e^{-s_i}\right)$$

(for $y_i = 0$ switch $s_i \leftarrow -s_i$)

Support vector machines

$$\text{loss} = \begin{cases} 1 - s_i, & s_i < 1 \\ 0, & s_i > 1 \end{cases}$$

$$\min_{\beta} \frac{1}{n}\left[\sum_{i:y_i=1} \text{loss}(1, x_i, \beta) + \sum_{i:y_i=0} \text{loss}(0, x_i, \beta)\right]$$

$$\min_{\beta} \frac{1}{n}\left[\sum_{i:y_i=1} \alpha\, \text{loss}_{0\text{-}1}(1, x_i, \beta) + \sum_{i:y_i=0} \text{loss}(0, x_i, \beta)\right]$$

$$\alpha = \frac{\sum_{i:y_i=1} 1}{n}$$

Overfitting and the Kernel Trick

Friday, May 12, 2017 11:57 AM

x_i - 2 dimensional feature vector

$\varphi_\ell(x_i)$ - ℓ^{th} feature in high-dimensional embedding

ex $\varphi_1(x_i) = 1$, $\varphi_2(x_i) = x_{i1}$, $\varphi_3(x_i) = x_{i2}$,
$\varphi_4(x_i) = x_{i1}^2$, $\varphi_5(x_i) = x_{i1} \cdot x_{i2}$, $\varphi_6(x_i) = x_{i2}^2$

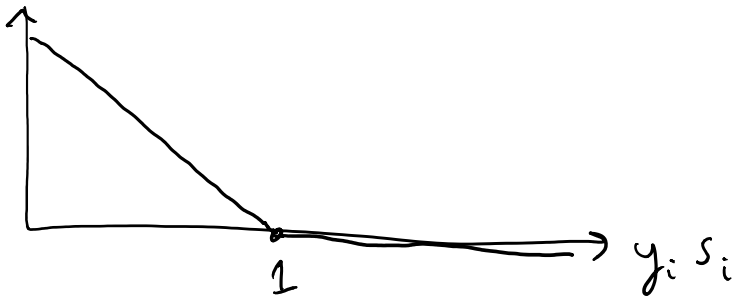
$z_i = \varphi(x_i)$ - 6 dimensional engineered feature vector

Support Vector machines

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \underbrace{\left(1 - y_i \left(\sum_{j=1}^{p} z_{ij} \beta_j\right)\right)_+}_{s_i} + \underbrace{\lambda \sum_{j=1}^{p} \beta_j^2}_{\text{unique solution}}$$

$y_i \in \{1, -1\}$

$$m_+ = \begin{cases} m, & m \geq 0 \\ 0, & m < 0 \end{cases}$$



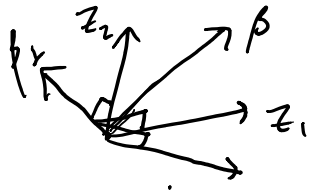
Use other losses ...

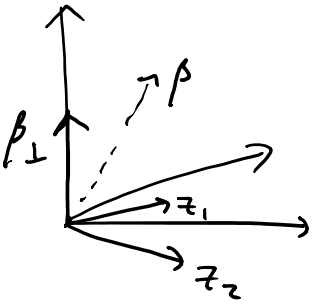
$$(\star) \quad \min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, z_i^T \beta) + \lambda \sum_{j=1}^{p} \beta_j^2$$

Kernel trick

$$\beta = \sum_i \alpha_i z_i + \beta_\perp$$

$\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ $\quad \beta_\perp^T z_i = 0$





$$z_i^T \beta = \sum_k \alpha_k z_i^T z_k + \cancel{z_i^T \beta_\perp}$$

$$z_i^T \beta = \sum_k \alpha_k z_i^T z_k + \cancel{z_i^T \beta_\perp} \to 0$$

$$(\star) \quad \min_{\alpha, \beta_\perp} \frac{1}{n} \sum_{i=1}^n \ell\left(y_i, \sum_k \alpha_k z_i^T z_k\right) + \underbrace{\lambda \sum_{j=1}^p \left(\sum_k \alpha_k z_{ij} + \beta_{\perp j}\right)^2}_{(R)}$$

$$(R) = \lambda \sum_j \left(\sum_k \alpha_k z_{kj}\right)^2 + \underbrace{2\beta_{\perp j}\left(\sum_k \alpha_k z_{kj}\right)}_{(C)} + \beta_{\perp j}^2$$

$$(C) = 2\lambda \sum_j \sum_k \alpha_k \beta_{\perp j} z_{kj} = 2\lambda \sum_k \alpha_k \underbrace{\sum_j \beta_{\perp j} z_{kj}}_{\beta_\perp^T z_k = 0}$$

(dependent on α)

$$(\star) \quad \min_{\alpha, \beta_\perp} \cdots + \lambda \sum_j \beta_{\perp j}^2 \qquad \beta_\perp = 0 \text{ for any minimizer}$$

$$(\star) \quad \min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell\left(y_i, \sum_k \alpha_k z_i^T z_k\right) + \lambda \underbrace{\sum_j \left(\sum_k \alpha_k z_{kj}\right)^2}$$

$$z_i^T z_k = \Phi(x_i)^T \Phi(x_k) =: K(x_i, x_k)$$

$$\sum_j \sum_{k_1, k_2} \alpha_{k_1} \alpha_{k_2} z_{k_1 j} z_{k_2 j}$$

$$= \sum_{k_1} \sum_{k_2} \alpha_{k_1} \alpha_{k_2} z_{k_1}^T z_{k_2}$$

$$(\star) \quad \min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell\left(y_i, \sum_k \alpha_k K(x_i, x_k)\right) + \lambda \sum_{k_1, k_2} \alpha_{k_1} \alpha_{k_2} K(x_{k_1}, x_{k_2})$$

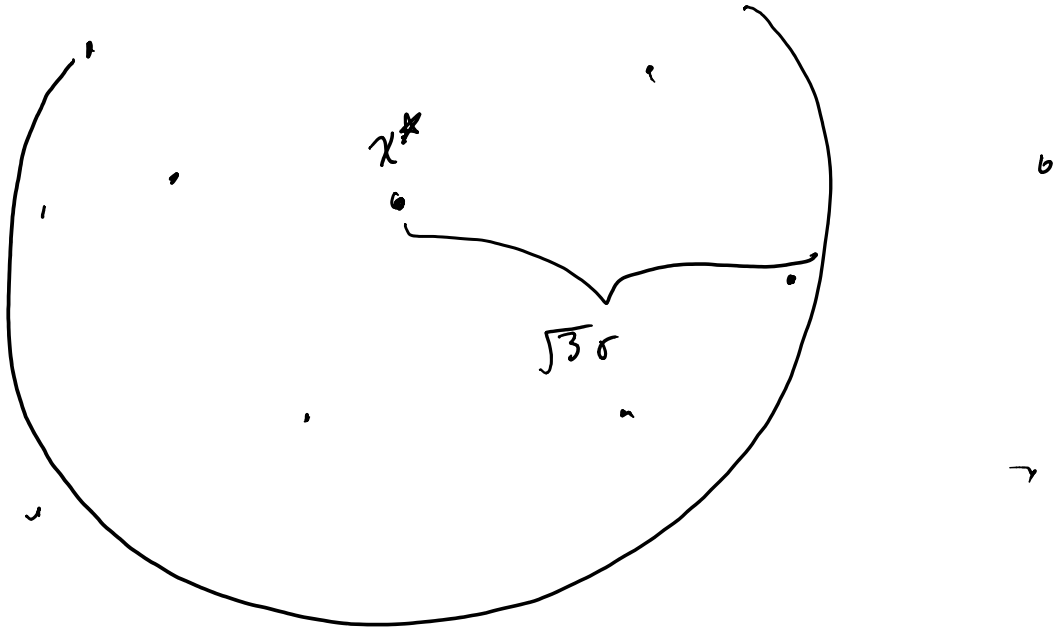
ex d^{th} degree poly kernel: $K(x_i, x_k) = (1 + x_i^T x_k)^d$

$d=2$: $(1 + x_{i1}x_{k1} + x_{i2}x_{k2})^2 = 1 + 2x_{i1}x_{k1} + 2x_{i2}x_{k2} + x_{i1}^2 x_{k1}^2 + x_{i2}^2 x_{k2}^2 + 2x_{i1}x_{i2}x_{k1}x_{k2}$

$= (1, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, \sqrt{2}x_{i1}x_{i2}, x_{i1}^2, x_{i2}^2)^T$

$(\dots x_k \dots)$

ex $K(x_i, x_k) = e^{-\|x_i - x_k\|_2^2/\sigma^2}$ Radial basis function



σ large: high bias low variance

σ small: low bias high variance