Statistical Learning Models for Text and Graph Data
Text Categorization 2: Clustering

Yangqiu Song

Hong Kong University of Science and Technology

yqsong@cse.ust.hk

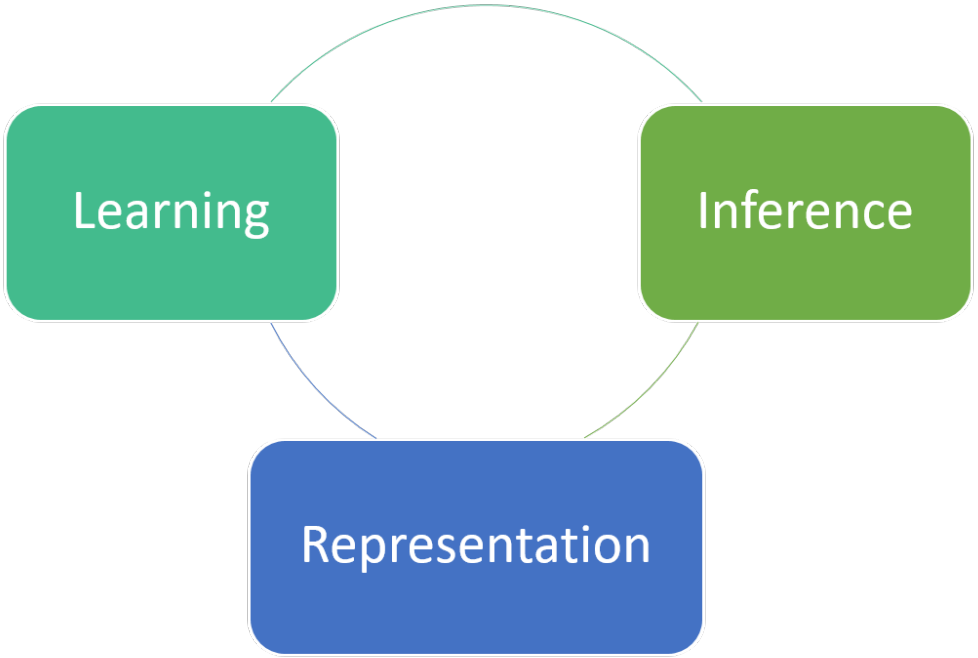
October 18, 2019

∗Contents are based on materials created by Noah Smith, Xiaojin (Jerry) Zhu, Eric Xing, Vivek Srikumar, Dan Roth

Reference Content

- Noah Smith. CSE 517: Natural Language Processing https://courses.cs.washington.edu/courses/cse517/16wi/
- Xiaojin (Jerry) Zhu. CS 769: Advanced Natural Language Processing. http://pages.cs.wisc.edu/~jerryzhu/cs769.html
- Eric Xing. 10715 Advanced Introduction to Machine Learning. https://www.cs.cmu.edu/~epxing/Class/10715/lectures/lecture1.pdf
- Vivek Srikumar. CS 6355 Structured Prediction. https://svivek.com/teaching/structured-prediction/spring2018/
- Dan Roth. CS546: Machine Learning and Natural Language . http://l2r.cs.uiuc.edu/~danr/Teaching/CS546-16/

Course Organization



- Representation: language models, word embeddings, topic models, knowledge graphs
- Learning: supervised learning,unsupervised learning, semi-supervised learning, distant supervision, indirect supervision, sequence models, deep learning, optimization techniques
- Inference: constraint modeling, joint inference, search algorithms

Overview

∗Contents are based on materials created by Noah Smith, Xiaojin Zhu, Eric Xing, Vivek Srikumar, Dan Roth

Clustering

- Clustering is an unsupervised learning method
- Given items $\mathbf{x}_1, \ldots, \mathbf{x}_M \in \mathbb{R}^d$, the goal is to group them into reasonable clusters
- We also need a pairwise distance/similarity function between items, and sometimes the desired number of clusters
- When documents are represented by feature vectors, a commonly used similarity measure is the cosine similarity

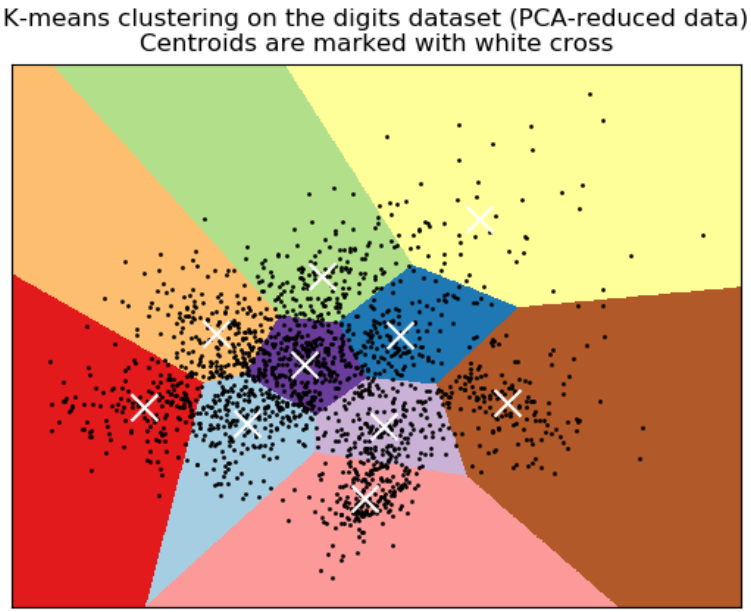
$$sim(\mathbf{x}, \mathbf{x}') = \cos(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^\top \mathbf{x}'}{||\mathbf{x}|| \cdot ||\mathbf{x}'||}$$

- This similarity has the nice property that document length is implicitly normalized (so that a long document can be similar to a short document)

K-Means Clustering

1. Randomly choose K centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
2. Repeat
   3. Assign $\mathbf{x}_1, \ldots, \mathbf{x}_M$ to their nearest centers to obtain $\hat{y}_m$, respectively
   4. Update $\boldsymbol{\mu}_k = \frac{1}{\sum_m I(\hat{y}_m = k)} \sum_m \mathbf{x}_m I(\hat{y}_m = k)$
5. Until the clusters no longer change

Step 3 is equivalent to creating a Voronoi diagram under the current centers

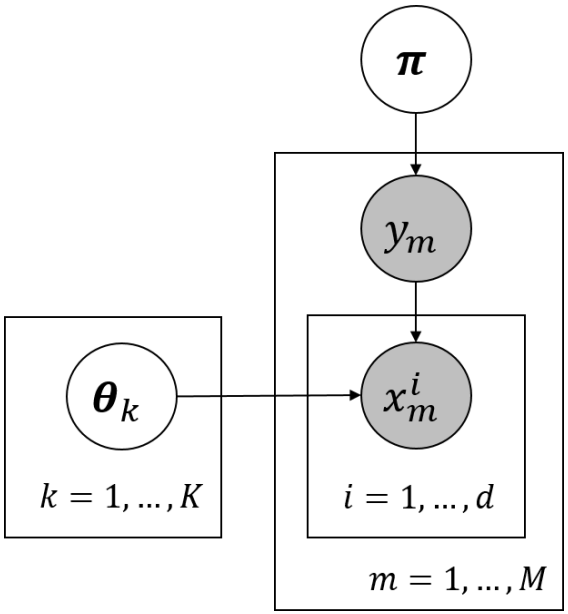


http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html

K-Means Clustering Remarks

- K-means clustering is sensitive to the initial cluster centers
- It is in fact an optimization problem with a lot of local optima
  - To be exact, k-means clustering is a special case of Gaussian Mixture Model (GMM) when the covariance of the Gaussian components tends to zero
- It is of course sensitive to k too
- Both should be chosen with care

Recall Naive Bayes Classifier: A Generative View

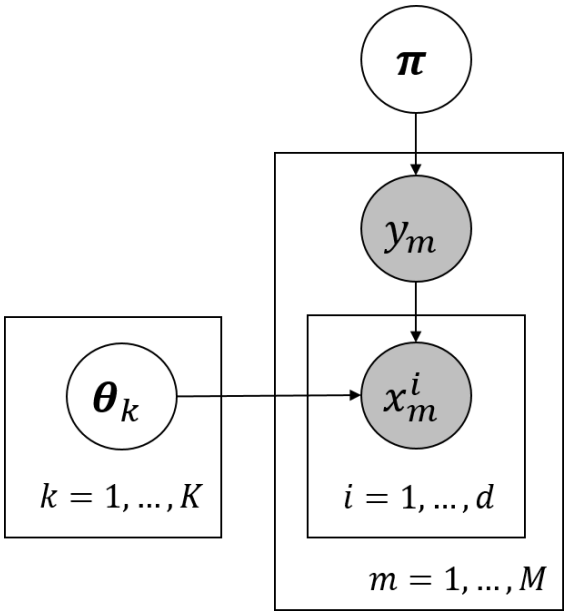


Both y_m and $\mathbf{x}_m = (x_m^1, \ldots, x_m^d)^T$ are observed variables; $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are parameters

Naive Bayes from Class Conditional Unigram Model

- For $m = 1, \ldots, M$
  - Choose $y_m \sim Multinomial(y_m|1, \boldsymbol{\pi})$
  - Choose $N_m = \sum_j^d x_m^j \sim Poisson(\xi)$
  - For $n = 1, \ldots, N_m$
    - Choose $v \sim Multinomial(v|1, \boldsymbol{\theta}_{*|y_m}) = \prod_{j=1}^d (\theta_{*|y_m}^j)^{v=j}$

Parameter Estimation (based on Multinomial)



Maximum likelihood of the training set:

$$\mathcal{J} = \log \prod_{m=1}^{M} P_{\boldsymbol{\pi}, \{\boldsymbol{\theta}_k\}}(\mathbf{x}_m, y_m)$$

$$\pi_k = \frac{|\{y_m = k\}|}{M}$$

$$\theta_k^j = \frac{\sum_{m, y_m = k} x_m^j}{\sum_{m, y_m = k} \sum_{j=1}^{d} x_m^j}$$

Both y_m and $\mathbf{x}_m = x_m^1, \ldots, \mathbf{x}_m^d$ are observed variables; $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are parameters

What if the documents are not labeled?

In naive Bayes, both y_m and $\mathbf{x}_m = (x_m^1, \ldots, x_m^d)^T$ are observed variables; $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are parameters

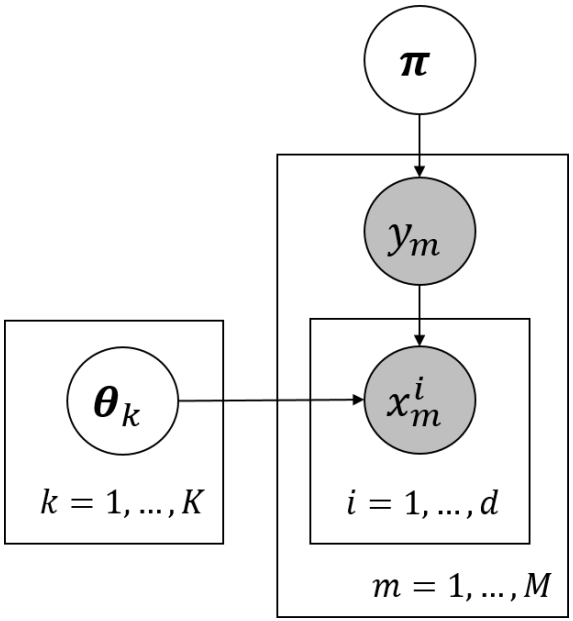


Figure: Native Bayes

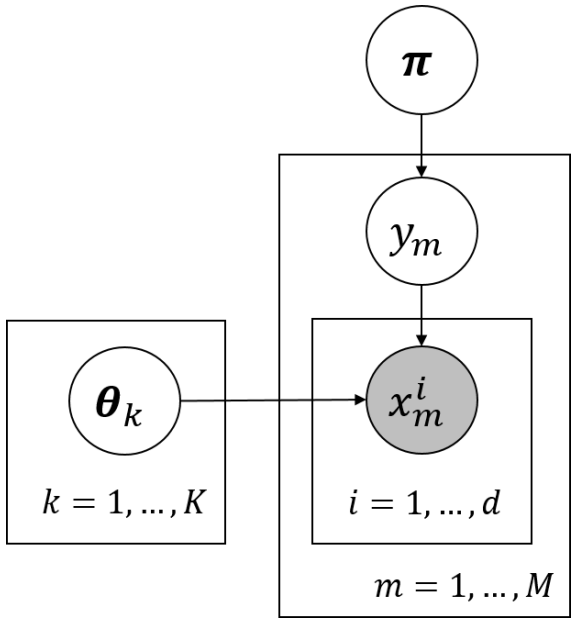


Figure: Mixture Model

However, in clustering problems, y_m is not observed (labeled before feeding into machine learning algorithm)

Expectation Maximization (EM) Algorithm

- EM might look like a heuristic method. However, it is not.
- EM is guaranteed to find a local optimum of data log likelihood
- Recall if we have complete data set $\{\mathbf{x}_m, y_m\}_{m=1}^M$ and denote parameter set as $\Theta = \{\boldsymbol{\pi}, \{\boldsymbol{\theta}_k\}\}$, the likelihood estimation of native Bayes is

$$\mathcal{J}_{NB}(\Theta) = \log \prod_{m=1}^{M} P_{\boldsymbol{\pi},\{\boldsymbol{\theta}_k\}}(\mathbf{x}_m, y_m) = \log P(\{\mathbf{x}_m, y_m\}_{m=1}^M | \Theta)$$

- However, now $\{y_m\}_{m=1}^M$ are not observed (labeled), so we treat them as hidden variables
- We instead maximize the marginal log likelihood:

$$\mathcal{J}(\Theta) = \log P(\{\mathbf{x}_m\}_{m=1}^M | \Theta)$$

Maximizing the Marginal Log Likelihood

We optimize following objective function:

$$\begin{aligned}
\mathcal{J}(\Theta) &= \log P(\{\mathbf{x}_m\}_{m=1}^M|\Theta) \\
&= \sum_{m=1}^M \log P(\mathbf{x}_m|\Theta) \\
&= \sum_{m=1}^M \log \sum_{y=1}^K P(\mathbf{x}_m, y|\Theta) \\
&= \sum_{m=1}^M \log \sum_{y=1}^K P(y|\Theta)P(\mathbf{x}_m|y, \Theta) \\
&= \sum_{m=1}^M \log \sum_{y=1}^K P(y|\boldsymbol{\pi})P(\mathbf{x}_m|y, \boldsymbol{\theta}_{*|y})
\end{aligned}$$

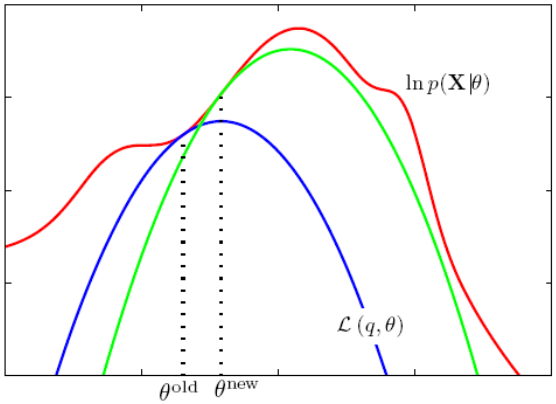
Compared to supervised learning:

$$\begin{aligned}
\mathcal{J}_{NB}(\Theta) &= \log \prod_{m=1}^M P_{\boldsymbol{\pi},\{\boldsymbol{\theta}_k\}}(\mathbf{x}_m, y_m) \\
&= \sum_{m=1}^M \log P_{\boldsymbol{\pi},\{\boldsymbol{\theta}_k\}}(\mathbf{x}_m, y_m) \\
&= \sum_{m=1}^M \log P(y_m|\boldsymbol{\pi})P(\mathbf{x}_m|y_m, \boldsymbol{\theta}_{*|y_m})
\end{aligned}$$

- It's more complicated with a summation inside the log!
- If we try to maximize the marginal log likelihood by setting the gradient to zero, we will find that there is no longer a nice closed form solution, unlike the joint log likelihood with complete data

EM Algorithm: General Idea

- EM is an iterative procedure to maximize the marginal log likelihood $\mathcal{J}(\Theta)$
- It constructs a concave, easy-to-optimize lower bound $\mathcal{J}(\Theta) \geq Q(\Theta, \Theta^t)$, where Θ is the variable and Θ^t is the previous, fixed, parameter
- The lower bound has an interesting property $Q(\Theta^t, \Theta^t) = \mathcal{J}(\Theta^t)$
- Therefore the new parameter Θ^{t+1} that maximizes $Q(\Theta^t, \Theta)$ is guaranteed to have $Q \geq \mathcal{J}(\Theta^t)$. Since Q lower bounds $\mathcal{J}$, we have $\mathcal{J}(\Theta^{t+1}) \geq \mathcal{J}(\Theta^t)$

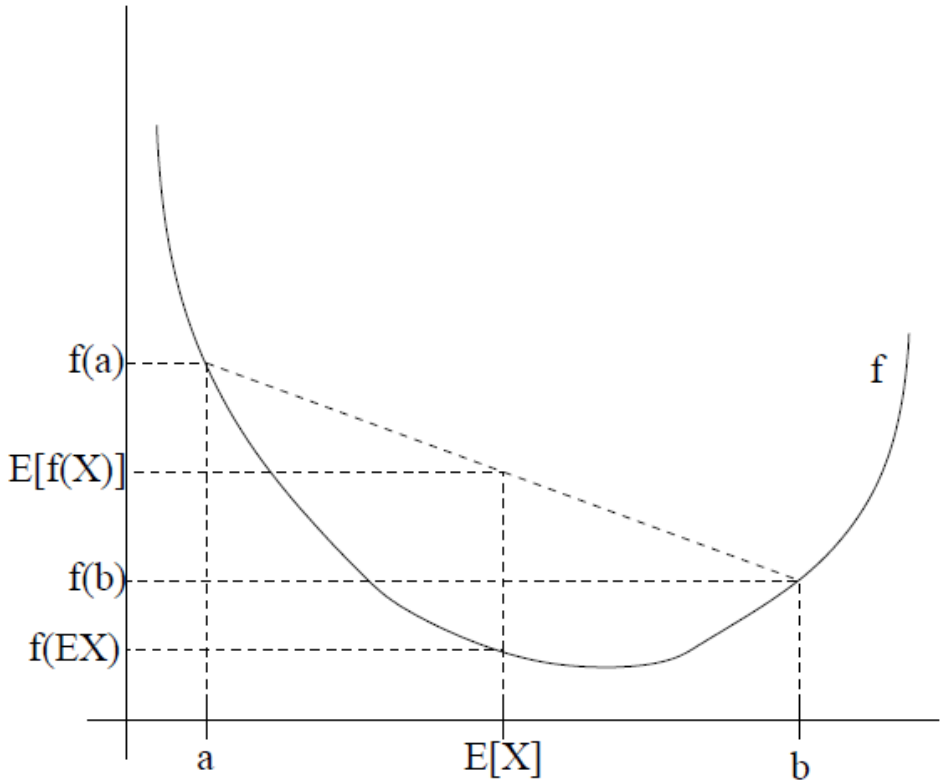


Lower Bound $Q(\Theta, \Theta^t)$

- The lower bound is obtained via Jensens inequality (concavity of log function)

$$\log \sum_i P_i f_i(x) \geq \sum_i P_i \log f_i(x)$$

which holds if the p_i's form a probability distribution



Lower Bound $Q(\Theta, \Theta^t)$ (Cont'd)

- The lower bound is obtained via Jensens inequality (concavity of log function)

$$\log \sum_i P_i f_i(x) \geq \sum_i P_i \log f_i(x)$$

  which holds if the p_i's form a probability distribution

- Then the lower bound can be derived:

$$\begin{aligned}
\mathcal{J}(\Theta^t) &= \sum_{m=1}^{M} \log \sum_{y=1}^{K} P(\mathbf{x}_m, y|\Theta^t) \\
&= \sum_{m=1}^{M} \log \sum_{y=1}^{K} q_{\mathbf{x}_m,y}(\Theta) \frac{P(\mathbf{x}_m,y|\Theta^t)}{q_{\mathbf{x}_m,y}(\Theta)} \\
&\geq \sum_{m=1}^{M} \sum_{y=1}^{K} q_{\mathbf{x}_m,y}(\Theta) \log \frac{P(\mathbf{x}_m,y|\Theta^t)}{q_{\mathbf{x}_m,y}(\Theta)} \\
&\doteq Q(\Theta, \Theta^t)
\end{aligned}$$

  where $\sum_{y=1}^{K} q_{\mathbf{x}_m,y}(\Theta) = 1$ is some distribution

E-step

$$\sum_{m=1}^{M} \log \sum_{y=1}^{K} q_{\mathbf{x}_m,y}(\Theta) \frac{P(\mathbf{x}_m, y|\Theta^t)}{q_{\mathbf{x}_m,y}(\Theta)} \geq \sum_{m=1}^{M} \sum_{y=1}^{K} q_{\mathbf{x}_m,y}(\Theta) \log \frac{P(\mathbf{x}_m, y|\Theta^t)}{q_{\mathbf{x}_m,y}(\Theta)}$$

- To make the bound tight for a particular value of Θ, we need for the step involving Jensens inequality in our derivation above to hold with equality
- For this to be true, we know it is sufficient that that the expectation be taken over a constant-valued random variable $\frac{P(\mathbf{x}_m,y|\Theta^t)}{q_{\mathbf{x}_m,y}(\Theta)} = c$
- This is easily done by choosing $q_{\mathbf{x}_m,y}(\Theta) \propto P(\mathbf{x}_m, y|\Theta^t)$
- Since $\sum_{y=1}^{K} q_{\mathbf{x}_m,y}(\Theta) = 1$, we have (considered as E-step)

$$q_{\mathbf{x}_m,y}(\Theta) = \frac{P(\mathbf{x}_m, y|\Theta^t)}{\sum_{y=1}^{K} P(\mathbf{x}_m, y|\Theta^t)} = P(y|\mathbf{x}_m, \Theta^t)$$

- The equation holds in the inequality iff $q_{\mathbf{x}_m,y} = P(y|\mathbf{x}_m, \Theta^t)$

M-step

- In M-step, we maximize the lower bound

$$\begin{aligned} Q(\Theta^t, \Theta) &= \sum_{m=1}^{M} \sum_{y=1}^{K} q_{\mathbf{x}_m, y} \log \frac{P(\mathbf{x}_m, y|\Theta)}{q_{\mathbf{x}_m, y}} \\ &= \sum_{m=1}^{M} \sum_{y=1}^{K} q_{\mathbf{x}_m, y} \log \frac{P(y_m|\boldsymbol{\pi}) P(\mathbf{x}_m|y_m, \boldsymbol{\theta}_{*|y_m})}{q_{\mathbf{x}_m, y}} \end{aligned}$$

- Now we can set the gradient of Q w.r.t. $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$'s to zero and obtain a closed form solution

$$\begin{aligned} \pi_k &= \frac{\sum_m q_{\mathbf{x}_m, y}}{M} \\ \theta_k^j &= \frac{\sum_m q_{\mathbf{x}_m, y} x_m^j}{\sum_m \sum_{j=1}^{d} q_{\mathbf{x}_m, y} x_m^j} \end{aligned}$$

- Compared to naive Bayes:

$$\begin{aligned} \pi_k &= \frac{|\{y_m = k\}|}{M} \\ \theta_k^j &= \frac{\sum_{m, y_m = k} x_m^j}{\sum_{m, y_m = k} \sum_{j=1}^{d} x_m^j} \end{aligned}$$

EM Algorithm

- Repeat
  - E-step: compute posterior of hidden variables

$$q_{\mathbf{x}_m,y} = P(y|\mathbf{x}_m, \Theta)$$

  - M-step: parameter estimation by maximizing the lower bound

$$\pi_k = \frac{\sum_m q_{\mathbf{x}_m,y}}{M}$$
$$\theta_k^j = \frac{\sum_m q_{\mathbf{x}_m,y} x_m^j}{\sum_m \sum_{j=1}^d q_{\mathbf{x}_m,y} x_m^j}$$

- Until the convergence of the objective function

- Randomly choose K centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
- Repeat
  - Assign $\mathbf{x}_1, \ldots, \mathbf{x}_M$ to their nearest centers to obtain $\hat{y}_m$, respectively
  - Update $\boldsymbol{\mu}_k = \frac{1}{\sum_m I(\hat{y}_m = k)} \sum_m \mathbf{x}_m I(\hat{y}_m = k)$
- Until the clusters no longer change

In practice, K-means is cheaper. We can run multiple times to find good initialization to mixture models.

Convergence of EM Algorithm

- E-step: With $q_{\mathbf{x}_m,y}(\Theta) = P(y|\mathbf{x}_m, \Theta^t)$, the equation holds, which leads

$$Q(\Theta^t, \Theta^t) = \mathcal{J}(\Theta^t)$$

- M-step: Since Θ^{t+1} maximizes $Q(\Theta^t, \Theta)$, we have

$$Q(\Theta^t, \Theta^{t+1}) \geq Q(\Theta^t, \Theta^t) = \mathcal{J}(\Theta^t)$$

- On the other hand, Q is lower bound of $\mathcal{J}$, we have:

$$\mathcal{J}(\Theta^{t+1}) \geq Q(\Theta^t, \Theta^{t+1}) \geq Q(\Theta^t, \Theta^t) = \mathcal{J}(\Theta^t)$$

- This shows EM algorithm always increase the objective function (log likelihood)
- By iterating, we arrive at a local maximum of it

A More General View of EM

- EM is general and applied to joint probability models whenever some random variables are missing
- EM is advantageous when the marginal is difficult to optimize, but the joint is easy
- To be general, consider a joint distribution $P(X, Z|\Theta)$, where X is the collection of observed variables, and Z unobserved variables
- The quantity we want to maximize is the marginal log likelihood

$$\mathcal{J}(\Theta) = \log P(X|\Theta) = \log \sum_{Z} P(X, Z|\Theta)$$

which we assume difficult to optimize

A More General View of EM (Cont'd)

- One can introduce an arbitrary distribution over hidden variables $Q(Z)$

$$\begin{aligned}
\mathcal{J}(\Theta) &= \log P(X|\Theta) = \log \textstyle\sum_Z P(X,Z|\Theta) \\
&= \textstyle\sum_Z Q(Z) \log P(X|\Theta) \\
&= \textstyle\sum_Z Q(Z) \log \frac{P(X|\Theta)Q(Z)P(X,Z|\Theta)}{P(X,Z|\Theta)Q(Z)} \\
&= \textstyle\sum_Z Q(Z) \log \frac{P(X,Z|\Theta)}{Q(Z)} + \sum_Z Q(Z) \log \frac{P(X|\Theta)Q(Z)}{P(X,Z|\Theta)} \\
&= \textstyle\sum_Z Q(Z) \log \frac{P(X,Z|\Theta)}{Q(Z)} + \sum_Z Q(Z) \log \frac{Q(Z)}{P(Z|X,\Theta)} \\
&= F(Q,\Theta) + KL[Q(Z)||P(Z|X,\Theta)]
\end{aligned}$$

- Note $F(Q,\Theta)$ is the right hand side of Jensen's inequality
  - If $KL > 0$, $F(Q,\Theta)$ is a lower bound of $\mathcal{J}(\Theta)$
- First consider the maximization of F on Q with Θ^t fixed
  - $F(Q,\Theta)$ is maximized by $Q(Z) = P(Z|X,\Theta^t)$ since $\mathcal{J}(\Theta)$ is fixed and KL attends its minimum zero (E-Step)
- Next consider the maximization of F on Θ with Q fixed as above
  - Note in this case $F(Q,\Theta) = Q(\Theta^t,\Theta)$ (M-Step)

Variations of EM

- Generalized EM (GEM) finds Θ that improves, but not necessarily maximizes, $F(Q, \Theta) = Q(\Theta, \Theta^t)$ in the M-step. This is useful when the exact M-step is difficult to carry out. Since this is still coordinate ascent, GEM can find a local optimum.
- Stochastic EM: The E-step is computed with Monte Carlo sampling. This introduces randomness into the optimization, but asymptotically it will converge to a local optimum.
- Variational EM: $Q(Z)$ is restricted to some easy-to-compute subset of distributions, for example the fully factorized distributions $Q(Z) = \prod_i Q(z_i)$. In general $P(Z|X, \Theta)$, which might be intractable to compute, will not be in this subset. There is no longer guarantee that variational EM will find a local optimum.
- If $Q(Z|\Phi)$ and $P(X|Z, \Theta)$ can be parameterized by neural networks, variational auto-encoder can be developed (Kingma and Welling (2014)).
    - Note a reparameterization trick should be applied to sample z

Overview

Evaluation

- Accuracy:

$$\begin{aligned} A(f) &= P(f(\mathbf{X}) = Y) \\ &= \sum_{\mathbf{x}\in\mathcal{V}, y\in\mathcal{Y}} P(\mathbf{X}=\mathbf{x}, Y=y) \cdot \begin{cases} 1 & \text{if } f(\mathbf{x}) = y \\ 0 & \text{otherwise} \end{cases} \\ &= \sum_{\mathbf{x}\in\mathcal{V}, y\in\mathcal{Y}} P(\mathbf{X}=\mathbf{x}, Y=y) I(f(\mathbf{x}) = y) \end{aligned}$$

  where P is the true distribution over data

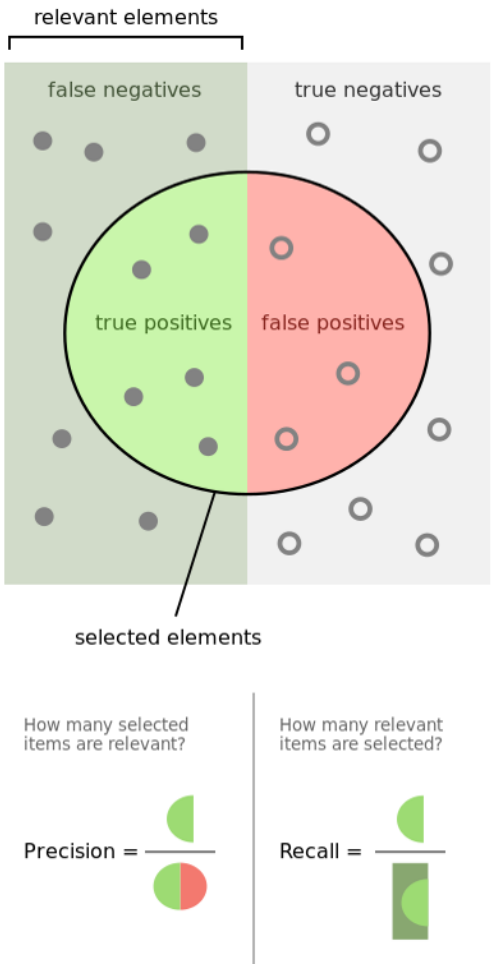
- Error is $1 - A(f)$
- This is estimated using a test dataset $\langle \bar{\mathbf{x}}_1, \bar{y}_1 \rangle, \ldots, \langle \bar{\mathbf{x}}_m, \bar{y}_m \rangle$:

$$\hat{A}(f) = \frac{1}{m} \sum_{i=1}^{m} I(f(\bar{\mathbf{x}}_i) = \bar{y}_i)$$

Issues with Test-Set Accuracy

- Class imbalance: if $P(L = \text{not spam}) = 0.99$, then you can get $\hat{A} \approx 0.99$ by always guessing "not spam"
- Relative importance of classes or cost of error types
- Variance due to the test data

Evaluation in the Two-Class Case



- Precision
  - Fraction of predicted positive documents that are indeed positive, i.e., P(human label $= 1 \mid$ prediction $= 1$)
- Recall
  - Fraction of positive documents that are predicted to be positive, i.e., P(prediction $= 1 \mid$ human label $= 1$)
- F-1 Score:

$$F_1 = 2 \cdot \frac{precesion \cdot recall}{precesion + recall}$$

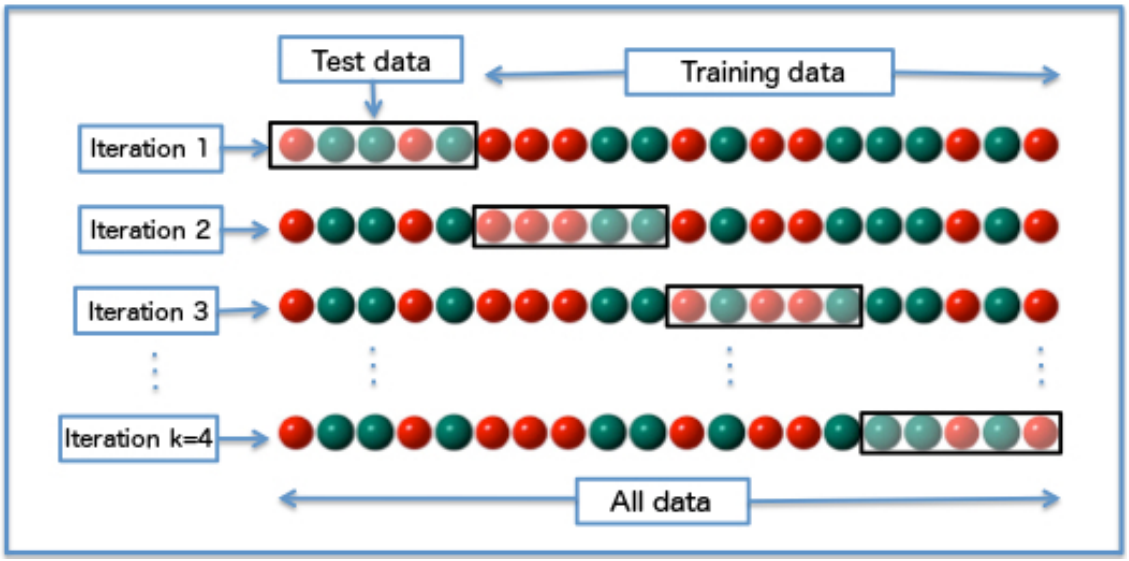
Evaluation in the Multi-Class Case

- Accuracy
- F1
  - Let TP_t, FP_t, FN_t denote the true-positives, false-positives, and false-negatives for the t-th label in label set $\mathcal{L}$ respectively
  - Micro-averaged $F_1 = \frac{2PR}{P+R}$ where $P = \frac{\sum_{t\in\mathcal{L}} TP_t}{\sum_{t\in\mathcal{L}} TP_t+FP_t}$ and $R = \frac{\sum_{t\in\mathcal{L}} TP_t}{\sum_{t\in\mathcal{L}} TP_t+FN_t}$
  - Macro-averaged $F_1 = \frac{1}{|\mathcal{L}|}\sum_{t\in\mathcal{L}} \frac{2P_tR_t}{P_t+R_t}$ where $P_t = \frac{TP_t}{TP_t+FP_t}$ and $R_t = \frac{TP_t}{TP_t+FN_t}$

Actual/ Predicted	Class 1	2	3	4	5	6	7	8	9	10	Total	Recall
Class 1	**9.06**		0.07	0.05	0.01	0.03	0.06	0.59	0.01	0.14	10	90.60
Class 2		**8.20**			0.52	0.04	0.30		0.53	0.42	10	82.00
Class 3	0.03		**9.52**	0.03	0.01	0.02	0.01	0.15	0.02	0.22	10	95.20
Class 4	0.01	0.01	0.01	**9.01**	0.13	0.12	0.52	0.10	0.05	0.06	10	90.10
Class 5		0.48	0.01	0.05	**2.67**	1.87	1.40		2.63	0.90	10	26.70
Class 6		0.11			0.86	**7.75**	0.56		0.10	0.62	10	77.50
Class 7	0.02	0.18		0.32	1.47	1.50	**3.66**	0.11	2.08	0.67	10	36.60
Class 8	0.20		0.05	0.01			0.02	**9.70**		0.03	10	97.00
Class 9		0.39	0.01		1.21	0.11	0.42		**6.84**	1.02	10	68.40
Class 10		0.24	0.13	0.01	0.95	1.01	0.43	0.01	1.85	**5.37**	10	53.70
Total	9.32	9.61	9.80	9.48	7.83	12.45	7.38	10.66	14.11	9.45	100	
Precision	97.21	85.33	97.14	95.04	34.10	62.25	49.59	90.99	48.48	56.83		

Model Estimation and Selection

- k-fold cross-validation
  - Partition all training data into k equal size disjoint subsets
  - Leave one subset for validation and the other k-1 for training
  - Repeat step (2) k times with each of the k subsets used exactly once as the validation data



Statistical Significance

- Suppose we have two classifiers f_1 and f_2
- Is f_1 better? The "null hypothesis," denoted H_0, is that it isn't. But if $\hat{A}(f_1) \gg \hat{A}(f_2)$, we are tempted to believe otherwise
- How much larger must $\hat{A}(f_1)$ be than $\hat{A}(f_2)$ to reject H_0?
- Frequentist view: how (im)probable is the observed difference, given $H_0 = true$?
- Caution: statistical significance is neither necessary nor sufficient for research significance or practical usefulness!

A Hypothesis Test for Text Classifiers

McNemar (1947)

- The null hypothesis: $A(f_1) = A(f_2)$
- Pick significance level α, an "acceptably" high probability of incorrectly rejecting H_0
- Calculate the test statistic, k (explained in the next slide)
- Calculate the probability of a more extreme value of k, assuming H_0 is true; this is the p-value
- Reject the null hypothesis if the p-value is less than α

The p-value is $P(\text{this observation} \,|H_0 \text{ is true})$, not the other way around

McNemar's Test: Details

- Assumptions: independent (test) samples and binary measurements. Count test set error patterns:
- The test is applied to a 2×2 contingency table, which tabulates the outcomes of two tests on a sample of n subjects

	f_1 is incorrect	f_1 is correct	
f_2 is incorrect	a	b	$a+b$
f_2 is correct	c	d	$n \cdot \hat{A}(f_2) = c+d$
	$a+c$	$n \cdot \hat{A}(f_1) = b+d$	n

Evaluate imbalance in the discordant b and c according to Binomial$(k, b+c, \frac{1}{2})$ (The probability of getting k successes in $b+c$ trials)

$$\text{test statistic } k = \min(b, c)$$

$$p - \text{value} = 2\sum_{0}^{k} \text{Binomial}(k; b+c, \frac{1}{2}) = \frac{1}{2^{b+c-1}} \sum_{j=0}^{k} \binom{b+c}{j}$$

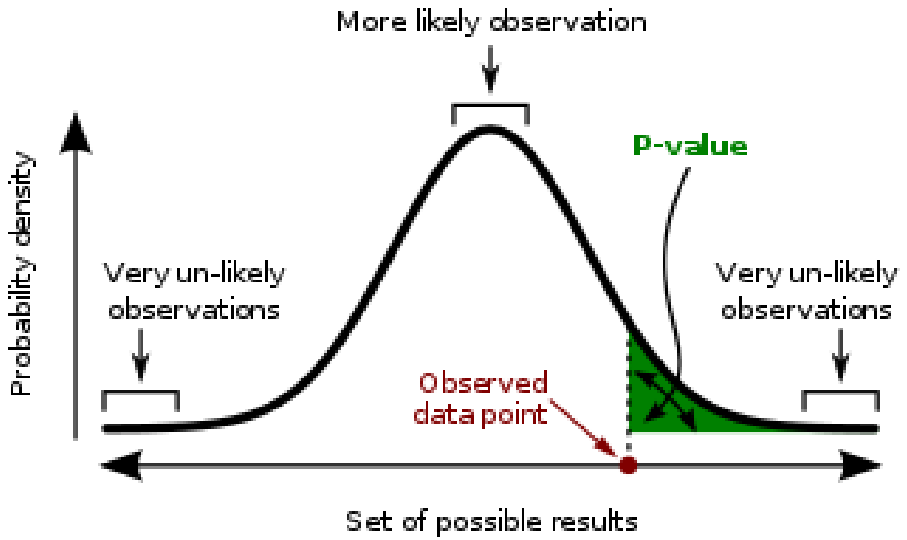
McNemar's Test: Details

Important:

Pr (observation | hypothesis) ≠ Pr (hypothesis | observation)

The probability of observing a result given that some hypothesis is true is *not equivalent* to the probability that a hypothesis is true given that some result has been observed.

Using the p-value as a "score" is committing an egregious logical error: **the transposed conditional fallacy.**



A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

Other Tests

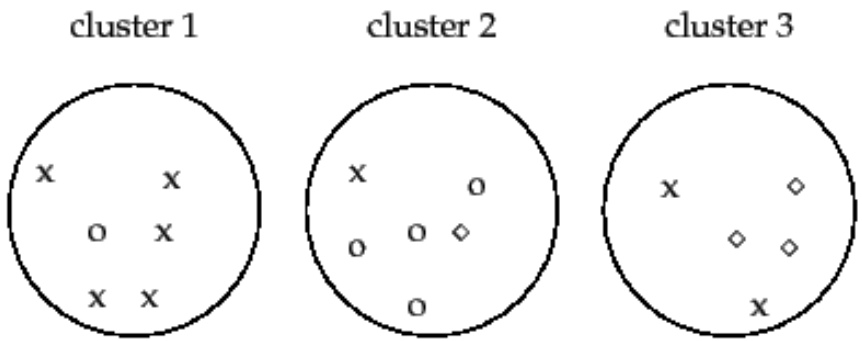
- Different tests make different assumptions
- Sometimes we calculate an interval that would be "unsurprising" under H_0 and test whether a test statistic falls in that interval (e.g., t-test and Wald test)
- In many cases, there is no closed form for estimating p-values, so we use random approximations (e.g., permutation test and paired bootstrap test)
- Read lots more in (Smith (2011)), appendix B

Metrics for Clustering

- Purity between two random variables CAT (category label) and CLS (cluster label) is defined as:

$$\text{Purity (CAT; CLS)} = \frac{1}{n}\sum_{j} \max n_{ij},$$

  - n is the number of documents
  - $n_{i,j}$ is the number of documents in category i as well as in cluster j



► **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ⋄, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

Sometimes Hungarian algorithm is used to match category and cluster $\frac{1}{n}\max\sum_i n_{i,f(i\to j)}$

Metrics for Clustering

- In probability theory and information theory, the mutual information (MI) of two random variables is a measure of the mutual dependence between the two variables.
- More specifically, it quantifies the "amount of information" (in units such as Shannons, more commonly called bits) obtained about one random variable, through the other random variable.
- NMI between two random variables CAT (category label) and CLS (cluster label) is defined as:

$$\mathrm{NMI}(\mathrm{CAT};\ \mathrm{CLS}) = \frac{\mathrm{I}(\mathrm{CAT};\ \mathrm{CLS})}{\sqrt{\mathrm{H}(\mathrm{CAT})\mathrm{H}(\mathrm{CLS})}},$$

where I(CAT; CLS) is the mutual information between CAT and CLS. The entropies H(CAT) and H(CLS) are used for normalizing the mutual information to be in the range of $[0, 1]$.

Metrics for Clustering

- In practice, we made use of the following formulation to estimate the NMI score (Strehl and Ghosh (2002)):

$$\text{NMI} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} n_{i,j} \log\left(\frac{n \cdot n_{i,j}}{n_i \cdot n_j}\right)}{\sqrt{\left(\sum_i n_i \log\frac{n_i}{n}\right)\left(\sum_j n_j \log\frac{n_j}{n}\right)}},$$

  - n is the number of documents
  - n_i and n_j denote the number of documents in category i and cluster j
  - $n_{i,j}$ is the number of documents in category i as well as in cluster j
- The NMI score is 1 if the clustering results perfectly match the category labels, and the score is 0 if data are randomly partitioned.
- The higher the NMI score, the better the clustering quality.

References I

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *ICLR*.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Smith, N. A. (2011). *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles — A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.