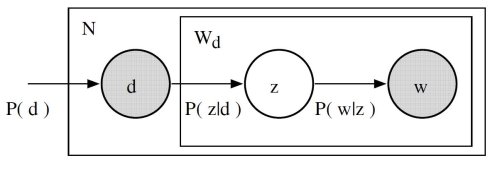
**我们的模型与pLSA、LDA的区别和联系**

最为显著的不同在于，我们的模型结合了两个视图下的信息进行聚类，而pLSA和LDA都只考虑某一个视图下的结果。

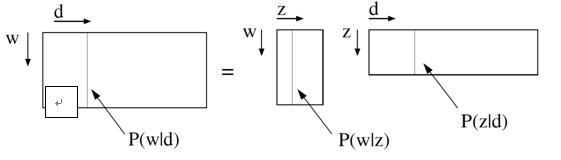
相似之处首先在于，输入数据都是词频，模型中都利用了“话题”进行聚类。

首先来看pLSA。pLSA模型中，将每两个词的共同出现视作条件独立的多项式分布的混合结果。这点与我们的模型是相同的。

PLSA的图



在这一模型中，引入了隐变量z（话题）进行中介。PLSA假设，在生成一篇文档的过程中，我们首先根据分布P(z | d)选取恰当的话题，然后按照这一话题下词的分布P(w | z)来选取一个词。对于这篇文章里的所有词，都要按照上述两个步骤来产生。



在PLSA中，目标是找到特定的话题-词分布P(w|z)和对应的topic混合比例P(z|d)。在参数估计中使用EM算法。

PLSA与LDA的区别在于，对文档-话题这一级，PLSA把这一级的所有变量都看作模型的参数，即有多少文档那么就有多少模型的参数；而LDA引入了一个超参数，对文档-话题这一级别进行建模，这样无论文档有多少，最外层模型显露出来的就只有一个超参数（对于文档-话题）

而在我们的模型中，模型的输入是按样本文档估计得到的两个视图下单词的共现概率，与文档个体无关。文档若具有代表性，则估计值就更加准确，将便于之后进行分类。如果说在LDA、PLSA模型中，关于话题的信息保存在文档之中（单篇文档将不能得到topic），那么在我们的模型中，关于一个视图的话题信息保存在另一个视图的话题信息之中（两者缺一不可，且样本文档一一对应）。换言之，两个视图由于用于描述同一些客体而具有关联性，这使得我们知道二者中的一个，就能够估计出另外一个。

相应地，我们有如下的生成过程。

1. 首先，选择一个文档来生成。这一文档有它自己的p(w, z | d)
2. 按照两个视图下话题的共现概率p(w, z | d)选择一对话题(w, z)，亦即第一个视图下选择话题w，第二个视图中选择话题z。
3. 按照第一个视图下词与话题的条件分布p(word | topic1 = w)选择一个词，作为这一视图下文档的一个词。
4. 按照第二个视图下词与话题的条件分布p(word | topic2 = z)选择一个词，作为这一视图下文档的一个词。
5. 重复上述步骤b) ~ d)，直到所有词都生成完毕。由于我们只考虑词频，故文档长度视为一致。

由于话题的特性，话题和词语的关系要来得紧密，不随文档发生变化，所以在这一过程中，分布p(word |topic1,2)同文档无关，可视为保持不变。在数学期望E[XY’] = P(word | topic1) \* P(w, z) \* P(word | topic2)中，P(w, z)体现了文档样本的平均值。而对每个文档来说，有它自己独立的那个P(w, z | d)，对总的P(w, z)有所贡献。

在进行聚类的时候，我们采取的方法是，按照文档的XY’来计算文档的P(w, z | d)，最大值所对应的一对话题w, z就认为是文档所从属的类别。这些都同只涉及一个视图信息的PLSA、LDA不同。

实现上，为了减少运算量，我们需要给定一个比较合适的对话题-词分布p(word | topic1,2)的初值。这一初始值可以由在两个视图上分别运行的PLSA、LDA给出，也可由kmeans根据E与E’，或原始输入给出。

P(d)

ND

P(w, z | d)

Wd

Wd

P(word |topic2=w)

P(word |topic1=z)