Naive Bayes for Image Classification

The naive Bayes method assumes that the probability that an image belongs to class C_i depends on how likely each patch p_j is to belong to class C_i. Moreover, patches are assumed to be independent, so we have

$$p(p_1, p_2, \ldots | C_i) = \prod_j p(p_j | C_i).$$

Applying Bayes's rule,

$$p(C_i | p_1, p_2, \ldots) = p(p_1, p_2, \ldots | C_i) \; \frac{p(C_i)}{p(p_1, p_2, \ldots)}.$$

We will use these probabilities in a Maximum A Posteriori (MAP) classifier, in which we simply take the largest class probability as the classification result. So, we don't really care about the normalization term $p(p_1, p_2, \ldots)$. Moreover, we will take the class priors $p(C_i)$ to be equal. So, we can write:

$$p(C_i | p_1, p_2, \ldots) \propto \prod_j p(p_j | C_i),$$

and the estimated class $\hat{C}$ is

$$\hat{C} = \arg\max_i \prod_j p(p_j | C_i).$$

The probabilities $p(p_j|C_i)$ are just proportional to the counts of the codewords in the training data for each class, and we form the product over all the detected patches in our test image. (Multiple detections of the same patch count multiple times, and their counts must be multiplied in the same number of times that they were detected.)

There is one wrinkle to the above. If a particular patch was never observed in the training data for a particular class, its probability would be zero, which would set the overall estimated probability to zero. To account for this, we use a technique called "Laplace smoothing", in which we pretend that, during training, we had one additional observation of each patch, in each class:

$$p(p_j | C_i) \propto 1 + \text{count}\big(p_j \in \textit{training}(C_i)\big).$$