The ‘summation hack’ as an outlier model

Thomas P. Minka

August 22, 2003 (revised)

Abstract

The ‘summation hack’ is the ad-hoc replacement of a product by a sum in a probabilistic expression. This hack is usually explained as a device to cope with outliers, with no formal derivation. This note shows that the hack does make sense probabilistically, and can be best thought of as replacing an outlier-sensitive likelihood with an outlier-tolerant one. This interpretation exposes the hack as an assumption about the outliers, allowing us to determine when it makes sense to use the hack.

1 Outliers in discriminative models

One way to formalize supervised learning is to cast it as a maximum-likelihood problem in a discriminative model. That is, a model which expresses the probability of a class label c directly as a function of the measured x. The probability of a dataset $D = \{(c_i, x_i), i = 1, ..., n\}$ is given by

$$p(D|\theta) = \prod_{i=1}^{n} p(c_i|x_i, \theta) \tag{1}$$

This is the approach used in logistic regression, for example. However, some practitioners prefer the following objective instead:

$$p(D|\theta) \approx \sum_{i=1}^{n} p(c_i|x_i, \theta) \tag{2}$$

It is often called the “minimum classification error” objective (Saul & Rahim, 1999), with no probabilistic justification. In fact, this ‘summation hack’ has a natural interpretation as an outlier model (but a very extreme one).

The model is this: to label a point, you first flip a coin with heads probability $(1 - e)$. If the coin turns up heads, you draw a label from $p(c_i|x_i, \theta)$. If the coin turns up tails, you draw a label from a uniform distribution (all labels equally likely). These latter occurrences are the outliers. Let C be the number of labels. Then the probability of drawing class c_i for x_i under this process is

$$p'(c_i|x_i, \theta) = e/C + (1 - e)p(c_i|x_i, \theta) \tag{3}$$

Now take a product of these likelihoods, as prescribed by (1). A Taylor expansion in $(1-e)$ gives

$$\prod_{i=1}^{n} p'(c_i|x_i,\theta) \quad = \quad (e/C)^n + (e/C)^{n-1}(1-e)\sum_{i=1}^{n} p(c_i|x_i,\theta) + O((1-e)^2) \tag{4}$$

If e is very close to 1 (a lot of outliers), then only the first two terms matter, and maximizing (4) over θ is equivalent to maximizing (2).

If your outlier assumption is not so extreme, you might want to set e to a reasonable value and work with (3) directly. This is the approach used in Minka (2001), chapter 5.

There is another outlier model which yields (2) directly, not as a limit. The model says that exactly one label in D was sampled from $p(c|x,\theta)$, and the others were sampled uniformly (they are all outliers). Under this model, the labels are no longer independent. Let i indicate which point is the inlier, chosen randomly from 1 to n. Then the probability of the data is

$$\begin{aligned} p(D|\theta) &= \frac{1}{n}\sum_{i=1}^{n} p(D|i,\theta) && (5)\\ &= \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{n} p(c_j|x_j,i,\theta) && (6)\\ &= \frac{1}{n}\sum_{i=1}^{n} p(c_i|x_i,\theta)(1/C)^{n-1} && (7) \end{aligned}$$

2 Outliers in generative models

Unsupervised learning can be cast as a maximum-likelihood problem with a generative model, i.e. a probability model for measurements x. The probability of a dataset $D = \{x_i, i = 1, ..., n\}$ is given by

$$p(D|\theta) \quad = \quad \prod_{i=1}^{n} p(x_i|\theta) \tag{8}$$

However, some practitioners prefer this objective, because it is more tolerant of outliers:

$$p(D|\theta) \quad = \quad \sum_{i=1}^{n} p(x_i|\theta) \tag{9}$$

To illustrate the difference, considering estimating the mean m of a Gaussian. Using (8) leads to the sample mean as the estimate of m. Using (9) leads to the following fixed-point equation for m:

$$m \quad = \quad \frac{\sum_i x_i p(x_i|m)}{\sum_i p(x_i|m)} \tag{10}$$

By iterating this equation, we repeatedly take the mean of a local window of the data, until a local maximum is reached. This "mean-shift" algorithm is very tolerant of outliers, and can be used to extract clusters one at a time (Comaniciu & Meer, 1997).

To explain (9), consider another coin-flip model: to sample a point x, you first flip a coin with heads probability $(1-e)$. If the coin turns up heads, you draw x from $p(x|\theta)$. If the coin turns up tails, you draw x from a uniform distribution ($1/A$ where A is the area of x's domain). This leads to the modified model

$$p'(x|\theta) \quad = \quad e/A + (1-e)p(x|\theta) \tag{11}$$

Now take a product of these likelihoods, as prescribed by (8). A Taylor expansion in $(1-e)$ gives

$$\prod_{i=1}^{n} p'(x_i|\theta) \quad = \quad (e/A)^n + (e/A)^{n-1}(1-e)\sum_{i=1}^{n} p(x_i|\theta) + O((1-e)^2) \tag{12}$$

If e is very close to 1 (a lot of outliers), then only the first two terms matter, and maximizing (12) over θ is equivalent to maximizing (9).

The alternative "one inlier" model can also be used, to get (9) exactly. In that model, the measurements are no longer considered independent.

3 Outliers in individual dimensions of Naive Bayes models

The outlier model in the previous section assumes that a data point is entirely an outlier or entirely an inlier. But if the measurement $\mathbf{x}$ is vector-valued, it can happen that some of the dimensions are corrupted, while others are not. It turns out that there is a 'summation hack' for naive Bayes models which can handle this situation as well.

Naive Bayes is a generative model for vector-valued measurements, where the measurements are conditionally independent:

$$p(\mathbf{x}|\theta) \quad = \quad \prod_{k=1}^{K} p(x_k|\theta) \tag{13}$$

However, this probability is sensitive to outliers in individual dimensions. The 'summation hack' in this case is

$$p(\mathbf{x}|\theta) \quad \approx \quad \sum_{k=1}^{K} p(x_k|\theta) \tag{14}$$

To explain it, once again consider a coin-flip model: to sample a feature x_k, you first flip a coin with heads probability $(1-e)$. If the coin turns up heads, you draw x_k from $p(x_k|\theta)$. If the coin turns up tails, you draw x_k from a uniform distribution ($1/A$ where A is the area of x_k's domain). Note that all features must have the same domain in order for this to work. This leads to the modified model

$$p'(x_k|\theta) \quad = \quad e/A + (1-e)p(x_k|\theta) \tag{15}$$

Now take a product of these likelihoods, as prescribed by (13). A Taylor expansion in $(1-e)$ gives

$$\prod_{k=1}^{K} p'(x_k|\theta) \quad = \quad (e/A)^K + (e/A)^{K-1}(1-e)\sum_{k=1}^{K} p(x_k|\theta) + O((1-e)^2) \tag{16}$$

If e is very close to 1 (a lot of outliers), then only the first two terms matter, which are monotonic in (14).

The alternative "one inlier" model can also be used, to get (14) exactly. But then it is no longer a Naive Bayes model, because the dimensions are coupled.

4 Outliers in Naive Bayes classification

Naive Bayes is often used for classification using Bayes' rule:

$$p(c|\mathbf{x}) \quad = \quad \frac{p(\mathbf{x}|c)p(c)}{\sum_c p(\mathbf{x}|c)p(c)} \tag{17}$$

Some people have replaced this formula with the following 'summation hack' which only looks at the per-dimension posteriors:

$$p(c|\mathbf{x}) \quad = \quad \frac{1}{K}\sum_{k=1}^{K} p(c|x_k) \tag{18}$$

See for example Joachims (1997) and Schiele & Crowley (2000) (section 8).

This formula can be explained by the "one inlier" assumption (which makes it no longer a strict Naive Bayes model). Let k be the one dimension of $\mathbf{x}$ which is the inlier. This dimension is sampled from $p(x_k|c)$ while the other dimensions are sampled from the marginal $p(x_j) = \sum_c p(x_j|c)p(c)$. The variable k is independent of the class c and has uniform prior. Thus we have

$$
\begin{aligned}
p(\mathbf{x}) &= \prod_{j=1}^{K} p(x_j) && (19)\\
p(\mathbf{x}|k,c) &= \frac{p(x_k|c)}{p(x_k)} p(\mathbf{x}) && (20)\\
p(\mathbf{x}|c) &= \frac{1}{K}\sum_{k=1}^{K} \frac{p(x_k|c)}{p(x_k)} p(\mathbf{x}) && (21)\\
p(c|\mathbf{x}) &= \frac{p(\mathbf{x}|c)p(c)}{p(\mathbf{x})} = \frac{1}{K}\sum_{k=1}^{K} \frac{p(x_k|c)p(c)}{p(x_k)} = \frac{1}{K}\sum_{k=1}^{K} p(c|x_k) && (22)
\end{aligned}
$$

What makes this generative model unusual is that all classes are involved in generating each point. Because this is no longer a Naive Bayes model, the correct maximum-likelihood estimate of $p(x_k|c)$ is **not** the Naive Bayes estimate, since k is unknown in the training data. Instead we have to use EM to train the model, which is usually not considered when applying the summation hack (18).

References

Comaniciu, D., & Meer, P. (1997). Robust analysis of feature spaces: Color image segmentation. *CVPR*.

Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. *ICML*. `http://www.cs.cornell.edu/People/tj/`.

Minka, T. P. (2001). *A family of algorithms for approximate Bayesian inference*. Doctoral dissertation, Massachusetts Institute of Technology. `http://www.stat.cmu.edu/~minka/papers/ep/`.

Saul, L. K., & Rahim, M. G. (1999). Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, *8*, 115–125. `http://www.cis.upenn.edu/~lsaul/papers.html`.

Schiele, B., & Crowley, J. L. (2000). Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, *36*, 31–50. `http://www-white.media.mit.edu/cgi-bin/tr_pagemaker#TR453`.