Prediction by partial matching

For professional Super Smash Bros. Melee player known as PPMD, see Kevin Nanney.

Prediction by partial matching (**PPM**) is an adaptive statistical data compression technique based on context modeling and prediction. PPM models use a set of previous symbols in the uncompressed symbol stream to predict the next symbol in the stream. PPM algorithms can also be used to cluster data into predicted groupings in cluster analysis.

1 Theory

Predictions are usually reduced to symbol rankings. The number of previous symbols, *n*, determines the order of the PPM model which is denoted as PPM(*n*). Unbounded variants where the context has no length limitations also exist and are denoted as *PPM**. If no prediction can be made based on all n context symbols a prediction is attempted with $n - 1$ symbols. This process is repeated until a match is found or no more symbols remain in context. At that point a fixed prediction is made.

Much of the work in optimizing a PPM model is handling inputs that have not already occurred in the input stream. The obvious way to handle them is to create a "never-seen" symbol which triggers the escape sequence. But what probability should be assigned to a symbol that has never been seen? This is called the zero-frequency problem. One variant uses the Laplace estimator, which assigns the "never-seen" symbol a fixed pseudocount of one. A variant called PPMD increments the pseudocount of the "never-seen" symbol every time the "never-seen" symbol is used. (In other words, PPMD estimates the probability of a new symbol as the ratio of the number of unique symbols to the total number of symbols observed).

2 Implementation

PPM compression implementations vary greatly in other details. The actual symbol selection is usually recorded using arithmetic coding, though it is also possible to use Huffman encoding or even some type of dictionary coding technique. The underlying model used in most PPM algorithms can also be extended to predict multiple symbols. It is also possible to use non-Markov modeling to either replace or supplement Markov modeling. The symbol size is usually static, typically a single byte, which makes generic handling of any file format easy.

Published research on this family of algorithms can be found as far back as the mid-1980s. Software implementations were not popular until the early 1990s because PPM algorithms require a significant amount of RAM. Recent PPM implementations are among the best-performing lossless compression programs for natural language text.

Trying to improve PPM algorithms led to the PAQ series of data compression algorithms.

A PPM algorithm, rather than being used for compression, is used to increase the efficiency of user input in the alternate input method program Dasher.

3 References

- Cleary, J.; Witten, I. (April 1984). "Data Compression Using Adaptive Coding and Partial String Matching". *IEEE Trans. Commun.* **32** (4): 396–402. doi:10.1109/TCOM.1984.1096090.
- Moffat, A. (November 1990). "Implementing the PPM data compression scheme". *IEEE Trans. Commun.* **38** (11): 1917–1921. doi:10.1109/26.61469.
- Cleary, J. G.; Teahan, W. J.; Witten, I. H. (1995). "Unbounded length contexts for PPM". In Storer, J. A.; Cohn, M. *Proceedings DCC '95*. Data Compression Conference: 28-30 Mar 1995, Snowbird, UT. IEEE Computer Society Press. pp. 52–61. doi:10.1109/DCC.1995.515495. ISBN 0-8186-7012-6.
- C. Bloom, Solving the problems of context modeling.
- W.J. Teahan, Probability estimation for PPM.
- SchüRmann, T.; Grassberger, P. (September 1996). "Entropy estimation of symbol sequences". *Chaos* **6** (3): 414–427. doi:10.1063/1.166191. PMID 12780271.

4 See also

- Language model
- N-gram

5 External links

- Suite of PPM compressors with benchmarks
- BICOM, a bijective PPM compressor
- “Arithmetic Coding + Statistical Modeling = Data Compression”, Part 2
- (Russian) PPMd compressor by Dmitri Shkarin
- PPM algorithm implementation (source code) by René Puchinger

6 Text and image sources, contributors, and licenses

6.1 Text

- **Prediction by partial matching** *Source:* https://en.wikipedia.org/wiki/Prediction_by_partial_matching?oldid=678332639 *Contributors:* Bryan Derksen, Michael Hardy, Kku, Ciphergoth, Babbage, Tobias Bergemann, DavidCary, ShaunMacPherson, Inkling, Gzornenplatz, Neilc, OverlordQ, Superborsuk, M.e, Moxfyre, Sladen, Antaeus Feldspar, CanisRufus, Matt Mahoney, Roboto de Ajvol, YurikBot, Piet Delport, DmitriyV, Incnis Mrsi, Betacommand, Thumperward, Chlewbot, LouScheffer, Damate, Sixstone~enwiki, Requestion, A3nm, Alleborgo, Speck-Made, Gigacephalus, Schuermann~enwiki, Addbot, SpBot, Ptbotgourou, Ant diaz, Xqbot, FrescoBot, EmausBot, John Cline, Valdmann, Dexbot, Prisencolin and Anonymous: 22

6.2 Images

- **File:Symbol_template_class.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/5/5c/Symbol_template_class.svg *License:* Public domain *Contributors:* ? *Original artist:* ?

6.3 Content license

-