Self-normalised distance with don't cares

Peter Clifford[1] Raphaël Clifford[2]
clifford@cs.bris.ac.uk

[1]Dept. of Statistics, University of Oxford, UK
clifford@stats.ox.ac.uk

[2]Bristol University, Dept. of Computer Science, UK
clifford@cs.bris.ac.uk

Outline

Self-normalised distance with don't cares

- Consider matching a small image segment or template within a larger scene.
- The intensity or brightness of an image occurrence is unknown. The larger scene may have parts with very different dynamic ranges
- Impossible to normalise the whole image according to one rule. One part may be in bright sunshine and another in shadow.
- Parts of either image contain *don't care* symbols. For example the background may not be important when looking for an object.
- Heuristic techniques are well known and used in practice.
- Which problems can we solve exactly in $O(n \log m)$ time?

The standard unnormalised problem

All problems start with a pattern p and a text t and we restrict ourselves to one dimension for notational simplicity.

- To perform pattern matching, a common tool is to calculate the cross-correlation (or equivalently cross-convolution) of a pattern with a text.
- An important property of the FFT is that all the inner-products,

$$p \otimes t \stackrel{\text{def}}{=} \left(\sum_{j=0}^{m-1} p_j t_{i+j}, \;\; 0 \leq i \leq n-m \right)$$

  can be calculated accurately and efficiently in $O(n \log m)$ time
- The squared L_2 norm,

$$\sum_{j=0}^{m-1} (p_j - t_{i+j})^2 = \sum_{j=0}^{m-1} (p_j^2 - 2 p_j t_{i+j} + t_{i+j}^2)$$

  can therefore be calculated for all i in the same time complexity.

The standard unnormalised problem

All problems start with a pattern p and a text t and we restrict ourselves to one dimension for notational simplicity.

- To perform pattern matching, a common tool is to calculate the cross-correlation (or equivalently cross-convolution) of a pattern with a text.
- An important property of the FFT is that all the inner-products,

$$p \otimes t \stackrel{\text{def}}{=} \left(\sum_{j=0}^{m-1} p_j t_{i+j}, \quad 0 \leq i \leq n-m \right)$$

  can be calculated accurately and efficiently in $O(n \log m)$ time
- The squared L_2 norm,

$$\sum_{j=0}^{m-1} (p_j - t_{i+j})^2 = \sum_{j=0}^{m-1} (p_j^2 - 2 p_j t_{i+j} + t_{i+j}^2)$$

  can therefore be calculated for all i in the same time complexity.

The standard unnormalised problem

All problems start with a pattern p and a text t and we restrict ourselves to one dimension for notational simplicity.

- To perform pattern matching, a common tool is to calculate the cross-correlation (or equivalently cross-convolution) of a pattern with a text.
- An important property of the FFT is that all the inner-products,

$$p \otimes t \stackrel{\text{def}}{=} \left(\sum_{j=0}^{m-1} p_j t_{i+j}, \quad 0 \leq i \leq n-m \right)$$

  can be calculated accurately and efficiently in $O(n \log m)$ time
- The squared L_2 norm,

$$\sum_{j=0}^{m-1} (p_j - t_{i+j})^2 = \sum_{j=0}^{m-1} (p_j^2 - 2 p_j t_{i+j} + t_{i+j}^2)$$

  can therefore be calculated for all i in the same time complexity.

The standard unnormalised problem with don't cares

In this version we still want to calculate the squared L_2 norm but we assume that all symbols have distance 0 from the don't care symbol.

- Define a new string p' with $p'_j = 0$ if $p_j = \phi$, and $p'_j = 1$ otherwise. Similarly, define $t'_i = 0$ if $t_i = \phi$, and 1 otherwise.
- We can now express the standard unnormalised problem with don't cares by

$$\sum_{j=0}^{m-1} p'_j t'_{i+j} (p_j - t_{i+j})^2$$

- This sum equals zero when either the pattern or text symbol is a don't care.

The standard unnormalised problem with don't cares

In this version we still want to calculate the squared L_2 norm but we assume that all symbols have distance 0 from the don't care symbol.

- Define a new string p' with $p'_j = 0$ if $p_j = \phi$, and $p'_j = 1$ otherwise. Similarly, define $t'_i = 0$ if $t_i = \phi$, and 1 otherwise.
- We can now express the standard unnormalised problem with don't cares by

$$\sum_{j=0}^{m-1} p'_j t'_{i+j} (p_j - t_{i+j})^2$$

- This sum equals zero when either the pattern or text symbol is a don't care.

The standard unnormalised problem with don't cares

In this version we still want to calculate the squared L_2 norm but we assume that all symbols have distance 0 from the don't care symbol.

- Define a new string p' with $p'_j = 0$ if $p_j = \phi$, and $p'_j = 1$ otherwise. Similarly, define $t'_i = 0$ if $t_i = \phi$, and 1 otherwise.
- We can now express the standard unnormalised problem with don't cares by

$$\sum_{j=0}^{m-1} p'_j t'_{i+j} (p_j - t_{i+j})^2$$

- This sum equals zero when either the pattern or text symbol is a don't care.

Exact matching with don't cares

A brief digression into exact matching with don't cares.

1. For binary alphabets it is well known that you can solve the problem by setting p' and t' so that $a = -1$, $b = 1$ and $\phi = 0$. Calculate $p' \otimes t'$.
2. Set p'' and t'' so that $a = 1, b = 1$ and $\phi = 0$. Calculate $p'' \otimes t''$
3. Find where $p' \otimes t'[i] = p'' \otimes t''[i]$

Example

$p = ab?ab$ and text $t = b?bbabba$
$p' = -1, 1, 0, -1, 1$ and $t' = 1, 0, 1, 1, -1, 1, 1, -1$
$p' \otimes t' = -3, 3, 0, -4$
Compare to $p'' \otimes t'' = 3, 3, 4, 4$

Exact matching with don't cares

A brief digression into exact matching with don't cares.

1. For binary alphabets it is well known that you can solve the problem by setting p' and t' so that $a = -1$, $b = 1$ and $\phi = 0$. Calculate $p' \otimes t'$.
2. Set p'' and t'' so that $a = 1, b = 1$ and $\phi = 0$. Calculate $p'' \otimes t''$
3. Find where $p' \otimes t'[i] = p'' \otimes t''[i]$

Example

$p = ab?ab$ and text $t = b?bbabba$
$p' = -1, 1, 0, -1, 1$ and $t' = 1, 0, 1, 1, -1, 1, 1, -1$
$p' \otimes t' = -3, 3, 0, -4$
Compare to $p'' \otimes t'' = 3, 3, 4, 4$

Exact matching with don't cares -non-binary alphabets

Define a new string p' with $p'_j = 0$ if $p_j = \phi$, and $p'_j = 1$ otherwise.
Similarly, define $t'_i = 0$ if $t_i = \phi$, and 1 otherwise.
We have an exact match with don't cares if and only if

$$\sum_{j=0}^{m-1} p'_j t'_{i+j} (p_j - t_{i+j})^2 = 0$$

Therefore,

$$\sum_{j=0}^{m-1} (p'_j p_j^2 t'_{i+j} - 2 p'_j p_j t'_{i+j} t_{i+j} + p'_j t'_{i+j} t_{i+j}^2) = 0$$

This takes $O(n \log m)$ to compute for all i using 3 cross-correlations.

- $O(n \log m \log |\Sigma|)$ deterministic [Fischer and Paterson (1974)]
- $O(n \log n)$ randomised [Indyk - 1998]
- $O(n \log m)$ randomised [Kalai - 2002]
- $O(n \log m)$ deterministic [Cole and Hariharan - 2002]
- Even simpler $O(n \log m)$ deterministic [Clifford and C. - 2007]

Exact matching with don't cares -non-binary alphabets

Define a new string p' with $p'_j = 0$ if $p_j = \phi$, and $p'_j = 1$ otherwise.
Similarly, define $t'_i = 0$ if $t_i = \phi$, and 1 otherwise.
We have an exact match with don't cares if and only if

$$\sum_{j=0}^{m-1} p'_j t'_{i+j} (p_j - t_{i+j})^2 = 0$$

Therefore,

$$\sum_{j=0}^{m-1} (p'_j p_j^2 t'_{i+j} - 2 p'_j p_j t'_{i+j} t_{i+j} + p'_j t'_{i+j} t_{i+j}^2) = 0$$

This takes $O(n \log m)$ to compute for all i using 3 cross-correlations.

- $O(n \log m \log |\Sigma|)$ deterministic [Fischer and Paterson (1974)]
- $O(n \log n)$ randomised [Indyk - 1998]
- $O(n \log m)$ randomised [Kalai - 2002]
- $O(n \log m)$ deterministic [Cole and Hariharan - 2002]
- Even simpler $O(n \log m)$ deterministic [Clifford and C. - 2007]

Exact matching with don't cares (accuracy)

It is useful to compare the two most recent solutions. First, that of Cole and Hariharan.

1. Don't cares in p and t are replaced by 0 and non-don't cares by 1 to give p' and t'.
2. Don't cares in p and t are replaced by 0 and any number a by $1/a$ to give p'' and t''.
3. Calculate $p'p \otimes t''$, $p' \otimes t'$ and $p'' \otimes t't$. An exact match at location i is declared if

$$((p'p) \otimes t'' - 2p' \otimes t' + p'' \otimes (t't))[i] = 0.$$

4. This quantity can be as small as $1/(N(N-1))$ when it is non-zero, which indicates how much precision is required.

Exact matching with don't cares (accuracy)

In simple terms the algorithm of Cole and Hariharan can be thought of as testing whether

$$\sum \frac{p'_j t'_{i+j}(p_j - t_{i+j})^2}{p_j t_{i+j}} = \sum (p'_j p_j t''_{i+j} - 2p'_j t'_{i+j} + p''_j t'_{i+j} t_{i+j}) = 0$$

A simple alternative is to test whether

$$\sum p'_j t'_{i+j}(p_j - t_{i+j})^2 = \sum (p'_j p_j^2 t'_{i+j} - 2p'_j p_j t'_{i+j} t_{i+j} + p'_j t'_{i+j} t_{i+j}^2)$$

equals 0.
The advantage for integer strings, is that all calculations are now in integers and we need only distinguish 0 from other integer values.

Self-normalised shift distance

Our first form of self-normalisation is called the *shift* distance.

- Crucially, the optimal shift can be different at every position i.

Therefore, at each position i we will calculate

$$A_i = \min_{\alpha} \sum (\alpha + p_j - t_{i+j})^2 p'_j t'_{i+j}.$$

To find A_i we differentiate with respect to α and obtain the minimising value as

$$\widehat{\alpha}_i = \frac{\sum (t_{i+j} - p_j) p'_j t'_{i+j}}{\sum p'_j t'_{i+j}} = \frac{((tt') \otimes p' - (pp') \otimes t')[i]}{(p' \otimes t')[i]}. \quad (1)$$

Substituting $\alpha = \widehat{\alpha}$, expanding and collecting terms we find A_i is the ith element of

$$(t^2 t') \otimes p' - 2(tt') \otimes (pp') + t' \otimes (p^2 p') - \frac{((tt') \otimes p' - t' \otimes (pp'))^2}{p' \otimes t'}$$

Self-normalised shift distance

Our first form of self-normalisation is called the *shift* distance.

- Crucially, the optimal shift can be different at every position i.

Therefore, at each position i we will calculate

$$A_i = \min_{\alpha} \sum (\alpha + p_j - t_{i+j})^2 p'_j t'_{i+j}.$$

To find A_i we differentiate with respect to α and obtain the minimising value as

$$\widehat{\alpha}_i = \frac{\sum (t_{i+j} - p_j) p'_j t'_{i+j}}{\sum p'_j t'_{i+j}} = \frac{((tt') \otimes p' - (pp') \otimes t')[i]}{(p' \otimes t')[i]}. \tag{1}$$

Substituting $\alpha = \widehat{\alpha}$, expanding and collecting terms we find A_i is the ith element of

$$(t^2 t') \otimes p' - 2(tt') \otimes (pp') + t' \otimes (p^2 p') - \frac{((tt') \otimes p' - t' \otimes (pp'))^2}{p' \otimes t'}$$

Self-normalised shift-scale distance

The minimised *shift-scale* distance at position i can be written as

$$B_i = \min_{\alpha,\beta} \sum (\alpha + \beta p_j - t_{i+j})^2 p'_j t'_{i+j}.$$

The minimising values of α and β can be obtained by differentiating $\sum(\alpha + \beta p_j - t_{i+j})^2 p'_j t'_{i+j}$. Substituting and collecting terms we find

$$B = \left((t^2 t') \otimes p'\right) - \left[(tt') \otimes (p')\right]^2 / (t' \otimes p') - T^2/U,$$

where

$$T = \left((tt') \otimes (pp')\right) - \left((tt') \otimes p'\right)\left(t' \otimes (pp')\right) / \left(t' \otimes p'\right)$$

and

$$U = \left(t' \otimes (p^2 p')\right) - \left(t' \otimes (pp')\right)^2 / \left(t' \otimes p'\right).$$

Notice these are the same 6 cross-correlations as before.

Self-normalised shift-scale distance

The minimised *shift-scale* distance at position i can be written as

$$B_i = \min_{\alpha,\beta} \sum (\alpha + \beta p_j - t_{i+j})^2 p'_j t'_{i+j}.$$

The minimising values of α and β can be obtained by differentiating $\sum(\alpha + \beta p_j - t_{i+j})^2 p'_j t'_{i+j}$. Substituting and collecting terms we find

$$B = \left((t^2 t') \otimes p'\right) - \left[(tt') \otimes (p')\right]^2 / (t' \otimes p') - T^2/U,$$

where

$$T = \left((tt') \otimes (pp')\right) - \left((tt') \otimes p'\right) \left(t' \otimes (pp')\right) / \left(t' \otimes p'\right)$$

and

$$U = \left(t' \otimes (p^2 p')\right) - \left(t' \otimes (pp')\right)^2 / \left(t' \otimes p'\right).$$

Notice these are the same 6 cross-correlations as before.

Exact shift and shift-scale matching

- Our algorithms immediately also give us $O(n \log m)$ time solutions for the exact shift and shift-scale matching problems with don't cares.
- Cole and Hariharan previously gave a solution for the exact shift (but not the shift scale) matching problem with don't cares that maps the input symbols into 0 for don't cares and complex numbers of modulus 1 otherwise
- The (complex) cross-correlation between these coded strings is found and a shift match is declared at location i if the modulus of the ith element of the cross-correlation is equal to $(p' \otimes t')[i]$.
- Our solution to this problem involves 6 integer cross-correlation calculations and some simple arithmetic.

Conclusions

- We have shown $O(n \log m)$ time algorithms for self-normalised shift and shift-scale distance calculation when don't cares are allowed in both the pattern and text.
- As a corollary we show how to solve the classic problem of exact matching with don't cares simply using only integer codes as well as exact shift and exact shift-scale matching all in $O(n \log m)$ time.
- An interesting open question is what cost we should pay for allowing shifts or shift-scaling (or transposition invariance) under norms other than L_2.