Automatic Training of Page Segmentation Algorithms: An Optimization Approach

Song Mao and Tapas Kanungo
Language and Media Processing Laboratory
Center for Automation Research
University of Maryland, College Park, MD 20742

Abstract

Most page segmentation algorithms have user-specifiable free parameters. However, algorithm designers typically do not provide a quantitative/rigorous method for choosing values for these parameters. The free parameter values can affect the segmentation result quite drastically and are very dependent on the particular dataset that the algorithm is being used on. In this paper, we present an automatic training method for choosing free parameters of page segmentation algorithms. The automatic training problem is posed as a multivariate non-smooth function optimization problem. An efficient direct search method — simplex method — is used to solve this optimization problem. This training method is then applied to the training of Kise's page segmentation algorithm. It is found that a set of optimal parameter values and their corresponding performance index can be found using relatively few function evaluations. The UW III dataset was used for conducting our experiments.

1 Introduction

Page segmentation is a crucial preprocessing step in OCR system. In many cases, OCR accuracy heavily depends on page segmentation accuracy. While numerous segmentation algorithms have been proposed in the literature [12, 6, 14, 11, 9, 1], relatively little research effort has been devoted to automatic training of algorithms with user-specifiable free parameters.

Some research algorithms [6, 11, 5] specify default parameter values. In performance evaluation literature, Hoover *et al.* [4] manually selected the algorithm parameters. A common aspect of these training methods is that a set of "optimal parameter values" are *manually* selected based on some assumption regarding the training dataset. To objectively optimize a segmentation algorithm on a given training dataset, a set of optimal parameter values should be *automatically* found by a training procedure.

In this article, we pose the automatic algorithm training problem as an optimization problem. We set-theoretically define a textline based performance metric, which is used to construct an object function. The objective function is a function of the algorithm parameters and the training dataset. This average performance metric on the training data set is used as the objective function value. The simplex search technique introduced by Nelder and Mead [10], which belongs to the class of direct search method [2], is used to find the optimal solution. This method is applied to Kise's Voronoi-diagram-based segmentation algorithm on the University of Washington III dataset [13].

This paper is organized as follows. In Section 2, we define page segmentation and error metrics. In Section 3, we pose the automatic training problem as an optimization problem. In Section 4, we specify the experimental protocol. In Section 5, we report experimental results and provide discussions. Finally, in Section 6, we give our conclusions.

2 The Page Segmentation Problem and Error Metrics

In this section we define page segmentation and the error metrics used. These definitions are based on set theory and mathematical morphology [3].

2.1 Page Segmentation Definition

Let I be a document image, and let G be the groundtruth of I. Let $Z(G) = \{Z_q^G, q = 1, 2, \ldots, \#Z(G)\}$ be a set of groundtruth zones of document image I where $\#$ denotes the cardinality of a set. Let $L(Z_q^G) = \{l_{qj}^G, j = 1, 2, \ldots, \#L(Z_q^G)\}$ be the set of groundtruth textlines in groundtruth zone Z_q^G. Let the set of all groundtruth textlines in document image I be $\mathcal{L} = \cup_{q=1}^{\#Z(G)} L(Z_q^G)$. Let A be a given segmentation algorithm, $Seg_A(\cdot, \cdot)$ be the segmentation function corresponding to the algorithm A. Let R be the segmentation result of algorithm A such that $R = Seg_A(I, \mathbf{p}^A)$ where $Z(R) = \{Z_k^R | k = 1, 2, \ldots, \#Z(R)\}$.

Let $D(\cdot) \subseteq \mathcal{Z}^2$ be the domain of its argument, the groundtruth zones and textlines have the following properties: 1) $D(Z_q^G) \cap D(Z_{q'}^G) = \phi$ for $Z_q^G, Z_{q'}^G \in Z(G)$ and $q \neq q'$, and 2) $D(l_i^G) \cap D(l_{i'}^G) = \phi$ for $l_i^G, l_{i'}^G \in \mathcal{L}$ and $i \neq i'$.

2.2 Error Measurements and Metric Definitions

While a performance metric is typically not unique, researchers can select a particular performance metric to study certain aspects of page segmentation algorithms, a set of error measurements is necessary. Let $T_X, T_Y \in Z^+ \cup \{0\}$ be two length thresholds (number of pixels) that determine if the overlap is significant or not. Let $E(T_X, T_Y) = \{e \in Z^2 | -T_X \leq X(e) \leq T_X, -T_Y \leq Y(e) \leq T_Y\}$ be a region of a rectangle centered at $(0,0)$ with a width of $2T_X+1$ pixels and a height of $2T_Y+1$ pixels where $X(\cdot)$ and $Y(\cdot)$ denote the X and Y coordinates of the argument respectively. We now define two morphological operations: dilation and erosion [3]. Let $A, B \subseteq Z^2$. Morphological *dilation* of A by B is denoted by $A \oplus B$ and is defined as:
$A \oplus B = \left\{c \in Z^2 | c = a + b \text{ for some } a \in A, b \in B\right\}$.
Morphological *erosion* of A by B is denoted by $A \ominus B$ and is defined as:
$A \ominus B = \left\{c \in Z^2 | c + b \in A \text{ for every } b \in B\right\}$.

Now, we define three types of textline based error measurements:

1) Groundtruth textlines that are missed:
$C_L = \{l^G \in \mathcal{L} | D(l^G) \ominus E(T_X, T_Y) \subseteq (\cup_{Z^R \in Z(R)} D(Z^R))^c\}$,

2) Groundtruth textlines whose bounding box is split:
$S_L = \{l^G \in \mathcal{L} | (D(l^G) \ominus E(T_X, T_Y)) \cap D(Z^R) \neq \phi, (D(l^G) \ominus E(T_X, T_Y)) \cap (D(Z^R))^c \neq \phi, \text{ for some } Z^R \in Z(R)\}$,

3) Groundtruth textlines that are horizontally merged:
$M_L = \{l_{qj}^G \in \mathcal{L} | \exists l_{q'j'}^G \in \mathcal{L}, Z^R \in Z(R), q \neq q', Z_q^G, Z_{q'}^G \in Z(G) \text{ such that } (D(l_{qj}^G) \ominus E(T_X, T_Y)) \cap D(Z^R) \neq \phi, (D(l_{q'j'}^G) \ominus E(T_X, T_Y)) \cap D(Z^R) \neq \phi, ((D(l_{qj}^G) \ominus E(0, T_Y)) \oplus E(\infty, 0)) \cap D(Z_{q'}^G) \neq \phi, ((D(l_{q'j'}^G) \ominus E(0, T_Y)) \oplus E(\infty, 0)) \cap D(Z_q^G) \neq \phi\}$.

Let the number of groundtruth error textlines be $\#\{C_L \cup S_L \cup M_L\}$ (miss-detected, split or horizontally merged), and the total number of groundtruth textlines is $\#\mathcal{L}$. We define the performance metric $\rho(I, G, R)$ as textline accuracy:

$$\rho(I, G, R) = \frac{\#\mathcal{L} - \#\{C_L \cup S_L \cup M_L\}}{\#\mathcal{L}}. \quad (1)$$

We only consider three types of textline errors — split, missed and horizontally merged. Our textline-based performance metric has the following features: 1) it is based on set theory and mathematical morphology, 2) it is independent of shape of zones, 3) it is independent of OCR recognition error, 4) it ignores the background information (white space, salt and pepper noise etc.), 5) segmentation errors can be localized, and 6) quantitative evaluations on lower level (e.g. textline, word and character) segmentation algorithms can be readily achieved with little modifications. However, this performance metric needs textline level groundtruth. In general, $\rho(I, G, R)$ can be any user-specified function.

3 Automatic Algorithm Training: The Optimization Problem

We pose the automatic segmentation algorithm training problem as an optimization problem. An optimization problem has three components, the objective function that gives a quantitative measure of goodness, a set of parameters that the objective function is dependent on, and a parameter subspace that defines acceptable or reasonable parameter values. The acceptable or reasonable parameter subspace is typically termed as the constraints of the optimization problem. The purpose of an optimization procedure is to find a set of parameter values for which the objective function gives the "best" (minimum or maximum) measure values. In this section, we first define the objective function for our page segmentation algorithm training problem, then we introduce a direct search algorithm to optimize the defined objective function, and finally we discuss the starting point selection in our optimization problem.

3.1 The Objective Function

Let $\mathbf{p}^A$ be the parameter vector for the segmentation algorithm A, let $\mathcal{T}$ be a training dataset, and let $\rho(I, G, Seg_A(I, \mathbf{p}^A))$ where $(I, G) \in \mathcal{T}$ be a performance metric. We define the objective function $f(\mathbf{p}^A; \mathcal{T}, A, \rho)$ to be minimized as the average textline error rate on the training dataset:

$$f(\mathbf{p}^A; \mathcal{T}, A, \rho) = \frac{1}{\#\mathcal{T}} \left[\sum_{(I,G) \in \mathcal{T}} 1 - \rho(G, Seg_A(I, \mathbf{p}^A)) \right].$$

where $\rho(G, Seg_A(I, \mathbf{p}^A))$ is given by Equation 1. This objective function has the following properties: 1) The function has no explicit mathematical form and is non-differentiable, 2) Only function evaluations are possible, 3) Obtaining a function value requires nontrivial computation. This objective function can be classified as a *multivariate non-smooth function*. In the following section, we describe an optimization algorithm to minimize this objective function.

3.2 The Simplex Search Method

Direct search methods are typically used to solve the optimization problem described in Section 4.1. We choose the simplex search method proposed by Nelder and Mead [10] to minimize our objective function.

We give the notation used to describe the simplex method: Let $\mathbf{q}_0$ be a starting point in segmentation algorithm parameter space, and let $\lambda_i, i = 1, \ldots, n$ be a set of scales. Let $\mathbf{e}_i, i = 1, \ldots, n$ be n orthogonal unit vectors in n-dimensional parameter space, let $\mathbf{p}_0, \ldots, \mathbf{p}_n$ be $(n+1)$ ordered points in n-dimensional parameter space such that their corresponding function values satisfy $f_0 \leq f_1 \leq, \ldots, \leq f_n$, let $\bar{\mathbf{p}} = \sum_{i=0}^{n-1} \mathbf{p}_i/n$ be the centroid of the n best (smallest) points, let $[\mathbf{p}_i\mathbf{p}_j]$ be the n-dimensional Euclidean distance from $\mathbf{p}_i$ to $\mathbf{p}_j$, let α, β, γ and σ be the *reflection, contraction, expansion and shrinkage coefficient,* respectively, and let T be the threshold for the stopping criterion. For a segmentation algorithm with n parameters, the Nelder-Mead algorithm works as follows:

1 Given $\mathbf{q}_0$ and the λ_i, form the initial simplex as
$\mathbf{q}_i = \mathbf{q}_0 + \lambda_i\mathbf{e}_i, i = 1, \ldots, n.$,
2 Relabel the $n+1$ vertices as $\mathbf{p}_0, \ldots, \mathbf{p}_n$ with
$f(\mathbf{p}_0) \leq f(\mathbf{p}_1) \cdots \leq f(\mathbf{p}_n)$,
3 Get a reflection point $\mathbf{p}_r$ of $\mathbf{p}_n$ by $\mathbf{p}_r = (1+\alpha)\bar{\mathbf{p}} - \alpha\mathbf{p}_n$
where $\alpha = [\mathbf{p}_r\bar{\mathbf{p}}]/[\mathbf{p}_n\bar{\mathbf{p}}]$.
4.1 If $f(\mathbf{p}_r) \leq f(\mathbf{p}_0)$, replace $\mathbf{p}_n$ by $\mathbf{p}_r$ and $f(\mathbf{p}_n)$ by
$f(\mathbf{p}_r)$, get an expansion point $\mathbf{p}_e$ of $\mathbf{p}_n$ by
$\mathbf{p}_e = (1-\gamma)\bar{\mathbf{p}} + \gamma\mathbf{p}_n$ where $\gamma = [\mathbf{p}_e\bar{\mathbf{p}}]/[\mathbf{p}_n\bar{\mathbf{p}}] > 1$.
If $f(\mathbf{p}_e) < f(\mathbf{p}_n)$, replace $\mathbf{p}_n$ by $\mathbf{p}_e$ and $f(\mathbf{p}_n)$ by $f(\mathbf{p}_e)$.
Go to step 5.
4.2 Else if $f(\mathbf{p}_r) \geq f(\mathbf{p}_{n-1})$, if $f(\mathbf{p}_r) < f(\mathbf{p}_n)$ replace
$\mathbf{p}_n$ by $\mathbf{p}_r$ and $f(\mathbf{p}_n)$ by $f(\mathbf{p}_r)$, get a contraction point $\mathbf{p}_c$
of $\mathbf{p}_n$ by $\mathbf{p}_c = (1-\beta)\bar{\mathbf{p}} + \beta\mathbf{p}_n$, $\beta = [\mathbf{p}_c\bar{\mathbf{p}}]/[\mathbf{p}_n\bar{\mathbf{p}}] < 1$.
If $f(\mathbf{p}_c) \geq f(\mathbf{p}_n)$, shrink the simplex around the best
vertex $\mathbf{p}_0$ by $\mathbf{p}_i = (\mathbf{p}_i + \mathbf{p}_0)\sigma, i \neq 0$, else replace $\mathbf{p}_n$
by $\mathbf{p}_c$ and $f(\mathbf{p}_n)$ by $f(\mathbf{p}_c)$, go to step 5.
4.3 Else, replace $\mathbf{p}_n$ by $\mathbf{p}_r$ and $f(\mathbf{p}_n)$ by $f(\mathbf{p}_r)$.
5 If $\sqrt{\sum_{i=0}^{n}(f(\mathbf{p}_i) - f(\bar{\mathbf{p}}))^2/n} < T$, stop else go to step 2.

3.3 Multiple Starting Point Selection

The objective function corresponding to each segmentation algorithm need not have a unique minimum. Furthermore, direct search optimization algorithms are *local* optimization algorithm. Thus, for each (different) starting point, the optimization algorithm could converge to a different optimal solution. We constrain the parameter values to lie within a reasonable range and randomly choose six starting locations within this range. The optimal solution corresponding to the lowest optimal value is chosen as the best optimal parameter vector.

4 Experimental Protocol

We select the University of Washington Dataset [13] for the algorithm training task. A training dataset of 100 document pages was randomly sampled from the selected 978 documents in the UW III dataset. The dataset contains geometric textline and zone groundtruth for each page. We compute a performance metric only on text regions.

Kise's algorithm [6] works as follows: 1) label connected components, 2) remove noise connected components, 3) generate the Voronoi diagram for each connected component using the sample points on its border, 4) delete superfluous Voronoi edges according to a area-spacing criterion to generate zone boundaries, 5) remove noisy zones.

Kise's algorithm has eleven free parameters and is insensitive to seven of them. We fix the seven parameters as follows: maximum height and width thresholds of a connected component, $C_h = 500$ pixels and $C_w = 500$ pixels, maximum connected component aspect ratio threshold, $C_r = 5$, minimum area threshold of a zone, $A_z = 50$ pixels^2 for all zones, and minimum area threshold, $A_l = 40000$ pixels, and maximum aspect ratio threshold, $B_r = 4$ for the zones that are vertical and elongated. The last parameter is the size of the smoothing window, which is fixed at $sw = 2$. The optimal values for the other four parameters are searched from the following ranges recommended by Kise:
1) sampling rate sr: {4-7}, 2) maximum size threshold of noise connected component nm: {10-40}, 3) margin control factor for Td2 fr: {0.01-0.5}, 4) area ratio threshold ta: {40-200}.

The machines we use are Ultra 1,2 and 5 Sun workstations running Solaris 2.6 operating system. After the training step, a set of optimal parameter values are found for each research algorithm.

5 Experimental Results and Discussions

From Figure 1 and Table 1, we can make the following observations [1]:

1) The error rates for all starting points converge in the range of 4.74% to 5.52%, 2) The convergence rate before first 30 function evaluations is much faster than that beyond 30
function evaluations, 3) The value parameter nm for most (five) starting points converges to 11 pixels, 4) There is relatively small variance in the convergence values of parameter sr, nm and ta, 5) There is relatively large variance of the convergence values of parameter fr, 6) There is a relatively large variance of the number of function evaluations corresponding to six starting points.

[1] Note that some numbers reported in this paper differ from those reported in our technical report [7]. In [7] we used Numerical Recipes version of Nelder-Mead algorithm whereas in this paper we use the original [10] algorithm.

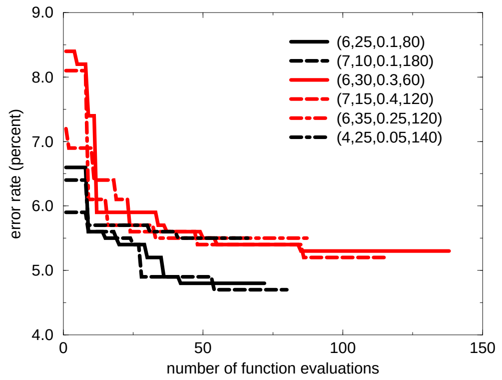

Figure 1. Convergence curves corresponding to six randomly selected starting points in the training of the Voronoi algorithm.

Table 1. Optimization results of Voronoi algorithm for six randomly selected starting points within a reasonable working parameter subspace.

start parameter values (sr, nm, fr, ta)	optimal parameter values (sr, nm, fr, ta)	error rate (percent)	number of function evaluations	timing (hours)
$(6, 25, 0.1, 80)$	$(6, 15, 0.079, 106)$	4.80	72	14.06
$(7, 10, 0.1, 180)$	$(6, 11, 0.083, 199)$	4.74	80	14.97
$(6, 30, 0.3, 60)$	$(6, 11, 0.147, 148)$	5.31	138	39.30
$(7, 15, 0.4, 120)$	$(8, 11, 0.098, 190)$	5.18	116	31.52
$(6, 35, 0.25, 120)$	$(6, 11, 0.246, 193)$	5.52	95	32.78
$(4, 25, 0.05, 140)$	$(4, 11, 0.138, 160)$	5.49	66	15.80

From the above observations, we can see that the Voronoi algorithm objective function has multiple local minima, but the performance at these local minima is stable. The algorithm only needs about 30 function evaluations to reach a stable performance. The optimal algorithm performance is insensitive to the value of parameter fr. The fact that the optimal value of parameter ta is big implies that the text and non-text connected components are well separated. The fact that the values of parameter fr are generally small indicate we should choose a conservative (large) interline spacing threshold. This training methodology is very general and has been applied to many page segmentation algorithms [8, 7].

6 Conclusions

We posed the automatic segmentation algorithm training problem as a multivariate non-smooth function optimization problem. A textline based performance metric was defined using set theory and mathematical morphology. This textline based metric was used to construct the objective function to be minimized. Nelder-Mead simplex method was then used to solve the optimization problem. An empirical analysis of the effect of initial parameter values and scales on optimization results was performed. From the experimental results, we found that a set of "optimal" parameter values and their corresponding "optimal" objective function value can be quickly found with relatively less computation.

References

[1] H. S. Baird, S. E. Jones, and S. J. Fortune. Image segmentation by shape-directed covers. In *Proceedings of International Conference on Pattern Recognition*, pages 820–825, Atlantic City, NJ, June 1990.

[2] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*, chapter 4. Academic Press, London and New York, 1993.

[3] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Publishing Company, Reading, MA, 1992.

[4] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldof, K. W. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:673–689, 1996.

[5] A. K. Jain and B. Yu. Document representation and its application to page decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:294–308, 1998.

[6] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding*, 70:370–382, 1998.

[7] S. Mao and T. Kanungo. A methodology for empirical performance evaluation of page segmentation algorithms. Technical Report CAR-TR-933, University of Maryland, College Park, MD, 1999. http://www.cfar.umd.edu/˜kanungo/pubs/tr-segeval.ps.

[8] S. Mao and T. Kanungo. Empirical performance evaluation of page segmentation algorithms. In *Proceedings of SPIE Conference on Document Recognition*, San Jose, CA, January 2000.

[9] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In *Proceedings of International Conference on Pattern Recognition*, volume 1, pages 347–349, Montreal, Canada, July 1984.

[10] J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[11] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1162–1173, 1993.

[12] L. O'Gorman and R. Kasturi. *Document Image Analysis*. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[13] I. Phillips. *User's Reference Manual*. CD-ROM, UW-III Document Image Database-III.

[14] F. Wahl, K. Wong, and R. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Vision, Graphics, and Image Processing*, 20:375–390, 1982.