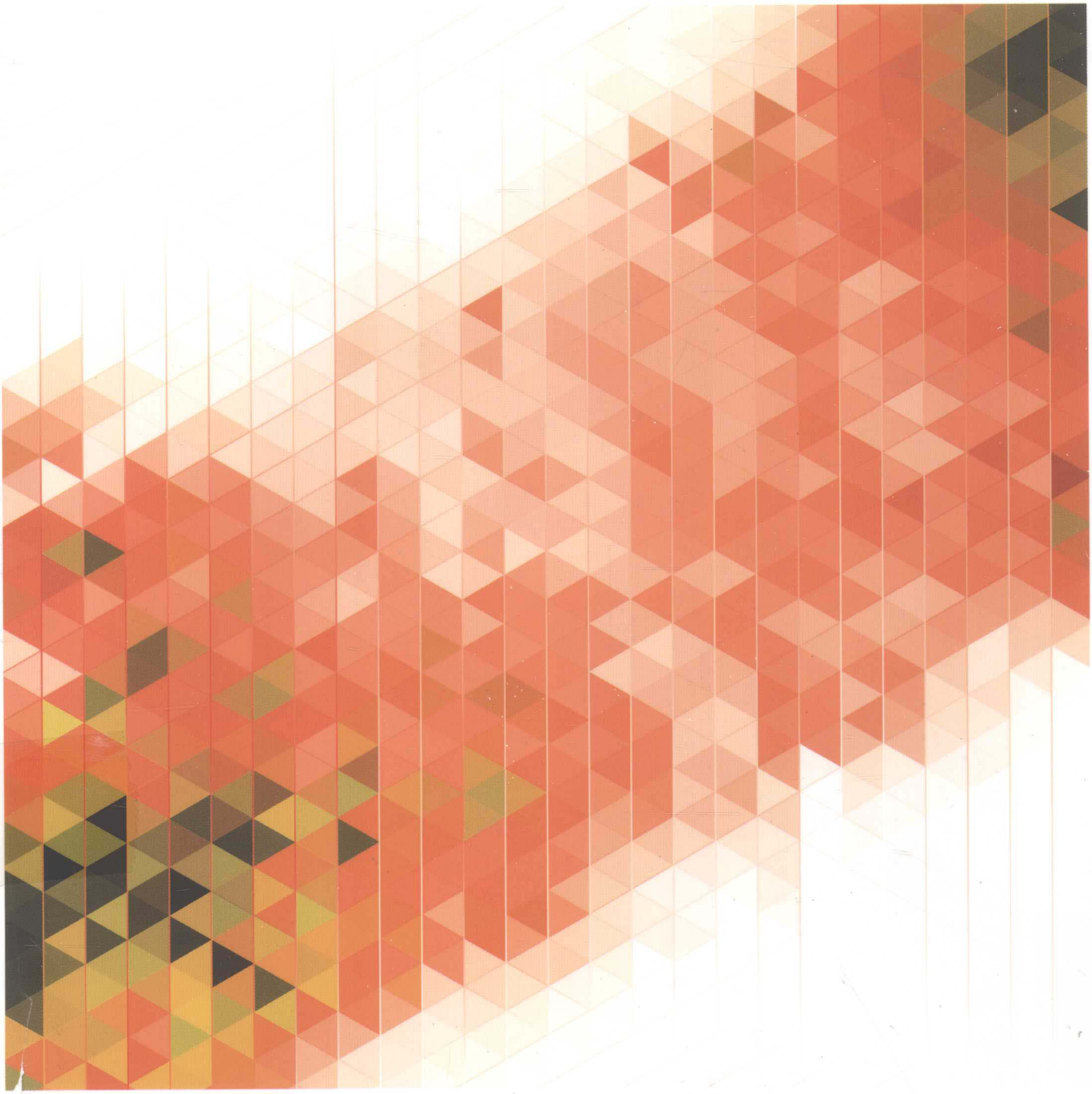
Advanced Structured Prediction

EDITED BY

Sebastian Nowozin, Peter V. Gehler,
Jeremy Jancsary, and Christoph H. Lampert



Advanced Structured Prediction

Edited by

Sebastian Nowozin, Peter V. Gehler, Jeremy Jancsary, and
Christoph H. Lampert

The MIT Press
Cambridge, Massachusetts
London, England

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu

This book was set in LaTeX by the authors and editors. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Advanced structured prediction / edited by Sebastian Nowozin, Peter V. Gehler, Jeremy Jancsary, and Christoph H. Lampert.
pages cm. - (Neural information processing series)
Includes bibliographical references and index.
ISBN 978-0-262-02837-0 (hardcover : alk. paper) 1. Machine learning. 2. Computer algorithms. 3. Data structures (Computer science) I. Nowozin, Sebastian, 1980– editor.
Q325.5.A295 2014
006.3'1-dc23
2014013235

10 9 8 7 6 5 4 3 2 1

Series Foreword

The yearly Neural Information Processing Systems (NIPS) workshops bring together scientists with broadly varying backgrounds in statistics, mathematics, computer science, physics, electrical engineering, neuroscience, and cognitive science, unified by a common desire to develop novel computational and statistical strategies for information processing and to understand the mechanisms for information processing in the brain. In contrast to conferences, these workshops maintain a flexible format that both allows and encourages the presentation and discussion of work in progress. They thus serve as an incubator for the development of important new ideas in this rapidly evolving field. The series editors, in consultation with workshop organizers and members of the NIPS Foundation Board, select specific workshop topics on the basis of scientific excellence, intellectual breadth, and technical impact. Collections of papers chosen and edited by the organizers of specific workshops are built around pedagogical introductory chapters, while research monographs provide comprehensive descriptions of workshop-related topics, to create a series of books that provides a timely, authoritative account of the latest developments in the exciting field of neural computation.

Michael I. Jordan and Thomas G. Dietterich

Preface

Machine learning is one of the fastest growing areas of computer science, and with good reason: predictive machine learning models trained on ever growing data sets provide relevant information to scientists and business decision makers alike, as well as enabling intelligent consumer applications.

Structured prediction refers to machine learning models that predict relational information that has structure such as being composed of multiple interrelated parts. For example, these models are used to predict a natural language sentence or segment an image into meaningful components. Structured prediction models are important in many application domains and have been used with great success in biology, computer vision, and natural language processing.

This volume is not the first on the topic of structured prediction; seven years ago, in 2007, MIT Press released the edited volume *Predicting Structured Data*. Since then structured prediction has blossomed into many application areas, but it has not settled down yet; there continues to be a stream of interesting and original work. In an introduction chapter, we summarize the state-of-the-art and recent developments. The remainder of the volume is a careful selection of contributed chapters.

We would like to thank all chapter contributors for their high-quality work, Marie Lufkin Lee from MIT Press for her support and patience, Suvrit Sra for help in preparing a LaTeX template for this volume, and Jasmin Pielorz for help with proofreading and copy-editing.

We dedicate this volume to the memory of Ben Taskar, a pioneer of the field.

Sebastian Nowozin, Peter V. Gehler,
Jeremy Jancsary, Christoph H. Lampert

Cambridge, Tübingen, Vienna, Klosterneuburg
January 2014

Contents

1 Introduction to Structured Prediction

Sebastian Nowozin Sebastian.Nowozin@Microsoft.com
Microsoft Research
Cambridge, United Kingdom

Peter V. Gehler Peter.Gehler@tuebingen.mpg.de
Max Planck Insitute for Intelligent Systems
72076 Tübingen, Germany

Jeremy Jancsary Jeremy.Jancsary@Nuance.com
Nuance Communications
Vienna, Austria

Christoph H. Lampert chl@ist.ac.at
IST Austria
A-3400 Klosterneuburg, Austria

Structured prediction refers to machine learning models that predict multiple interrelated and dependent quantities. These models are commonly used in computer vision, speech recognition, natural language processing, and computational biology to accurately reflect prior knowledge, task-specific relations, and constraints. A wide variety of types of models is used, and they are expressive and powerful, but exact computation in these models is often intractable. This difficulty, paired with the practical significance, has resulted in a broad research effort in recent years to design structured prediction models and approximate inference and learning procedures that are computationally efficient. This chapter gives an introduction to structured prediction and summarizes the main approaches. It includes a discussion of the research trends in the field since 2007 and provides further references for the interested reader.

1.1 Structured Prediction

The general structured prediction problem is defined as follows. Given an observation $x \in \mathcal{X}$, make a prediction $y \in \mathcal{Y}(x)$ as

$$y = f(x). \tag{1.1}$$

The set $\mathcal{Y}(x)$ is typically finite but exponentially large, and its size may depend on the input x. A popular choice is to use an index set $I = \{1, 2, \ldots, m\}$ and define both input x and prediction y as

$$x = (x_1, \ldots, x_m), \qquad \text{and} \qquad y = (y_1, \ldots, y_m).$$

For example, I can index all words in a sentence or all pixels in an image.

Researchers working in structured prediction are concerned with the *representation* of the function f, procedures for *evaluating* $f(x)$ for a given input x, and *learning* f from a class of functions $\mathcal{F}$ given annotated training data consisting of pairs (x, y) of data instances.

We describe these three aspects below, but first we would like to define how to measure the quality of a structured prediction model (1.1) by means of loss functions.

Loss Functions and Decision Rules. A generally accepted criterion for assessing the quality of our model is that of the *expected loss* of our model as a function of the true generating probability distribution $q(x, y)$ and a *loss function*[1] $\ell : \mathcal{Y}(x) \times \mathcal{Y}(x) \to \mathbb{R}$. The distribution $q(x, y)$ is the sampling distribution we encounter when our model (1.1) is used; for example, it could be the joint distribution of emails x and spam/no-spam decisions y that are sent to a particular email address. We do not know q, but a standard assumption is that we are able to obtain independent and identically distributed (iid) samples from it. The loss function $\ell(z, y)$ quantifies—on an arbitrary but fixed scale—the loss suffered if z happens to be the truth and we decide for y. The quality of a structured prediction model can now be quantified as the *risk*,

$$\mathcal{R}(f, q, \ell) = \mathbb{E}_{(x,y)\sim q}\left[\ell(y, f(x))\right]. \tag{1.2}$$

1. Alternatively, an equivalent definition can be made using *utility functions*; we want to maximize utility or minimize loss, but except for a change in sign, both definitions are identical. The loss function can be more generally defined as $\ell : \mathcal{Y} \times \mathcal{D} \to \mathbb{R}$, where $\mathcal{D}$ is the *decision domain*, which can differ from $\mathcal{Y}$.

Because $\mathcal{R}$ depends on the unknown distribution q, the expectation is approximated using a data set $D = \{(x^{(i)}, y^{(i)})\}_{i=1,\dots,N}$ sampled iid from q, yielding the *empirical risk*,

$$\mathcal{R}_{\text{emp}}(f, D, \ell) = \frac{1}{N} \sum_{i=1}^{N} \ell(y^{(i)}, f(x^{(i)})). \tag{1.3}$$

While there are different philosophies with respect to how to best build structured prediction models and which loss functions are relevant to an application, the criterion (1.3) is widely accepted.

The best possible risk, which is the lowest possible, is known as the *Bayes risk*. It is defined by making the optimal decisions with the knowledge of q, that is, $\mathcal{R}_{\text{Bayes}}(q, \ell) = \mathcal{R}(f_{\text{Bayes}}, q, \ell)$, where f_{Bayes} is the Bayes-optimal predictor,

$$f_{\text{Bayes}}(x) = \operatorname*{argmin}_{y \in \mathcal{Y}(x)} \mathbb{E}_{z \sim q(z|x)} \left[\ell(z, y)\right]. \tag{1.4}$$

Representation. For representing the function f, different choices exist; one popular branch of the literature defines $f(x)$ as the maximizer of an auxiliary optimization problem,

$$f(x) = \operatorname*{argmax}_{y \in \mathcal{Y}(x)} F(x, y, \theta), \tag{1.5}$$

where $\theta \in \Theta$ are model parameters. In many applications, solving (1.5) corresponds to solving a combinatorial optimization problem. The function $F(x, y, \theta)$ to be maximized is commonly parametrized as a linear form,

$$F(x, y, \theta) = \langle \phi(x, y), \theta \rangle, \tag{1.6}$$

where $\Theta = \mathbb{R}^d$ and $\phi(x, y)$ is a *joint feature map*, transforming x and y into a large but fixed size feature vector. The class of functions is now indexed by θ, and we have

$$\mathcal{F} = \{F(\cdot, \cdot, \theta) \mid \theta \in \mathbb{R}^d\}. \tag{1.7}$$

Another approach to construct structured prediction functions is by starting with a probabilistic model and applying Bayesian decision theory (Berger, 1985). For this we assume that we have a model for the conditional distribution $p(y|x; \theta)$ over $\mathcal{Y}(x)$. Together with a loss function ℓ, we can then use the *Bayes decision rule*,

$$f(x) = \operatorname*{argmin}_{y \in \mathcal{Y}(x)} \mathbb{E}_{z \sim p(z|x)} \left[\ell(z, y)\right]. \tag{1.8}$$

This rule is identical to (1.4), except we replaced the unknown distribution q with our model p. Intuitively (1.8) selects our prediction so that we minimize our expected loss under every possibility z, weighted by our beliefs about the state of the world as encoded in $p(z|x)$. The similarity between (1.8) and (1.4) implies that if p equals the true distribution q, then our decisions made using the Bayes decision rule will be optimal, that is, they will achieve the *Bayes risk*.

Evaluation. Using either definition (1.5) or (1.8), in order to make predictions, we need to solve an optimization problem, an instance of an *inference problem*. Depending on the structure of F and $\mathcal{Y}(x)$, problem (1.5) may be intractable to solve exactly, and we need to develop *approximate inference* methods. When using (1.5), such methods are often called *energy minimization methods*, and a large part of the structured perdiction literature is concerned with their properties. In case (1.8) is used, the tractability depends on the distribution p, the loss function ℓ, and the set $\mathcal{Y}(x)$. For example, if the so-called 0/1-loss $\ell_{0/1}(z, y) = 1_{\{y \neq z\}}$ is used, the problem reduces to the *maximum-aposteriori* (MAP) decision rule,

$$f(x) = \underset{y \in \mathcal{Y}(x)}{\operatorname{argmax}}\ p(y|x). \tag{1.9}$$

If the loss function decomposes additively over individual dimensions of its arguments, then we can solve (1.8) in two steps, where first a set of low-dimensional marginal distributions $p(y_i|x)$ is inferred, and then decisions are independently made by minimizing $\mathbb{E}_{z_i \sim p(z_i|x)}\left[\ell_i(z_i, y_i)\right]$ (Marroquin et al., 1987). Inferring the marginal distributions $p(y_i|x)$, also known as *marginal beliefs*, requires probabilistic inference methods for the model. In the last fifteen years, a large number of approximate inference methods have been developed to this end.

One important class of methods, the linear programming relaxations, apply to discrete graphical models (Wainwright and Jordan, 2008). For these models (1.5) can be reformulated as an integer linear program, which can be relaxed to a polynomial-time solvable linear program for which specialized message-passing algorithms have been developed. These algorithms are now popularly used and provide robust inference for otherwise challenging models, but until recently, understanding the structure and limitations of the linear programming relaxation approach has been an open question.

Learning. Structured prediction models can be learned in different ways from a given data set of iid samples. If the direct form of the predictor (1.5) is adopted, then the most popular choice is *regularized risk minimization*

(Vapnik and Chervonenkis, 1974), in which we minimize the regularized empirical risk,

$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \; \Omega(f) + \frac{1}{N} \sum_{i=1}^{N} \ell(y^{(i)}, f(x^{(i)})). \tag{1.10}$$

Here $\Omega(f)$ is a regularizer that controls the capacity of the learned model $\hat{f}$. While (1.10) has served as motivation for a large number of general machine learning methods, the application to structured prediction problems was only enabled through the work of Tsochantaridis et al. (2004), who showed how (1.10) can be implemented in the structured case when the linear form (1.6) for the definition of f is used.

They propose the *structured support vector machine*, which learns the parameters θ of the predictor f by solving the problem

$$\underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \; \frac{1}{2}\|\theta\|^2 + \frac{\lambda}{N} \sum_{i=1}^{N} L(y^{(i)}, x^{(i)}, \theta), \tag{1.11}$$

where $\lambda > 0$ is a regularization parameter, and we define

$$L(y^{(i)}, x^{(i)}, \theta) = \max_{y \in \mathcal{Y}(x^{(i)})} \left[\ell(y^{(i)}, y) - F(x^{(i)}, y^{(i)}, \theta) + F(x^{(i)}, y, \theta) \right]. \tag{1.12}$$

It can be shown that $L(y^{(i)}, x^{(i)}, \theta) \geq \ell(y^{(i)}, f(x^{(i)}))$, that is, L is an upper bound of ℓ for any θ. Therefore, (1.11) is an upper bound of the empirical risk (1.3), and by minimization of the upper bound, we can find model parameters with low empirical risk. Given enough training data, the empirical risk will be close to the true risk (1.2). While it is not trivial to solve (1.11), it is a convex optimization problem, and the tractability of the formulation has enabled a large number of structured prediction applications.

When the probabilistic perspective is adopted, learning of the model is performed using the model likelihood, using either maximum likelihod estimation (MLE) or Bayesian inference (Koller and Friedman, 2009). The model can either be generative $p(x, y|\theta)$ or discriminative $p(y|x, \theta)$ as in conditional random fields (Lafferty et al., 2001). The generative model provides an explicit model for the inputs x, whereas the discriminative model always conditions on an observed x. For the following example, let us use a discriminative model. We specify a *prior distribution* $p(\theta)$ for the model parameters and then solve

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \; p(\theta) \prod_{i=1}^{N} p(y^{(i)}|x^{(i)}, \theta), \tag{1.13}$$

for the maximum likelihood estimate $\hat{\theta}$ or use Bayes rule to define a posterior belief over θ given the data set D as

$$\begin{aligned} p(\theta|D) &= \frac{p(\theta)\, p(D|\theta)}{p(D)} &(1.14)\\ &= \frac{p(\theta)\prod_{i=1}^{N} p(y^{(i)}|x^{(i)},\theta)}{\int_\Theta p(\theta)\prod_{i=1}^{N} p(y^{(i)}|x^{(i)},\theta)\,\mathrm{d}\theta} &(1.15)\\ &\propto p(\theta)\prod_{i=1}^{N} p(y^{(i)}|x^{(i)},\theta). &(1.16)\end{aligned}$$

At test time, given a new observation x, we proceed as follows. In case the MLE is used, the predictive distribution is derived from the point estimate $\hat{\theta}$ simply as $p(y|x,\hat{\theta})$. In case the posterior $p(\theta|D)$ is used, the predictive distribution marginalizes over all parameter uncertainty as

$$p(y|x) = \int_\Theta p(y|x,\theta)\, p(\theta|D)\,\mathrm{d}\theta. \tag{1.17}$$

Learning structured prediction models is challenging because it is usually intractable to perform exact computation of the required quantities, with few exceptions—for example, in so-called linear chain models. In both (1.13) and (1.16), we use $p(y^{(i)}|x^{(i)},\theta)$, but this important distribution typically cannot be exactly computed. Likelihood-based learning of structured models has therefore required approximations; a large variety of approximate inference and estimation methods have been proposed. A rough grouping of these methods is into stochastic and deterministic approximations.

Stochastic approximations perform Monte Carlo simulations to approximate expectations and integrals in the learning objective. Examples are MCMC-MLE (Descombes et al., 1999) and stochastic maximum likelihood approaches such as contrastive divergence (Hinton, 2002; Carreira-Perpiñán and Hinton, 2005).

Deterministic approximations typically optimize an auxiliary objective function; the class of *variational approximations* such as *mean field* methods (Saul and Jordan, 1995; Xing et al., 2003), *loopy belief propagation* (Yedidia et al., 2004), or more generally methods derived from the minimization of statistical divergence measures (Minka, 2005) are commonly used for otherwise intractable models. Another class of deterministic approximations instead modify the likelihood function itself to obtain tractable estimators; these include the *pseudolikelihood* (Besag, 1972, 1977), more general *composite likelihoods* (Lindsay, 1988; Varin et al., 2010), and *score matching* (Hyvärinen, 2005).

All these approximations have different trade-offs with respect to computational effort, the robustness and accuracy of the inference results, as well as the theory that is known about them. It is fair to say that while great progress has been made, for most models, there is not yet a clear favorite among the above approximate methods.

1.2 Recent Developments

We now briefly summarize the most significant developments in the field of structured prediction since around 2007, when the previous volume in this series was published (Bakır et al., 2007).

Joint Optimization over Parameters and Inference. Meshi et al. (2010) introduced a clever method to approximately solve (1.10) for discrete graphical models. In their method they rewrite ℓ as a maximization problem over vectors μ in the *local polytope* so that (1.10) is of a min-max structure with multiple inner maximization problems. The inner problems corresponding to ℓ are then dualized using convex duality to obtain a joint min-min problem over parameters θ as well as dual message vectors. The advantage of this is that one can now interleave message passing updates and parameter updates, whereas previously every parameter update required a message passing scheme to run to convergence. The result is an efficient learning method. A similar but more general method has been proposed by Hazan and Urtasun (2010). In this volume, Chapter 9 by Justin Domke continues this line of work to learn more expressive non-linear model potentials in which the parameter update step is replaced by a non-linear logistic regression subproblem.

Integrated Estimation and Inference. A recent take on how to deal with the intractability of structured prediction models is due to Domke (2011), Ross et al. (2011b), and Stoyanov et al. (2011). The idea is to take a model and an iterative approximate inference procedure, such as loopy belief propagation, and to view them as one computational unit that iteratively transforms some initial state into an inference result. As such, when using a fixed number of inference iterations, it is just a non-linear differentiable mapping from parameters and observations to inference results. This mapping is parametrized by the original model parameters, and as long as we can compute the gradient with respect to these parameters, a gradient-based optimizer can be used to minimize the empirical risk (Domke, 2013; Ross et al., 2011b). The combination of model and approximate