Extending Symptom-Checking Applications for Virtual Healthcare Interaction

Albert Goldfain and James DelloStritto

Blue Highway, Welch Allyn 2-212 Center for Science & Technology Syracuse, New York 13244-4100 {agoldfain|jdellostritto}@blue-highway.com

Abstract

This paper describes four tools that support and extend the functionality of symptom-checking applications. Prototype implementations for these tools are developed using MAT-LAB, OWL, and KML.

Introduction

Various symptom-checking applications have been developed to help users determine the likely causes of symptoms (e.g., Mayo Clinic¹, WebMD², Health Central³). Such applications provide an intuitive and easy-to-navigate user interface through which the patient selects a symptom or set of symptoms and through which detailed information is displayed about the probable causes. Valuable advice can be delivered quickly without sophisticated underlying intelligence. All web-based symptom-checking applications warn that they are not a substitute for a professional medical opinion, but because they deliver medically sound information, they have become valuable resources.

In what follows, we describe four software extensions for a general symptom-checking application which extend their functionality and increase their value in the context of virtual healthcare interaction.

Fusing Sign and Symptom Spaces

The decreasing cost of vital-sign sensing and home monitoring will increase the deployment of wearable monitoring devices for home healthcare (Lukowicz 2008). This trend promises clinical-grade information for the cost of a household appliance. Objective measurements tend to be a more reliable basis for clinical decision making than symptoms alone. Measurements in a clinical context reveal objective *signs* of disease and disorder rather than subjectively experienced symptoms. As such, patients with access to device measurements should be able to use enter sign data alongside their symptoms.

For the purpose of computational clustering, a patient's set of symptoms can be formalized as a feature vector which

Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

http://www.mayoclinic.com/health/symptom-checker/DS00671

forms a point in a multidimensional *symptom space*. Values for the feature vector components are either binary (1 = symptom present or 0 = symptom absent), or enumerated (slight, mild, severe, chronic). The set of signs can also be formalized as a feature vector, however the dimensions of the corresponding *sign space* will be either continuous or more dense than symptom space. In the ideal case, clusters of points in these spaces will be linearly separable and will fall within regions corresponding to known diseases and disorders.

In general, there is a many-to-many relationship between signs and symptoms, so attempting to accurately correlate signs with symptoms can be computationally expensive. However, clustering in the (topological) product of sign and symptom space should enhance performance.

We have developed a MATLAB prototype implementation of a tool to perform clustering in the product of a sign and symptom space. The tool normalizes both sign and symptom feature vectors. Given a suitable sign space metric d_{sig} for normalized sign vectors x_i , a suitable symptom space metric d_{sym} for a normalized symptom vector y_i , a *p*-product metric is defined for the "fused" product space :

 $d_p((x_1, x_2), (y_1, y_2)) = \sqrt[p]{d_{sig}(x_1, x_2)^p + d_{sym}(y_1, y_2)^p}$

The individual sign and symptom metrics can be scaled by the user if they wish to assign more relevance to signs or symptoms.

Training for Rare and Relevant Symptoms

Two tacit presuppositions of symptom-checking applications is that the menu of symptoms is exhaustive and that each is descriptive enough to capture a subjective feeling of the user. This is especially difficult because, for example, different users have different pain thresholds, so a term such as 'acute stabbing pain in the leg' may mean different things. Even setting aside these difficulties, a user may exhibit a rare symptom (or combination of symptoms) or may find that the list is not exhaustive (i.e., a situation where "none of the above" would be the best choice).

Here again, the formalism of a metric symptom space (or sign and symptom product space) can be valuable. When a patient's feature vector falls outside the regions of known disease the metric space, or otherwise is indescribable given the finite set of choices, the instance should be stored off

² http://symptoms.webmd.com/default.htm

³http://www.healthcentral.com/symptom-checker/

in a database. Eventually, it may be the case that enough similarities exist between such patient instances that a new disease or disorder is discovered. Even short of this discovery, symptom combinations that are unknown or rare should be used for training healthcare professionals because they may represent tough or borderline cases. Statistical analysis tends towards the expected, which is very important, but this can make the unexpected look like outlier noise. In a medical context, it is important not to let these cases fall through the cracks. When users find symptoms hard to describe or find no results for their symptom combination, the system should probe further.

We extend the prototype implementation described above to allow for partially unspecified feature vectors (indescribable symptoms). Furthermore, we are able to detect when the distance from a feature vector exceeds a threshold distance from any prototypical symptom vector for a disease or disorder. Such instances are stored in a database back end. Such a database can be used to train both human and automated classifiers.

Visualizing Symptom Spread

Certain infectious diseases can spread quickly when pathogen strains evolve and mutate faster than the relevant data can be collected. As such, the visualization of disease spread can support a planned response and can help guide treatment decisions. Epidemiologists measure the virulence and pathogenicity of an infectious agent on the basis of clinically reported cases. Unfortunately, these numbers reveal nothing about symptomatic individuals who have not yet visited a clinic. In areas with very few clinics, it may be the case that more people can search for symptoms than can actually visit clinics.

Some symptom-checking applications provide an optional ZIP code field as part of the user interface. This information does not violate the privacy of the user and can be used to geographically track the spread of similar symptoms. For example, if a significant number of people query for flu-like symptoms around the same ZIP code, and if the trend is toward similar symptom queries beginning west of the ZIP code, this may be an indicator of a flu spreading west. The underlying assumption is that the spread of (the same) symptoms is a good indicator of spread of (the same) disease, and that people are searching on their own behalf or for someone geographically close to them.

We have a preliminary implementation of a tool which associates ZIP codes with KML-formatted latitude and longitude coordinates. Symptom queries are timestamped and tagged with KML data. In this format, it can be used for display in Google Earth (see Figure 1). Similar tools exist for tracking clinically reported data (e.g., Health Map's Global disease alert map⁴).

Ontological Support

A sophisticated, broad coverage symptom-checking application should be founded on a high-quality ontology of dis-



Figure 1: Tracking disease spread through symptom queries.

ease and disorder. Such an ontology would provide: a consistent terminology for automated reasoning, interoperability between different symptom-checking repositories, and a machine usable analysis of the relations obtaining between symptoms (e.g., the 'triggered by', 'relieved by', and 'accompanied by' relations used in the Mayo Clinic symptom checker). Furthermore, an ontology can serve as a tool for annotating symptom data for semantic search applications or any other applications for which symptom queries form datasets.

We have completed a clinical phenotype ontology (CPO) based on (Scheuermann, Ceusters, and Smith 2009) which provides a formal characterization of signs, symptoms, and disorders. CPO treats disorders as the underlying basis for disease at all scales of biological reality (gene, cell, organ, organism, and population). The ontology available in both OWL and OBO formats.

Conclusion

Although the tools described are only at a preliminary or prototype stage of implementation, the ease with which the core symptom-checking model can be extended gives us reason for optimism. Collectively, such tools enhance the value of symptom checking and deliver valuable information to the user.

References

Lukowicz, P. 2008. Wearable Computing and Artificial Intelligence for Healthcare Applications. *Artificial Intelligence in Medicine* 42(2):95–98.

Scheuermann, R. H.; Ceusters, W.; and Smith, B. 2009. Toward an Ontological Treatment of Disease and Diagnosis. *Proceedings of the 2009 AMIA Summit on Translational Bioinformatics*.

⁴http://www.healthmap.org/en