CSE 586/EE554

Topics in Computer Vision

Course Introduction
Spring 2012

Course Goals

Gain practical knowledge in Computer Vision

- focusing on solution methods
- understanding the underlying math
- knowing when/how to apply methods

Develop skills for being a successful grad student

- programming
- critical thinking
- locating and reading research papers
- technical writing
- presentation skills / rational debate

Course Layout

Traditionally, a vision course is organized by “vision topics”, e.g.

- Stereo
- Tracking
- Recognition
- Segmentation

We are going to do it a little differently

Course Layout

We will organize based on mathematical methods used e.g.

- Mixture Models / EM
- Linear/Quadratic Programming
- Graph Cuts
- Graphical Models
- Monte Carlo Approaches

Underlying vision topics will still be there, but we are indexing them differently and refocusing our effort towards understanding underlying solution techniques.

Why Doing it This Way?

By focusing on solutions rather than problems, we hope to gain:

- practical knowledge about solving vision problems
- intuition about when to apply each method
- competency in using the techniques
- course has "broader impact"

Potential drawback:

This is going to be a very mathematical course.

Course Layout

To address development of research skills, each student will be an <u>active</u> learner

- Implementing useful programs
- Reading primary source material (conferences, journals)
- Writing critical analyses of papers read
- Oral presentation (using projector in front of the class)

Why Doing it This Way?

By focusing on active learning, we hope to gain:

- better practical understanding
- practice research skills
- jump start your own research

Potential drawback:

This course has a heavy workload (but it is very manageable if you work at it a little bit each day)

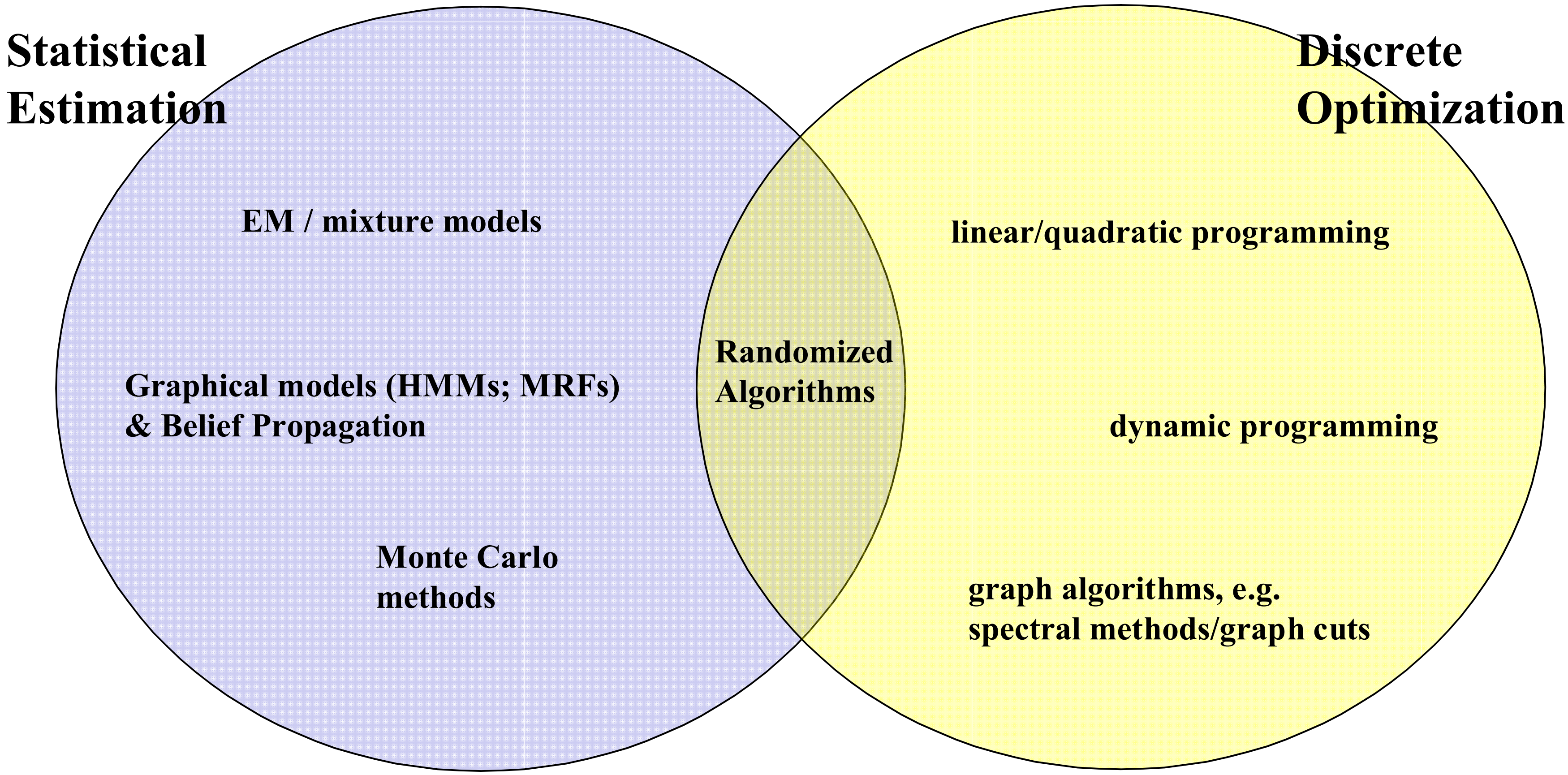
Sample Topics

- Mixture Models and EM
- Linear Programming
- Monte Carlo Methods
- Graphical Models
- “Sparse” Methods
- Graph Cuts / Spectral Methods
- Procrustes Analysis
- Subspace Methods
- Variational Calculus

We surely won’t cover all of these. Some of them alone are suitable for a whole semester course.

Robert Collins
CSE586

Optimization Regimes



How Course is Conducted

Course is broken into “units” lasting two or more weeks.

Each unit focuses on one particular solution method (e.g. EM algorithm).

I will give lectures during the unit to introduce and elaborate on aspects of the basic mathematical approach.

During the unit, we will implement sample code using that solution method, and read/discuss papers that are examples of using that solution method. The papers and examples will typically span different areas of computer vision.

Homework

We will have frequent homework that is a combination of "theory" and programming. We will implement many practical sample programs to reinforce learning the mathematical method being discussed. It should become second nature to you to sit down an "bang out" some simple test code to see if you understand an idea, to verify some result you computed analytically, or to generate a graphical visualization to gain more insight into a problem.

I would suggest using matlab (or octave) because it is easier to program for the kinds of things we will be doing, but you can use any language you want.

Reading/Critiques

During each unit we will each read a set of assigned papers that provide examples of using the method to solve a vision problem. You will write a short critique evaluating one or more of the assigned papers, identifying their pros and cons, and comparing/contrasting how they use the method we are discussing. You will also find another related paper on your own to read and critique. After discussing all assigned papers we will go around the room and have people tell what the paper they found independently is about, why they selected that paper, and whether they recommend it or not.

This whole exercise is meant to develop skills in reviewing, writing, and communication.

Final Project

We will have a final term project that is meant to be a non-trivial exploration of the use of one of the methods we have discussed to some problem and dataset that is of interest to you. The hope is that you will find something we learn in this class useful in your own research, and this project is a way to get started on that. Projects can be done either individually or in groups (I would expect a group project to be larger in scope than a single person project).

Exams

One in-class midterm.

It will be very much related to the homework problems you have done up to that point, and conceptual questions about some of the papers we have read (like, “in your opinion, what was the main point of paper XYZ”)

Grading

- *Midterm:* 20%
- *Final Project:* 20%
- *Homework:* 30%
- *Written Critiques* 20%
- *Oral Presentation:* 10%

Academic Integrity: Although you are encouraged to talk to each other to understand the course material and assignment instructions, when it comes time to doing the assignments, every student is expected to submit their own original work. For programming, standard and publicly available code libraries (such as simple signal processing or linear algebra libraries) may be used after seeking consent of the course instructor or TA.

Sources of Vision Papers

Journals

- *PAMI: IEEE Trans. on Pattern Analysis and Machine Intelligence*
- *IJCV: Int. Journal of Computer Vision*
- *CVIU: Computer Vision and Image Understanding*
- *GMIP: Graphical Modeling and Image Processing*
- *MVA: Machine Vision Applications*
- *IP: IEEE Trans. on Image Processing*

Conference Proceedings

- *CVPR: IEEE Conf. on Computer Vision and Pattern Recognition*
- *ICCV: Int. Conf. on Computer Vision*
- *ECCV: European Conf. on Computer Vision*
- *BMVC: British Machine Vision Conference*
- *ICPR: Int. Conf. on Pattern Recognition*
- *SPIE: Int. Society for Optical Engineering*

Anatomy of a Technical Paper

Abstract
Introduction (to problem)
Related Work
Method (of solution)
Experiments
Conclusion

Acceptable alternative

Abstract
Introduction
Method
Experiments
Related Work
Conclusion

Sometimes “Method” can be expanded into two sections. For example, one focusing on the mathematical theory and one about how it is applied to a vision problem. Or one on description of the method followed by a section on implementation details.

How to Read a Paper

Skim the abstract, intro and conclusion section.
Try to quickly identify
1) the topic area
2) the problem being addressed
3) the approach being taken to solve that problem

Look at the figures, plots and tables
What is the input and output?
Do the results look worthwhile?

Now read the paper to fill in the details.

How to Evaluate/Review a Paper

Refer to Alan Smith's, "The Task of the Referee" on the course web page.

Some questions to ask yourself:

- What is the main contribution of the paper?
- Is the abstract appropriate and adequate digest of the paper?
- Does the introduction clearly state the background?
- Is the paper's coverage of the chosen topic comprehensive?
- Is the performance of the proposed system/algorithm rigorously characterized?
- Does the paper provide enough information to replicate their experiments?
- Is software made available to the community?
- Is the list of references adequate?
- Are the relative lengths of sections appropriate for the material covered?
- Are there too-many/too-few figures? Are all figures appropriately captioned?
- How readable is the paper (regardless of technical content)?

How to Evaluate/Review a Paper

In addition to the generic questions as to what the paper is about and how well it is written, we will also have some method-specific evaluation questions:

- How does the paper illustrate the math method we are discussing in class?
- Is the math necessary or gratuitous for this problem?
- What about the data or problem statement makes our math method applicable?
- What would need be changed for the method NOT to be applicable?
- Does this application add new insights or extend the basic method in any way?
- How does this math method compare to others that could be used?

These are more important for us than the questions on the previous page.

A man who has the knowledge but lacks the power to express it clearly is no better off than if he never had any ideas at all. -- Thucydides, ~400BC

How to Write a Paper

Refer to Armando Fox's writing and presentation page linked to in the course web page.

Refer also to the annotated technical paper skeleton given on the course web page.

Most important:

Determine what your main points are.

State your main points as clearly as possible.

Repeat the main points 3 times (abstract, body, conclusion)

Tell them what you are going to say, say it, then remind them of what you just said.

Critique Format

Please format your critiques with roughly the following sections (in this order)

1 Reviewer: your name and the date
2 Citation: the title, author, year, and publication citation of the paper
3 A one paragraph summary (abstract) of the topic area.
4 A short overview of each paper including a) key ideas, b) technical approaches and c) results. Pay particular attention to the use of the mathematical method we are discussing in class at that time.
5 Comparison with other papers you have read, including strong points and weak points of this paper with respect to others.
6 Questions and issues this raises in your mind

Critique Format

Critiques are graded on a three-level scale: check-minus, check, check-plus. Above average resourcefulness, initiative, creativity and depth of analysis gets a check-plus. Missing any required sections (1-6) or obvious lack of effort on any of them results in a check-minus.

Pay attention to your speling and grammar of English. :-)

How to Give a Presentation

Conference presentation format... ONLY 10-12 minutes.
You must be concise!

Should look like an oral critique with slides, but with expanded emphasis on explaining how our math method is used to solve the problem considered in the paper.

Most important:
- avoid cluttered, wordy presentations with lots of dense math notation
- strive for clear, intuitive presentation
- pictures help get the idea across

Robert Collins
CSE586

How to Give a Presentation

From Stephen Scott's tips on giving a presentation

Concept Class of One-Dimensional Patterns

- The instance space $\mathcal{X}_n$ consists of all configurations of n points on the real line
- A concept is set of all configs. from $\mathcal{X}_n$ within unit distance under Hausdorff metric of some "ideal" configuration of k points, where Hausdorff distance between configs. P and Q is

$$H(P,Q) = \max\left\{\max_{p\in P}\left\{\min_{q\in Q}\{d(p,q)\}\right\}, \max_{q\in Q}\left\{\min_{p\in P}\{d(p,q)\}\right\}\right\}$$

  and $d(p,q)$ is distance between p and q

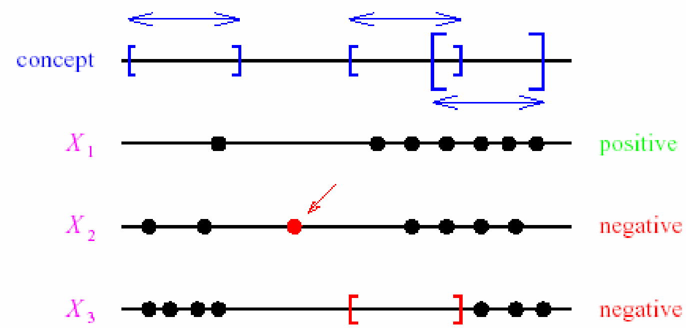
- If P is any configuration of points on $\mathbb{R}$, then concept corresponding to P is $C_P = \{X \in \mathcal{X}_n : H(P,X) \leq 1\}$
- X is a positive example of C_P if $X \in C_P$ and is a negative example otherwise
- Concept class of one-dimensional patterns is $\mathcal{C}_{k,n} = \{C_P : P \text{ is a configuration of } \leq k \text{ points from } \mathbb{R}\}$

9

BAD

Concept Class of One-Dimensional Patterns

- Each concept c is a set of fixed-width intervals on real line
- Each example X is a set of points on real line
- Example X is positive if and only if:
  1. each of X's points lies in an interval from c
  2. each interval of c contains a point from X



10

BETTER

Searching for Material On-Line

The web has become a wonderful resource.

i.e. “Google it”