# Lecture 14

# Canonical SDP Relaxation for CSPs*

## 14.1 Recalling the canonical LP relaxation

Last time, we talked about the canonical LP relaxation for a CSP. A CSP($\Gamma$) is comprised of $\Gamma$, a collection of predicates $R$ with label domain $D$. The canonical LP relaxation is comprised of two parts. Given an instance $\mathcal{C}$, with the domain of the variables being $D$, a constraint will be written as $C = (R, S) \in \mathcal{C}$. Then a solution to the LP relaxation contains two objects. First, for each $v \in V$, a probability distribution over labels for $v$. Formally, we have LP variables $(\mu_v[\ell])_{v \in V, \ell \in D}$ subject to

$$\forall v \in V, \qquad \sum_{\ell \in D} \mu_v[\ell] = 1$$
$$\forall \ell \in D, \quad \mu_v[\ell] \geq 0.$$

Second, for all $C = (R, S) \in \mathcal{C}$ we have a probability distribution $\lambda_C$ over "local assignments" $S \to D$. These are similarly encoded with $\sum_C |D|^{|S|}$ many LP variables.

The objective function is

$$\max \sum_{C=(R,S)\in\mathcal{C}} w_c \mathop{\mathbf{Pr}}_{L \sim \lambda_C}[L(S) \text{ satisfies } R].$$

Finally, the thing that ties the $\mu$'s and the $\lambda$'s together is the *consistent marginals condition* (a collection of linear equalities):

$$\forall C = (R, S) \in \mathcal{C} \quad \forall v \in S \quad \forall \ell \in D, \qquad \mathop{\mathbf{Pr}}_{L \sim \lambda_C}[L(v) = \ell] = \mu_v[\ell].$$

We also showed that rounding the canonical LP relaxation of Max-SAT using plain randomized rounding achieved a $(1 - 1/e)$ approximation ratio. Recall that plain randomized rounding assigns to variables $v$ in the following way:

*Lecturer: Ryan O'Donnell. Scribe: Jamie Morgenstern, Ryan O'Donnell.

$$F(v) = \begin{cases} 1 & : w.p. \quad \mu_v[1] \\ 0 & : w.p. \quad \mu_v[0] \end{cases}$$

The proof of this approximation factor looked at $p_c$, the probability a particular clause was satisfied by $L \sim \lambda_c$, and the probability that $F$ satisfied that clause. In the last lecture it was shown that

$$\mathbf{Pr}[F \text{ satisfies } C] \geq 1 - \left(1 - \frac{p_c}{|S|}\right)^{|S|} \tag{14.1}$$

When $|S| \leq 2$, 14.1 $\geq (3/4)p_c$ which implies that this algorithm gets a 3/4-factor for clauses of length at most 2.

On the other hand, for clauses of length at least 2, the trivial random algorithm (assigning each variable to 1 or 0 with probability 1/2) satisfies 3/4 of clauses, yielding a 3/4 approximation. Can we get the best of both worlds, and combine the results for trivial random and plain randomized rounding of the LP relaxation to get a 3/4 approximation for Max-SAT?

The answer is yes, by combining the two assignment schemes. If we create our assignment $F$ as

$$F(v) = \begin{cases} 1 & : w.p. \quad \text{avg}\{\mu_v[1], 1/2\} \\ 0 & : w.p. \quad \text{avg}\{\mu_v[0], 1/2\} \end{cases}$$

then $F$ will satisfy 3/4 of clauses in expectation for Max-SAT. Showing this will be on the homework.

In fact it is possible to do better than a 3/4 approximation for various versions of Max-SAT. Below we give a laundry list of results proven using *SDPs* to improve this approximation ratio for Max-SAT.

| | |
|---|---|
| $(\alpha_2(\beta), \beta)$ approximation for Max-2SAT, where | |
| $\alpha_2(\beta) \geq .940\beta$ | [LLZ02] |
| and $\alpha_2(1-\epsilon) \geq 1 - O(\sqrt{\epsilon})$ | [CMM07] |
| $(\frac{7}{8}\beta, \beta)$ for Max-3SAT, | [KZ97] (computer-assisted), |
| | [Zwi02] (computer-verified) |
| $(\frac{7}{8}, 1)$ for Max-4SAT | [HZ99] |
| $(.833\beta, \beta)$ for Max-SAT | [AW00] |

It is reasonable to conjecture that there is a polynomial-time $(\frac{7}{8}\beta, \beta)$-approximation algorithm for Max-$k$SAT for any $k$.

## 14.2 Canonical CSP SDP relaxation

The SDP relaxation is similar to the LP relaxation, but with an important generalization. We will have exactly the same $\lambda_C$'s for each constraint, and the same objective function. Rather than having the $\mu_v$'s, however, we'll have a *collection of joint real random variables*

$(I_v[\ell])_{v \in V, \ell \in D}$. We will also have constraints which cause these random variables to hang together with the $\lambda_C$'s in a gentlemanly fashion.

The random variables $I_v[\ell]$ will be called *pseudoindicator random variables.* We emphasize that they are jointly distributed. You should think of them as follows: there is a box, and when you press a button on the side of the box ("make a draw"), out comes values for each of the $|V||D|$ random variables.

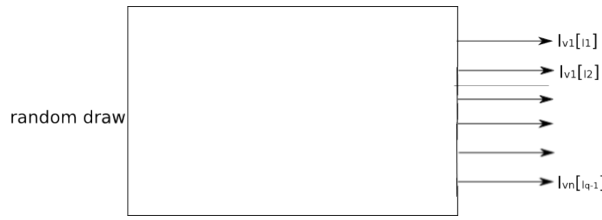


Figure 14.1: The pseudoindicator joint draw $I_v[l]$.

For now, never mind about how we actually represent these random variables or enforce conditions on them; we'll come to that later.

We'd love if it were the case that these pseudoindicator random variables were actual indicator random variables, corresponding to a genuine assignment $V \to D$. However, we can only enforce something weaker than that. Specifically, we will enforce the following two sets of conditions:

1. **Consistent first moments:**

$$\begin{aligned} \forall C = (R, S) \in \mathcal{C} \\ \forall v \in S \\ \forall \ell \in D \\ \Pr_{L \sim \lambda_C}[L(v) = \ell] = \mathbf{E}[I_v[\ell]] \end{aligned} \tag{14.2}$$

2. **Consistent second moments:**

$$\begin{aligned} \forall C = (R, S) \in \mathcal{C} \\ \forall v, v' \in S \\ \forall \ell, \ell' \in D \\ \Pr_{L \sim \lambda_c}[L(v) = \ell \wedge L(v') = \ell'] = \mathbf{E}\Big[I_v[\ell] \cdot I_{v'}[\ell']\Big] \end{aligned} \tag{14.3}$$

(We emphasize that $v$ and $v'$ need not be distinct, and $\ell$ and $\ell'$ need not be distinct.)

We also emphasize again that these pseudoindicator random variables are *n*ot independent, so the expected value of their product is not the product of their expected values.

We will show we can solve this optimally as an SDP (there are actually vectors "hiding inside the box"). Also, as explain more carefully in the next lecture, assuming the Unique Games Conjecture the best polynomial-time approximation for *any* CSP is given by this SDP.

Now, a few remarks about this relaxation. First:

**Remark 14.1.** For all $v, \ell$, $\mathbf{E}\Big[I_v[\ell]\Big] = \mathbf{E}\Big[I_v[l]^2\Big]$.

*Proof.* Consider any $C \ni v$. Apply (2) with $v = v'$, $\ell = \ell'$. Then, we have

$$\mathop{\mathbf{Pr}}_{L\sim\lambda_c}[L(v) = l \wedge L(v) = l] = \mathbf{E}\big[I_v[\ell]^2\big].$$

Of course also

$$\mathop{\mathbf{Pr}}_{L\sim\lambda_c}[L(v) = \ell \wedge L(v) = \ell] = \mathop{\mathbf{Pr}}_{L\sim\lambda_c}[L(v) = \ell].$$

Finally, apply (1) which says

$$\mathop{\mathbf{Pr}}_{L\sim\lambda_c}[L(v = \ell)] = \mathbf{E}\big[I_v[\ell]\big]$$

□

This is somewhat nice because $\mathbf{E}[I^2] = \mathbf{E}[I]$ is something satisfied by a genuinely 0-1-valued random variable $I$. In fact, our pseudoindicator random variables may take values outside the range $[0, 1]$. Still, they will at least satisfy the above.

Now, we will show that this "SDP relaxation" is in fact a relaxation (we still haven't explained why it's an SDP):

**Theorem 14.2.** $\mathrm{Opt}(\mathcal{C}) \leq \mathrm{SDPOpt}(\mathcal{C})$

*Proof.* Let $F$ be a legitimate (optimal) assignment. Then we can construct $\lambda_C$'s and $I_v[\ell]$'s which achieve $\mathrm{Val}(F)$.

$$\lambda_C[L] = \begin{cases} 1 & : \text{ if } L \text{ is consistent with } F \\ 0 & : \text{ o/w} \end{cases}$$

Then, let $I_v[\ell]$ be the *constant* random variables

$$I_v[\ell] \equiv \begin{cases} 1 & : \text{ if } F(v) = \ell \\ 0 & : \text{ o/w} \end{cases}$$

It is easy to check that these $\lambda_C$'s and $I_v[\ell]$'s satisfy the consistent first and second moment constraints and have SDP value equal to $\mathrm{Val}(F)$. □

Now, we show that the SDP relaxation is at least as tight as the LP relaxation.

**Theorem 14.3.** SDPOpt($\mathcal{C}$) $\leq$ LPOpt($\mathcal{C}$)

*Proof.* Given an SDP solution $\mathcal{S}$ achieving SDPOpt, $\mathcal{S} = (\lambda_C, I_v[\ell])$, we must construct an LP solution and show its objective is no less than that of $\mathcal{S}$. Use the same $\lambda_C$'s for the LP solution. Since the objective value depends only on the $\lambda_C$'s, the objective value for the LP solution will be the same as the SDP value of $\mathcal{S}$. It remains to construct the distributions $\mu_v$ which are consistent with the $\lambda_C$'s. Naturally, we set

$$\mu_v[\ell] = \mathbf{E}\Big[I_v[\ell]\Big].$$

Please note that this is indeed a probability distribution, because if we select any $C \ni v$ and apply (1), we get

$$\mathbf{E}\Big[I_v[\ell]\Big] = \mathop{\mathbf{Pr}}_{L\sim\lambda_C}[L(v) = \ell]$$

and the RHS numbers are coming from the genuine probability distribution $\lambda_C|_v$. The fact that the $\lambda_C$'s and the $\mu_v$'s satisfy the LP's "consistent marginals" condition is equivalent to (1). $\square$

Next, fix some $v \in V$. If the pseudoindicator random variables $(I_v[\ell])_{\ell \in D}$ were really legitimate constant random variables indicating a genuine assignment, we'd have $\sum_{\ell \in D} I_v[\ell] = 1$. In fact, this is true with probability 1 in any SDP solution:

**Proposition 14.4.** *Given a valid SDP solution, for any $v$, let $J = J_v = \sum_{\ell \in D} I_v[\ell]$. Then $J \equiv 1$.*

*Proof.* We will calculate the mean and variance of $J$. By linearity of expectation,

$$\mathbf{E}[J] = \sum_{\ell} \mathbf{E}[I_v[\ell]] = 1.$$

And,

$$\mathbf{E}[J^2] = \mathbf{E}\left[\left(\sum_{\ell} I_v[\ell]\right)\left(\sum_{\ell'} I_v[\ell']\right)\right]$$

By linearity of expectation, this is just

$$= \sum_{\ell,\ell'} \mathbf{E}\Big[I_v[\ell] \cdot I_v[\ell']\Big]$$

Choose any constraint $C \ni v$. By (2), we have

$$= \sum_{\ell,\ell'} \mathop{\mathbf{Pr}}_{L\sim\lambda_C}[L(v) = \ell \wedge L(v) = \ell']$$

Here every term with $\ell \neq \ell'$ is 0. So this reduces to

$$= \sum_{\ell} \mathop{\mathbf{Pr}}_{L \sim \lambda_c} [L(v) = \ell]$$
$$= 1$$

Then, computing the variance of $J$:

$$\mathbf{Var}[J] = \mathbf{E}[J^2] - \mathbf{E}[J]^2 = 0$$

Any random variable with zero variance is a constant random variable, with value equal to its mean. Thus, $J \equiv 1$. $\square$

**Theorem 14.5.** *Condition (1) in the SDP is superfluous, in the sense that dropping it leads to an equivalent SDP (equivalent meaning that the optimum is the same for all instances).*

*Proof.* On the homework. $\square$

Given the above theorem, we focus for a while on the optimization problem in which joint pseudoindicator random variables only need to satisfy (2). Let's now answer the big question: how is this optimization problem an SDP?

## 14.3 Why is it an SDP and how do we construct the pseudoindicators?

Let us define the numbers
$$\sigma_{(v,\ell),(v',\ell')} = \mathbf{E}\Big[I_v[\ell] \cdot I_{v'}[\ell'].\Big]$$

As this notation suggests, we will define a matrix $\Sigma$ from these numbers. It will be an $N \times N$ matrix (for $N = |V||D|$), with rows and columns indexed by variable/label pairs:

$$\Sigma = (\sigma_{(v,\ell),(v',\ell')})$$

Now let us ask what the consistent second moments condition (2) is saying? The second moments constraint is satisfied if and only there exists a collection of $N$ random variables (the pseudoindicators) whose second moment matrix is $\Sigma$. But, if you recall, this is equivalent definition #5 from Lecture 10 of PSD-ness of the matrix $\Sigma$. Thus our optimization problem — which has linear constraints on the variables $\lambda_C$ and $\sigma_{(v,\ell),(v',\ell')}$, together with the condition that $\Sigma$ is PSD — is indeed an SDP!

We still need to discuss how to actually "construct/sample from" pseudoindicator random variables ($I_v[\ell]$) corresponding to the Ellipsoid Algorithm's output $\Sigma$. It's much like in the beginning of the Goemans–Williamson algorithm: given $\Sigma$ PSD, you compute (a very accurate approximation to) a matrix $U \in \mathbb{R}^{N \times N}$ such that $U^\top U = \Sigma$. The columns of $U$ are vectors $\vec{y}_{v,\ell} \in \mathbb{R}^N$ such that $\vec{y}_{v,\ell} \cdot \vec{y}_{v',\ell'} = \sigma_{(v,\ell),(v',\ell')}$. How does this help?

The key idea is that **you can think of a vector as a random variable, and a collection of vectors as a collection of joint random variables**. How? A vector $\vec{y} \in \mathbb{R}^N$ defines a random variable $Y$ as follows: to get a draw from $Y$, pick $i \in [N]$ uniformly at random and then output $Y = \vec{y}_i$. A collection of vectors

$$\vec{y}^{(1)}, \dots, \vec{y}^{(d)}$$

defines a collection of jointly distributed random variables

$$Y^{(1)}, \dots, Y^{(d)}$$

as follows: to get one draw from (all of) the $Y^{(j)}$'s, pick $i \in [N]$ uniformly at random and then output $Y^{(j)} = (\vec{y}^{(j)})_i$ for each $j \in [d]$. In this way, we can view the vectors that the SDP solver outputs (more precisely, the vectors gotten from the columns of the factorization $U^\top U = \Sigma$), namely

$$\vec{y}_{(v_1,\ell_1)}, \dots, \vec{y}_{(v_n,\ell_q)},$$

as the collection of jointly distributed pseudoindicators,

$$Y_{v_1}[\ell_1], \dots, Y_{v_n}[\ell_q].$$

Why does this work? The idea is that "**inner products are preserved**" (up to a trivial scaling):

**Observation 14.6.** Given vectors $\vec{y}, \vec{y}' \in \mathbb{R}^N$, the equivalent random variables $Y$, $Y'$ satisfy:

$$\mathbf{E}[YY'] = \sum_{i=1}^{N} \frac{1}{N} \vec{y}_i \vec{y}'_i = \frac{1}{N} \vec{y} \cdot \vec{y}'$$

We'll make a slight definition to get rid of this annoying scaling factor:

**Definition 14.7.** We introduce the scaled inner product

$$\langle\langle \vec{y}, \vec{y}' \rangle\rangle := \frac{1}{N} \vec{y} \cdot \vec{y}'$$

Solving the SDP is equivalent to coming up with the pseudoindicator random variables, with this slight need to scale. Given $\vec{y}_{v,\ell}$ as in the original SDP, we define

$$\vec{z}_{v,\ell} = \sqrt{N} \vec{y}_{v,\ell}$$

Then,

$$\langle\langle \vec{z}_{v,\ell}, \vec{z}_{v',\ell'} \rangle\rangle = N \langle\langle \vec{y}_{v,\ell}, \vec{y}_{v',\ell'} \rangle\rangle = \vec{y}_{v,\ell} \cdot \vec{y}_{v',\ell'} = \sigma_{(v,\ell),(v',\ell')}$$

So actually, the joint random variables corresponding to this collection of vectors $\vec{z}_{v,l}$'s will be the pseudoindicator random variables.

## 14.4 Summary

There are several equivalent perspectives on the canonical SDP relaxation for a CSP.

- Pseudoindicator random variables which satisfy the first and second moment consistency constraints. This perspective is arguably best for understanding the SDP. for using an SDP solver
- Pseudoindicator random variables which just satisfy the consistent second moments constraints. This perspective is arguably best when constructing SDP solutions by hand.
- Vectors $(\vec{y}_{v,\ell})$ satisfying the first and second "moment" consistency constraints. This perspective is the one that's actually used computationally, on a computer.

There is one more equivalent perspective that we will see in the next lecture, which is arguably the best perspective for developing "SDP rounding algorithms":

- *Jointly Gaussian* pseudoindicator random variables which satisfy the consistent first and second moment constraints.

In the next lecture we will see how to make the pseudoindicators jointly Gaussian, and why this is good for rounding algorithms.

# Bibliography


[AW00] T. Asano and D.P. Williamson. Improved approximation algorithms for max sat. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 96–105. Society for Industrial and Applied Mathematics, 2000. 14.1

[CMM07] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 62–68. Society for Industrial and Applied Mathematics, 2007. 14.1

[HZ99] E. Halperin and U. Zwick. Approximation algorithms for max 4-sat and rounding procedures for semidefinite programs. *Integer Programming and Combinatorial Optimization*, pages 202–217, 1999. 14.1

[KZ97] H. Karloff and U. Zwick. A 7/8-approximation algorithm for max 3sat? In *focs*, page 406. Published by the IEEE Computer Society, 1997. 14.1

[LLZ02] D. Livnat, M. Lewin, and U. Zwick. Improved rounding techniques for the max 2-sat and max di-cut problems. In *Proc. of 9th IPCO*, pages 67–82, 2002. 14.1

[Zwi02] U. Zwick. Computer assisted proof of optimal approximability results. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 496–505. Society for Industrial and Applied Mathematics, 2002. 14.1